User's Guide for POWERSIM and Associated Programs

Eric Redding and Scott Fish
Institute for Advanced Technology
The University of Texas at Austin

August 1996

IAT.M 0003

Certificate of Technical Review

Scott Fish

Patrick H. Sullivan

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1996	Manual; Aug 1994 — Dec 1995

4. TITLE AND SUBTITLE
User's Guide for POWERSIM and Associated Programs

5. FUNDING NUMBERS
Contract # DAAA21-93-C-0101

6. AUTHOR(S)
Eric Redding and Scott Fish

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Advanced Technology
The University of Texas at Austin
4030-2 W. Braker Lane, #200
Austin, TX 78759

8. PERFORMING ORGANIZATION REPORT NUMBER
IAT.M 0003

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-WT-T
Aberdeen Proving Ground, MD 21005-5066

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The view, opinions and/or findings contained in this report are those of the author(s) and should not be considered as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(Maximum 200 words)*

This report provides an overview of the power simulation code developed by the IAT for analysis of energy and power flow throughout combat vehicles equipped with electric weapons. The purpose of the code, which has been nicknamed POWERSIM, is to provide an intuitive modeling environment for this analysis under the rather complex mission sequences associated with battlefield operation of these vehicles. The utility of simulating the power flow through the vehicle under these conditions hinges on assessing the impact of all the power system components on the overall vehicle system performance under conditions (movement and firing) which are considered realistic for the application of interest. The user of this code, can through the appropriate graphical user interfaces, (1) configure complex mobility and firing sequences, (2) configure a hybrid electric combat vehicle with electric weapons, (3) run simulations for mobility loads alone, (4) run simulations with mobility and firepower, and (5) perform some simple analysis of the simulation results. Additional analysis can easily be performed on the wealth of data which is generated on individual component conditions as needed through user defined functions. The code described here is the first complete version of the code, and development continues on adding component models and increasing the accuracy and speed of the existing component models.

14. SUBJECT TERMS
combat vehicles; electric guns; electric vehicles; hybrid electric combat vehicles; hybrid electric vehicles; simulation

15. NUMBER OF PAGES
52

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Executive Summary

This manual provides an overview of the power simulation code developed by the Institute for Advanced Technology for analysis of energy and power flow throughout combat vehicles equipped with electric weapons. The purpose of the code, which has been nicknamed POWERSIM, is to provide an intuitive modeling environment for this analysis under the rather complex mission sequences associated with battlefield operation of these vehicles. The utility of simulating the power flow through the vehicle under these conditions hinges on assessing the impact of all the power system components on the overall vehicle system performance under conditions (movement and firing) which are considered realistic for the application of interest. The user of this code, can through the appropriate graphical user interfaces, (1) configure complex mobility and firing sequences, (2) configure a hybrid electric combat vehicle with electric weapons, (3) run simulations for mobility loads alone, (4) run simulations with mobility and firepower, and (5) perform some simple analysis of the simulation results. Additional analysis can easily be performed on the wealth of data which is generated on individual component conditions as needed through user defined functions.

The code described here is the first complete version and development continues on adding component models and increasing the accuracy and speed of the existing component models.

Table of Contents

1 Introduction

1.1 Manual Overview

This manual is broken up into several sections. The first section includes chapters devoted to an overview of the program and the manner in which the simulation is conducted. The appendices then go through and specifically illustrate how to conduct an actual simulation with test cases provided. The last few appendices are meant as a reference guide for the experienced user seeking to analyze parameters not previously addressed.

1.2 POWERSIM Introduction

POWERSIM is a program designed to simulate and monitor the power and energy flow throughout a hybrid electric combat vehicle. These simulations can be used to examine the effects of power system architectures and various component characteristics on the overall vehicle performance. Flexibility in modeling both the desired mission profiles for the vehicle, and the vehicle power system are the cornerstones of POWERSIM. The POWERSIM simulations are conducted in the Matlab Simulink environment.

1.3 Matlab/ Simulink

Matlab is a high level programming language composed of highly efficient mathematical tools and associated graphics commands. Simulink is a program written in Matlab, known as a toolbox of Matlab, which augments these capabilities with a graphical programming interface for the development of simulations for solving dynamic problems. Simulink allows the user to concentrate on model development rather than on the simulation method (of which there are several to choose from). It also contains tools for the development of control systems and evaluating model performance. Simulink allows fixed or variable time step algorithms from simple Euler to Runge-Kutta to Adams-Gear techniques as selected by the user based on the type of system being studied.

Typically, simulations are conducted in batch processing and then trends determined by varying certain parameters can be established. Therefore, there is no real time data currently available since that would slow the simulation process down dramatically. However, if the user seeks to monitor certain parameters as the simulation progresses, then the code will have to be altered with Simulink scopes inserted. Consult the Simulink documentation for this.

1.4 Steps to Simulation

1.4.1 Mission Description

The mission definition of the simulation allows the user to prescribe a mission over specified terrain through the use of both graphics-based plots and text interfaces. In the process of defining this mission, the user inputs a beginning and endpoint for the entire mission, as well as annotate up to 25 intermediate waypoints. For each waypoint, a velocity is specified, along with a resting time if waypoint velocity is zero. Then the user graphically inputs the direction of movement while passing through this waypoint. Finally, the program interpolates the values from each waypoint and produces an array of values defining the vehicle location, velocity, and direction of travel in much more detail.

The second portion of the mission input allows the user to specify firing sequences. First, firing points are noted on the plot, then the user is allowed to dictate the energy per shot, number of shots at that position, and time between multiple shots.

1.4.2 Vehicle Systems Description

A vehicle systems input window is utilized to simply input a detailed description of the individual system parameters on board the vehicle. The window incorporates a graphical user interface (GUI). This graphical interface allows the user to decide the functional type of vehicle, power plant for the vehicle, and various parameters about each subsystem on that vehicle, including the prime power, the mobility, armament, and pulsed-power systems.

1.4.3 Simulations

After defining the mission and the vehicle, an actual simulation of such a mission with that vehicle can be conducted by the user. This simulation requires two parts. First, the simulation runs without a firing sequence to provide baseline data by which to judge the subsystems on board the vehicle. Then, the mission is simulated again with the desired firing sequence so that the effect on energy levels can be seen.

1.4.4 Post-Processing \ Resulting Plots

The last step in the simulation process is to attempt to understand what the simulation's results are indicating with regard to the system. This analysis is done by plotting the time history of various systems from the simulation against each other. For example, a very descriptive plot is the energy level of the entire system versus time plotted alongside the energy demand of the various components. Likewise, another plot presented to the user is the delay time of a shot caused by an energy demand exceeding the available energy.

2 File Structure

2.1 Introduction

For the most efficient method of using POWERSIM, the data files are split into several subdirectories. Each type of data file has its own subdirectory, thus there is a directory for missions, vehicles, terrains, and finally a data directory for results files. The following sections list the directory tree followed by the actual computational and graphical user interface (GUI) files.

2.2 Directory List

homedir	current directory that houses POWERSIM, GUI's
homedir/data	data directory for output results from simulations
homedir/mission	directory for missions
homedir/mission/ill	directory for Adobe Illustrator plots of mission input
homedir/vehicle	directory where vehicle description files are stored
homedir/vehicle/tmp	directory for temporary vehicles files
homedir/terrain	directory for terrain files

2.3 Power Simulation File List

psim.m	starts GUI's, declares global variables
mainmenu.m	main menu
newveh_gui.m	vehicle system inputs
runsim_gui.m	runs actual simulations
analysis_gui.m	GUI allowing easy access to plots
editaux_gui.m	auxiliary systems GUI, reached through vehicle GUI
dialogbox.m	multipurpose-use GUI in middle of screen
mission_input.m	m-file that defines missions
f_path_edit.m	defines the actual path inside the missions; called by mission_input
f_power.m	defines firing power; called by mission_input
f_fire_position.m	defines firing positions; called by mission_input
update_var.m	m-file that calculates mass and updates several other variables; called by newveh_gui.m
savedat.m	m-file that is used to create pure results files without extraneous parameters
saveveh.m	m-file, same as savedat.m, except specialized for newveh_gui.m
turbine.m	m-file run to create data file turbine.mat, which is loaded by update_var.m
diesel.m	m-file run to create data file diesel.mat, which is loaded by update_var.m
runsim3.m	script m-file executed from runsim_gui.m
powersim1.0.m	Simulink file called by runsim3.m
fire_controller3.m	trigger mechanism called by POWERSIM1.0.m Simulink

auxiliaries.m	m-file to calculate auxiliaries, called by POWERSIM1.0.m
main_systems_plot.m	plot to look at several main systems simultaneously
prime_mover_plot.m	plot comparing prime mover from mobility case to full shot sequence case
auxiliaries_plot.m	plot to see the load from auxiliary systems in the mission
shot_analysis.m	plot to examine shot times, strengths, and delays
save_message.m	small file that saves output directly from POWERSIM1.0
helpme.m	small file to tell what plots are in which windows
documentation.txt	this file list, including file descriptions

3 POWERSIM

3.1 Introduction

This section describes in general the actual models of the POWERSIM simulation that have been programmed into the Matlab Simulink code. Simulink allows a great degree of flexibility in such models, allowing the designer to quickly adapt a baseline model to a particular vehicle. In doing so, the modular can provide a greater degree of depth and thus authenticity in truly representing the systems being modeled. The next section discusses the baseline model currently available.

3.2 Current Model

The current model is based upon a hybrid-electric drive combat vehicle incorporating a pulsed-power weapon system. In this model, the subsystem blocks are primarily concerned with vehicle dynamics, the weapon system, or the energy generation and storage system. Figure 3-1 is the graphical top-level model from Simulink.

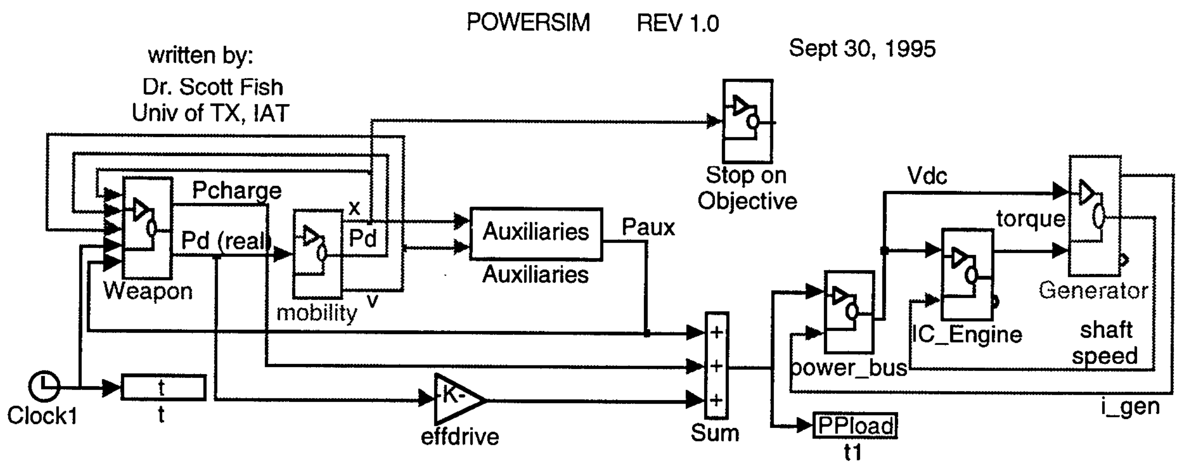

Fig. 3-1. Simulink Model - Top Level

3.3 Current Subsystem Blocks

The following is a list of the major subsystems on this simulated tank. In addition to these subsystems, several predefined global parameters remain at constant values throughout the simulation but are referenced by various subsystems at each time step. These parameters are based on inputs saved in the vehicle or mission files.

3.3.1 Main Weapon

Description:

The purpose of this subsystem block is to simulate the weapon system on board a hybrid electric tank. This block includes a pulsed power charging system, an energy storage system, and a simulated trigger. The charging system and the trigger mechanism are both calculated within a Matlab m-file (required to be in the same directory as the powersim file). The energy storage system in this instance consists of a resettable integrator that resets to the previous energy setting less the energy of the current shot to give a new available energy store value.

Inputs: Position (m), Velocity (m/s), Desired Drive Power (W), Auxiliary Power (W),
Time (s)

Outputs: Desired Pulsed-Power Charger Power (W), Actual Drive Power (W)

3.3.2 Mobility

Description:

The purpose of this subsystem block is to simulate the vehicle dynamics. Functionally, this system works by summing the drag associated with the rolling resistance, aerodynamics, and slope, then subtracting this from the total power available from the bus. The remaining power is assumed to go to acceleration, less efficiency losses, and is integrated once to determine velocity and a second time to find distance traveled.

The driver block is also included here, and works by driving the difference between desired velocity and actual velocity to zero through the use of a PID controller. The output of the PID is limited to a range bounded by zero and the theoretical maximum drive power of the vehicle. This maximum drive power is constant and stated in the vehicle definition.

Inputs: Actual Drive Power (W)

Outputs: Position (m), Velocity (m/s), Desired Drive Power (W)

3.3.3 Engine

Description:

The purpose of this subsystem block is to simulate an internal combustion engine that provides a torque to the generator. For maximum efficiency, this engine is designed to maintain a roughly constant shaft speed, modeled through the use of induced engine lag and a controller driving the difference between the real speed and the constant reference speed to zero. The torque output is calculated through a two-dimensional lookup table describing the relationship between shaft speed, throttle setting, and torque. Another lookup table is used to calculate the fuel consumed at that same time step as well and save this value to the Matlab workspace.

Inputs: Shaft Speed (rad/sec), Bus Voltage (V)

Outputs: Engine Torque (Nm), Power (W)

3.3.4 Generator

Description:

The purpose of this subsystem block is to simulate the generator used to provide energy to the power bus. The generator model contains a PID controller to control the current, and maintains constant proportionality between torque input and current output. Furthermore, it takes the inertia of the generator shaft compared with the current output to determine the shaft speed to feedback to the engine.

Inputs: Voltage of the power bus (V), Torque from the Prime Mover (Nm)

Outputs: Current (A), Generator / Engine Shaft Speed (rad/sec)

3.3.5 Power Bus

Description:

The purpose of this subsystem block is to simulate the electrical bus. Included in this modeling process are integration blocks modeling the battery or flywheels for the storage of energy. Just in front of this block is a load collector, which sums the loads of the various components requiring power from the electrical bus. Then, just inside the block, that load power is divided by the nominal bus voltage to get the load current and is subtracted from the generator current to determine the remaining current to be sent to energy storage. The power bus is designed to handle the continuous loads as well as maintain a weapon energy storage.

Inputs: Load Power (W), Generator Current (A)

Outputs: Actual Bus Voltage (V)

3.3.6 Auxiliaries

Description:

The purpose of this subsystem block is to simulate the load created by the various on board auxiliary systems such as the power turret, power loader, and the sum of the remaining auxiliaries (a constant in this case). This model is calculated primarily within a script file that determines when the auxiliary devices and associated power demands should enter into the system. For example, the power loader and turret drive must have greater power as a shot is being taken than just the power required to move over terrain.

Inputs: Position (m), Velocity (m/s)

Outputs: Auxiliary Load Power (W)

3.4 Future Improvements

Future planned improvements include the inclusion of a better trigger mechanism block as well as auxiliary subsystem block, both of which rely on programs other than the simulation's mathematical model. As work progresses, each of the blocks will undoubtedly be refined as better baseline data becomes available and such level of detail becomes required.

4 Mission Definition

4.1 Introduction

The mission definition sequence utilizes many programs to put together a set of arrays that define the position, velocity, and firing sequences of the vehicle, with respect to a designated terrain map, throughout a mission. These arrays are interpolations of waypoints input by the user that specify position, speed, and direction of travel at that instant in time.

4.1.1 Input Files

The only input files necessary are the terrain files, as described in Section 4.2. However, the mission can currently have the firing sequence redefined by simply using the mission data files previously defined. The program utilizes the waypoints previously defined then allows the user to graphically input a new shot sequence.

4.1.2 Files Created

For a mission named <missname>, the following four files are the result of the mission input procedure:

<missname>.mat	mission's waypoint and velocity parameters
<missname>wp.mat	a listing of only the mission's waypoints
< missname>fp.mat	mission firing positions and shot energy levels
<missname>pwr.mat	interpolated mission position, velocity, slope, etc.

For example, if the user were to name the mission 'tempmiss', then the following files would be created:

tempmiss.mat
tempmisswp.mat
tempmissfp.mat
tempmisspwr.mat

4.2 Terrain

The terrain file is the basis by which the mission is defined. In essence, the terrain file is simply a contour map that allows the mission definition program to determine the slope and elevation of the mission being defined. Terrain files are generated by using the contour plotting abilities of Matlab based upon some reference topographic map. Figure 4-1 is a plot of a typical terrain file, in this case the standard input file, which is a generic contour plot named hlsmooth.mat.

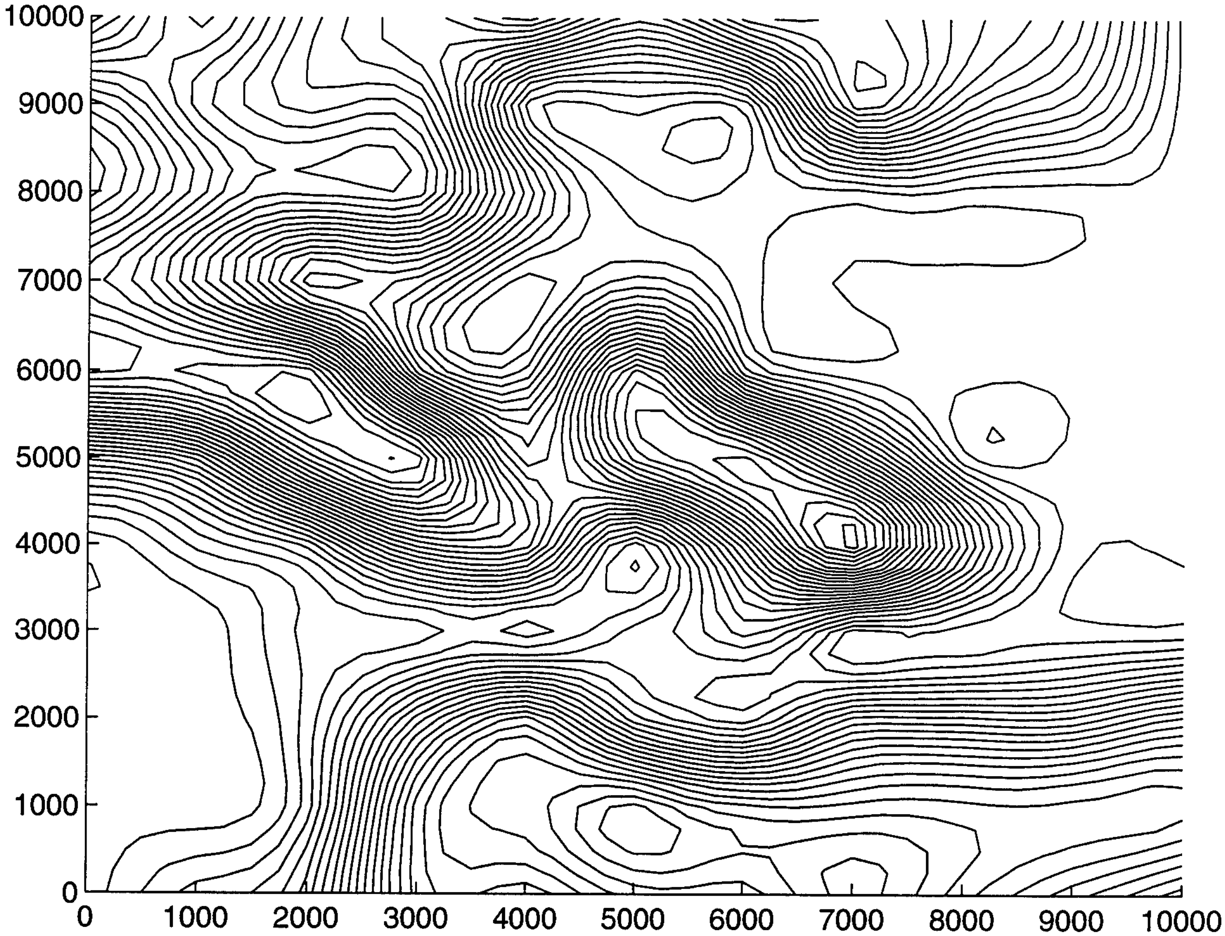

Figure 4-1. Typical Terrain Plot

4.3 Waypoint Definition

Waypoints are the specified intermediate points a vehicle passes though in the process of reaching the final destination. Waypoints typically delineate a path used to bypass obstacles such as harsh slopes or known threats and, in this case, are simply defined as reference points on a given terrain map.

In the first part of the mission input procedure, the waypoints are entered after a beginning and end point are noted. Between 2 and 25 points can be noted on the terrain plot, besides the beginning and end points.

4.4 Velocity Definitions

The velocities are the next parameter to be defined by the user. These are done one at a time, from waypoint to waypoint. After inputting the velocity at each point, the user is also prompted to indicate the direction of travel when crossing that waypoint. Once the waypoints and velocities are input, the program then interpolates these waypoints' directions and velocities and defines a contiguous pathline to follow.

4.5 Firing Sequence

4.5.1 Firing Positions

The next process in the mission definition is the designation of firing positions. In this step, the user simply clicks the mouse at the desired firing position, then enters a text description of the shot at that position in the Matlab command line window.

4.5.2 Shot Description

The shot description consists of three parameters, the energy of the shot, the number of shots at that position, and the time between shots if multiple shots are being taken at that position. Figure 4-2 shows the resulting splined course of travel as well as the firing points.

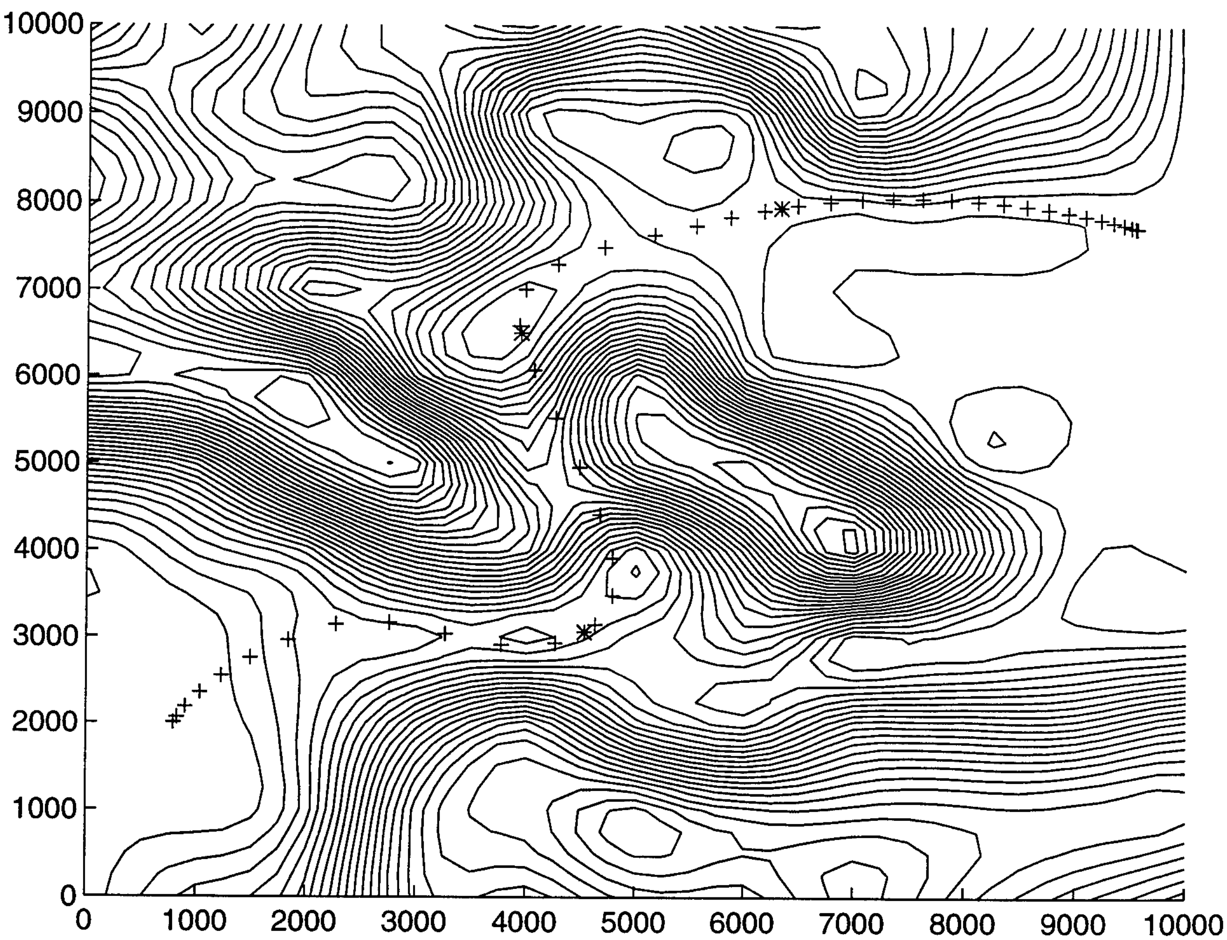

Figure 4-2. Terrain Plot with Waypoints and Firing Positions

4.6 Mission Revision

Presently, only the firing sequence can be redefined. The shot sequence redefinition is accomplished by running the mission input program again, then typing the name of the mission already defined. The program prompts the user to enter new waypoints or use existing waypoints. By using the existing waypoints, the user will be transferred to the firing sequence definition portion of the program.

4.7 Future Improvements

This area of the simulation process could be greatly improved with the addition of graphical user interfaces that would reduce the amount of erroneous inputs as well as speed up mission production time.

5 Vehicle Definition

5.1 Introduction

The Vehicle System Inputs window allows the user to optimize the majority of system parameters to fit the particular vehicle being simulated. The input process consists of several windows of design parameters, most of which can be left at their default values, which in turn are used by the program to calculate additional dependent parameters such as vehicle mass and energy storage requirements. Figure 5-1 shows a picture of the actual GUI.

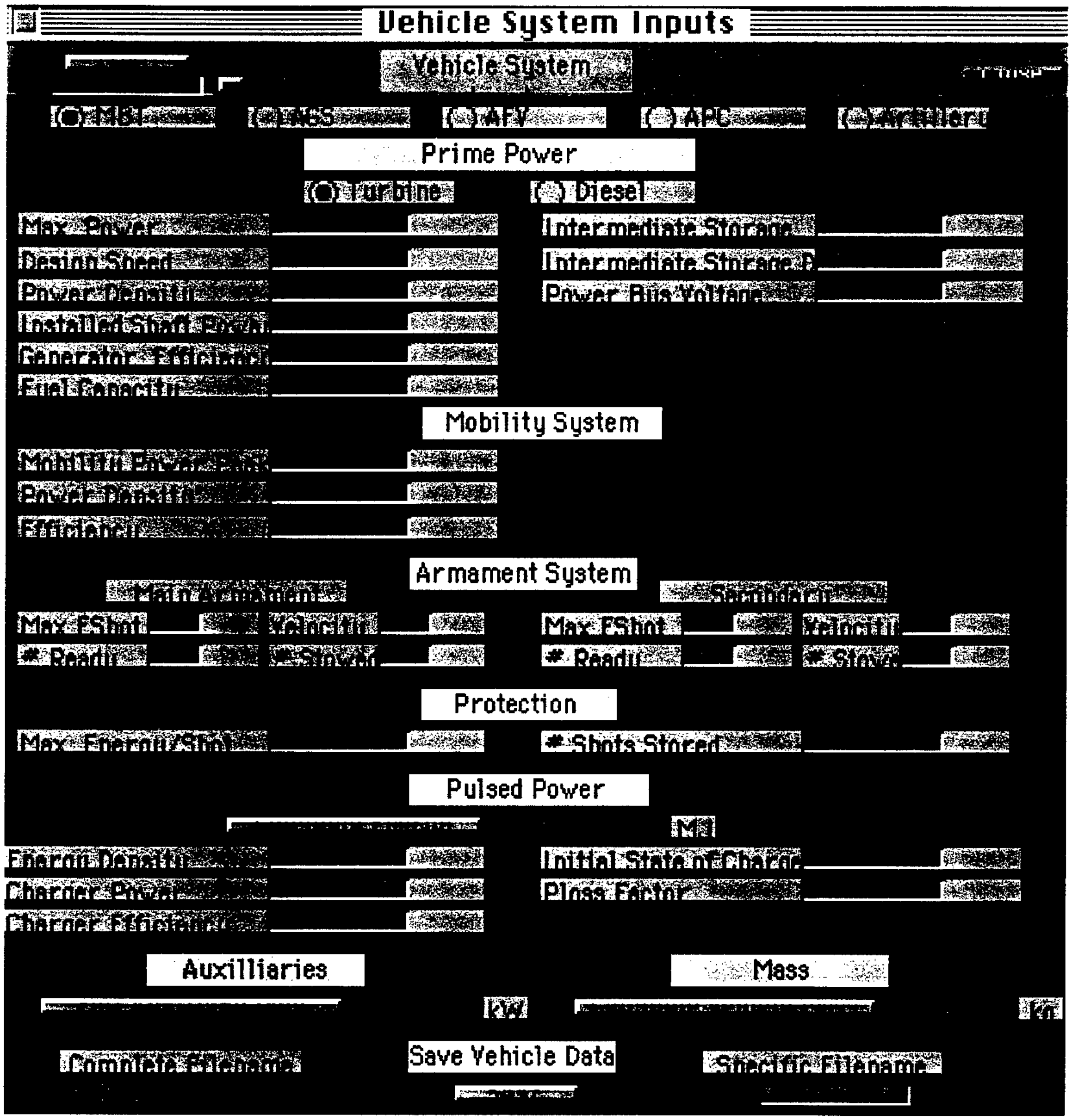

Figure 5-1. Vehicle Systems Input GUI

5.2 Top Level Design Constraints

5.2.1 Functionality

The Vehicle System Inputs window allows five different types of vehicles to be defined. They are:

Main Battle Tank (MBT)
Armored Gun System (AGS)
Armored Fighting Vehicle (AFV)
Armored Personnel Carrier (APC)
Artillery

These inputs are utilized in calculating the mass of the vehicle and providing base measurements for various subsystems required for each vehicle.

5.2.2 Prime Mover Type

The Vehicle System Inputs window allows the user to determine whether the prime mover will be a turbine or diesel configuration. This data is utilized in determining which baseline data to use in scaling the new vehicle parameters.

5.3 Prime Power Specifications

These parameters are the basic inputs necessary to define the power and efficiency of the prime power. These values include the power of the engine, the rotational speed of the energy storage system, the power of the generator, the density at which energy can be stored, and the efficiency of such a system once installed in the vehicle. Also, the intermediate energy storage systems parameters are defined in this section as well.

5.4 Mobility Specifications

These parameters are basic to defining the power and efficiency of the mobility subsystem. They include the maximum power of the drive system along with the power storage density of that system. Finally, an efficiency for the entire drivetrain is requested.

5.5 Armament Systems

5.5.1 Main Armament

These parameters define the maximum energy of the primary armament, the velocity of that shot, the number of shots immediately available, and the number of shots stowed on board.

5.5.2 Secondary Armament

These parameters define the maximum energy of the secondary armament, the velocity of that shot, the number of shots immediately available, and the number of shots stowed on board. Note: These values are not currently implemented into the power simulation because of a shortage of baseline data.

5.6 Defense System

These parameters define the maximum energy of the defensive armament and the number of shots available. Note: These values are not currently implemented into the power simulation because of a shortage of baseline data.

5.7 Pulsed-Power System

These parameters define the power and efficiency of the pulsed-power charger. They include the energy density of the charging system, the maximum power and efficiency of the charger, the initial state of charge for simulation purposes, and the loss factor associated with maintaining the charge level.

5.8 Auxiliaries

This lower level window interface allows specific inputs to define power demands of the various auxiliary systems. Currently, these demands are totaled and used as one load input on the bus, except for the powered turret and power loader, which have a variable power rating depending on whether or not these systems are on-line immediately preceding a shot. These dynamic loads are taken into account in the POWERSIM simulation.

5.9 Mass Calculation

This section executes an algorithm to determine the mass of the vehicle based on the various parameters set in the interface. At present, very little baseline data exists to scale such calculations, and so the mass should be considered an approximation and may have to be manually set for a simulation.

5.10 File Management

Several file buttons exist in the Vehicle System Inputs window. They include one box in the lower right which can be used to designate the specific filename for the vehicle. When the save button is pressed, the program automatically augments the specific filename with the type of vehicle and the suffix '_veh' to enable one to discern between the various vehicle and other data files. The files from this interface are all saved in the vehicle subdirectory as indicated in Section 2, File Structure. Loading is accomplished by the boxes at the top of the main vehicle window.

5.11 Future Improvements

Currently, the file management scheme does not fully comply with Matlab for Windows 3.1 since many of the filenames are extraordinarily long. This will be revised in the future.

6 Conducting Simulations

6.1 Simulating Mobility Only

Conducting a simulation of the mobility of the vehicle along the mission path is a step necessary for the purposes of studying the demands placed on the entire system by the pulsed-power armament system. For the user, the only necessary inputs are to have the proper data files chosen at the vehicle and mission prompts. The user then simply clicks on each push-button in the Run Simulation GUI in succession. A picture of the GUI is shown in Figure 6-1 for reference. This step must be taken before a full simulation can be accomplished. The simulations were broken up in this manner to allow for a different shot sequence to be run while still using the mission profile.

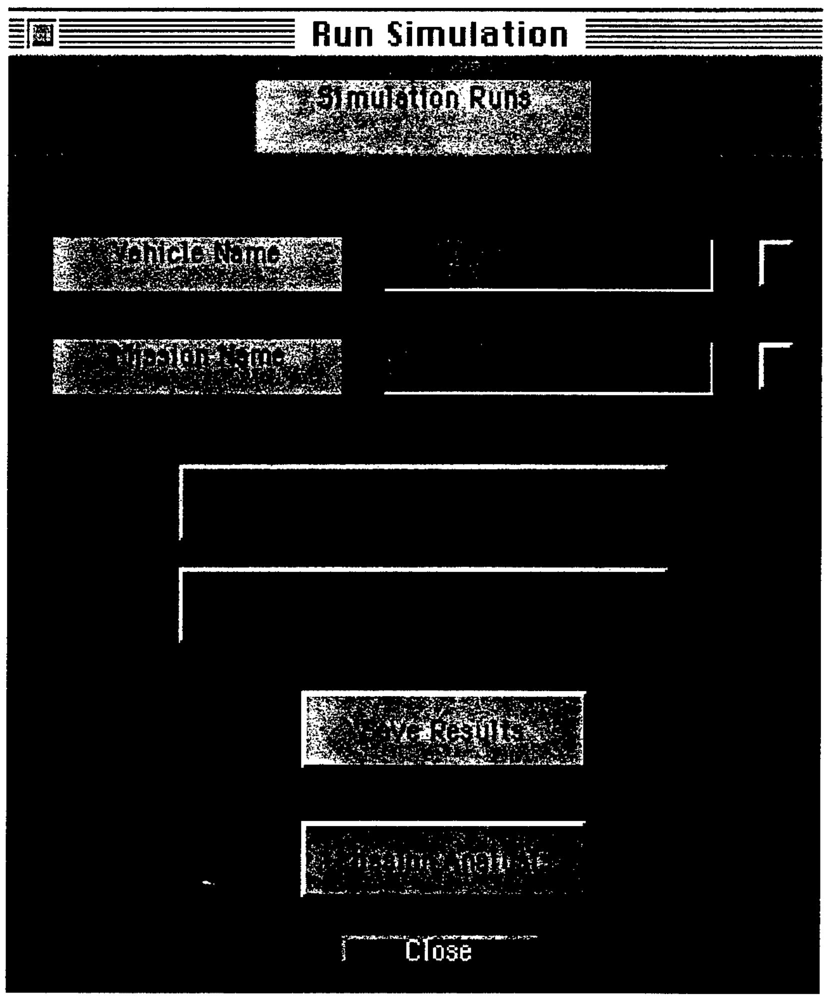

Figure 6-1. Run Simulation GUI

6.2 Simulating Mobility and Shot Sequence Simultaneously

The second step in the simulation process is to conduct a simulation of the launch vehicle with a full shot sequence. Once finished with both simulations, the user should save the results. At this point, the user can select to bring up the analysis tools or return to the mission definition section and redefine another shot sequence for simulation.

6.3 Future Improvements

Currently, the file management scheme does not fully comply with Matlab for Windows 3.1 since many of the filenames are extraordinarily long. This will be revised in the future.

7 Post Processing

7.1 Introduction

The majority of post processing of the results of the simulation take the form of plots of various output parameters versus time. Section 7.2 lists the various plots currently available. The user should note that many different parameter values are available in the Matlab workspace and can be easily plotted if desired. Figure 7.1 shows the main menu for post processing through "Mission Analysis."

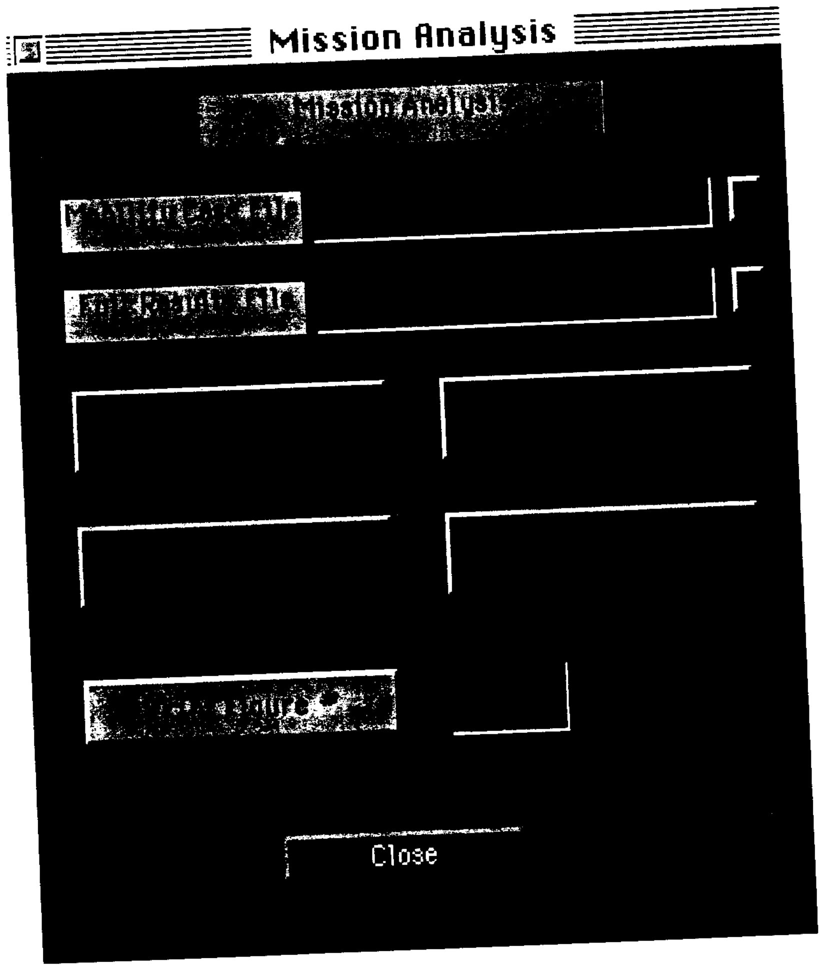

Figure 7-1. Mission Analysis GUI

7.2 Plots Available

7.2.1 Main Systems Plot

Figure 7-2 shows three distinct, but related plots. They include a time-based plot of the overall energy of the system; as well as a plot of the energy demands of several subsystems; and finally, a plot of the intermediate energy stores.

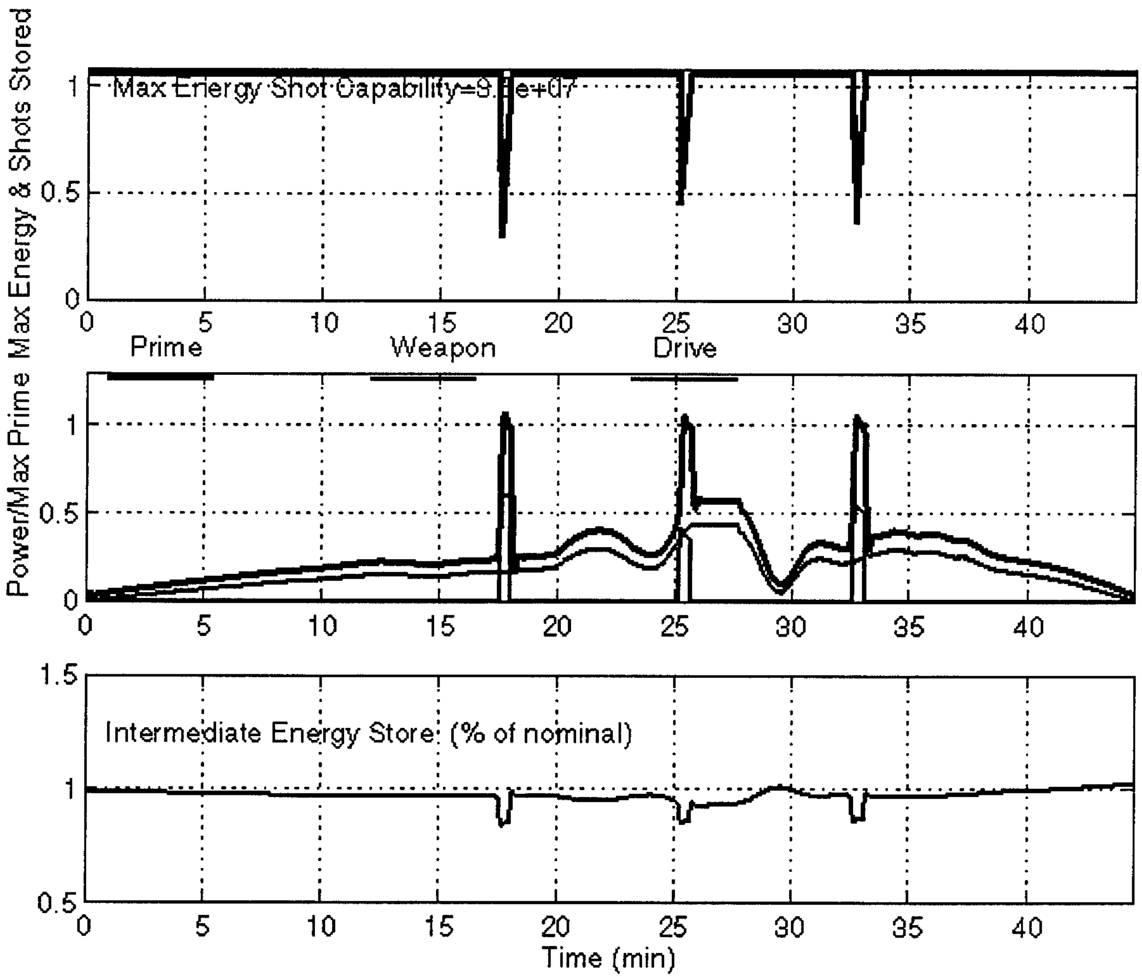

Figure 7-2. Main Systems Plot

7.2.2 Prime Mover Plot (Figure 7-3)

This plot simply shows the power required by the drive system both in the mobility-only simulation as well as in the full shot sequence simulation. In this instance, if the energy available in the system drops below a designated minimum energy, then energy would have been diverted from the drivetrain to enable the shot.

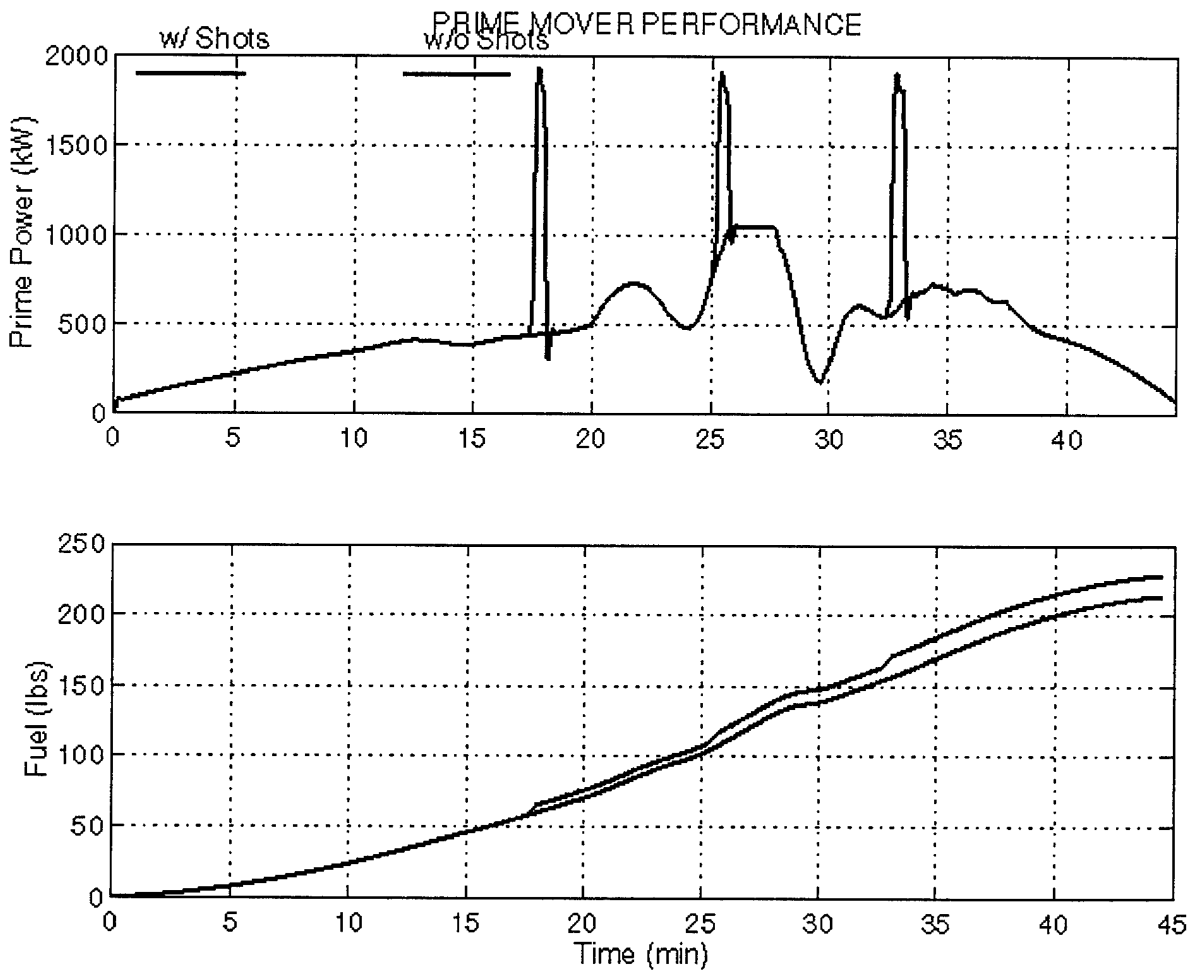

Figure 7-3. Prime Mover Plot

7.2.3 Auxiliaries Plot (Figure 7-4)

This plot simply shows the power required of the auxiliaries both in the mobility-only simulation as well as in the full shot sequence simulation.

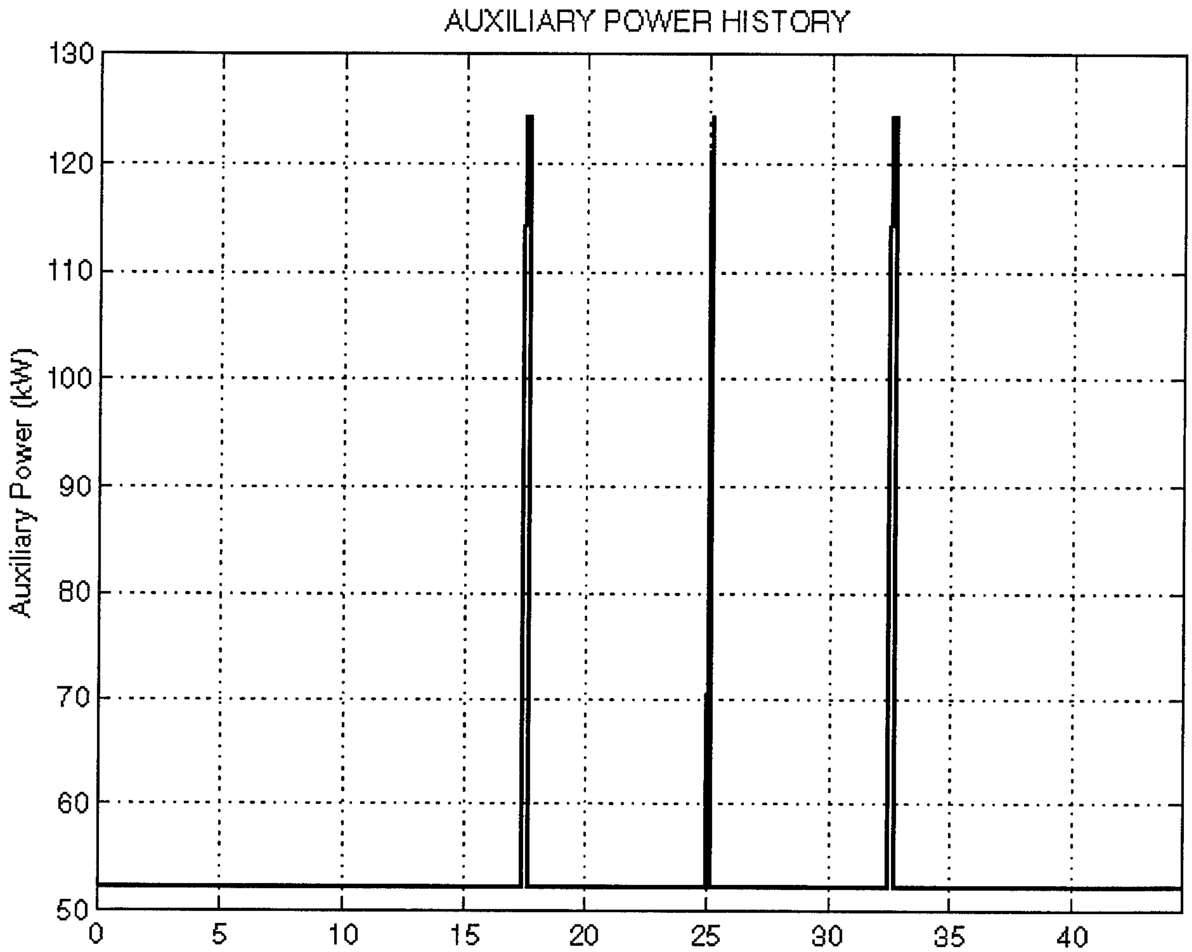

Figure 7-4. Auxiliaries Plot

7.2.4 Shot Analysis Plot (Figure 7-5)

This plot is primarily used to trace the shot delays that occur when the energy available in the main weapon block is less than the shot demands. This results in the shot not occurring for several time steps in the simulation process and indicates a flaw in the vehicle design since the energy balance is too low.

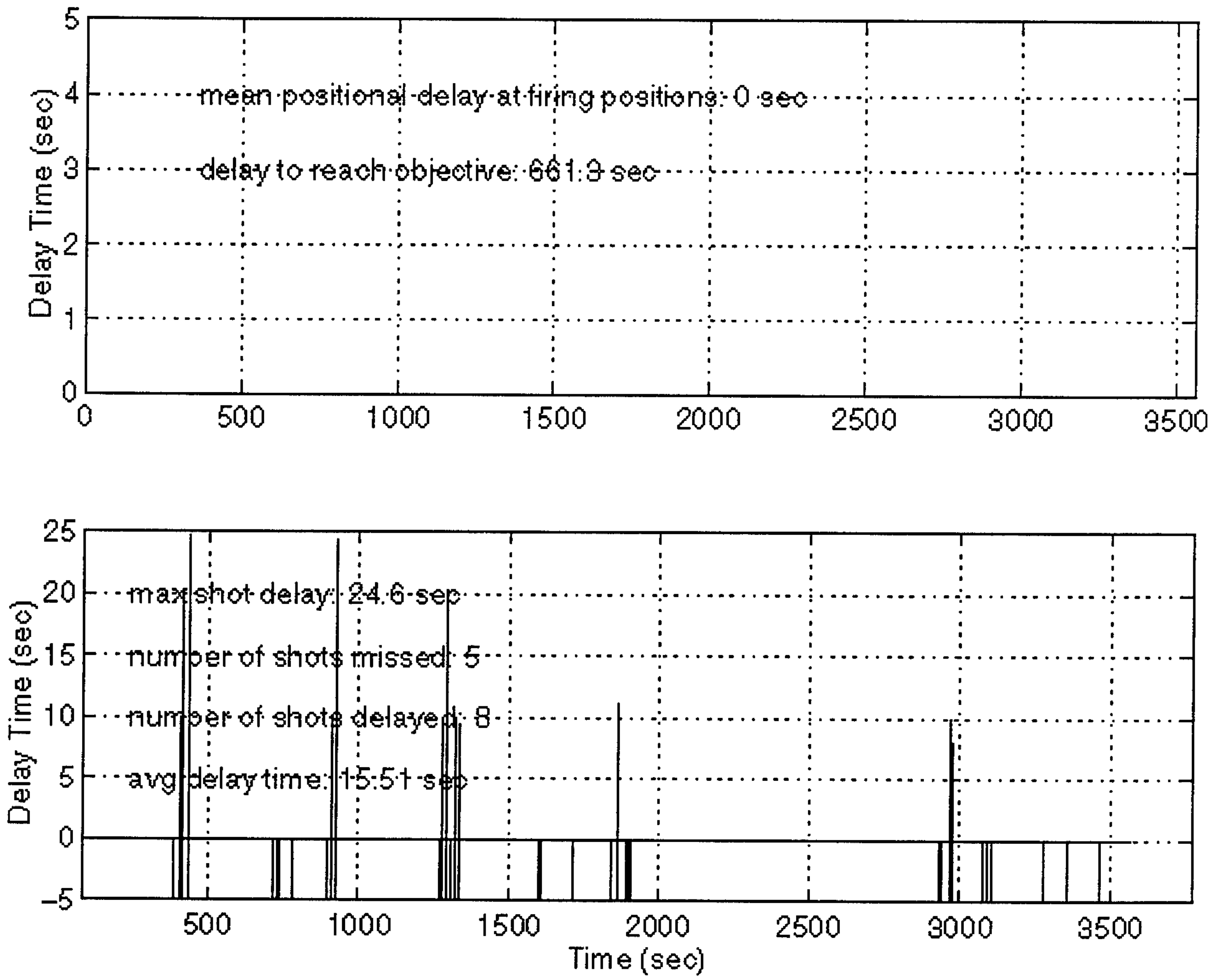

Figure 7-5. Shot Analysis Plot

8 Acknowledgment

This work was supported by the U.S. Army Armament Research, Development and Engineering Center (ARDEC) and the Army Research Laboratory (ARL) under contracts DAAA21-90-D-0009 and DAAA21-93-C-0101 respectively.

Appendix A: Tutorial on Use of Mission Input Program

A.1 Introduction

The steps necessary to produce a mission are perhaps the most complex in the entire simulation process. However, with careful attention, a highly detailed mission definition can be obtained for the position and speed of the vehicle as well as the terrain being encountered. The following section is a step-by-step process to define a simple mission, with each step offering helpful hints to better use the mission input program.

A.1.1 Starting the POWERSIM Program

To start the POWERSIM program, change to the directory with the POWERSIM files in it (the default is psim), then type 'psim' at the prompt. To get to the Mission Input program, use the mouse to click on the button labeled 'Mission Design' under the main menu (see Figure A-1).

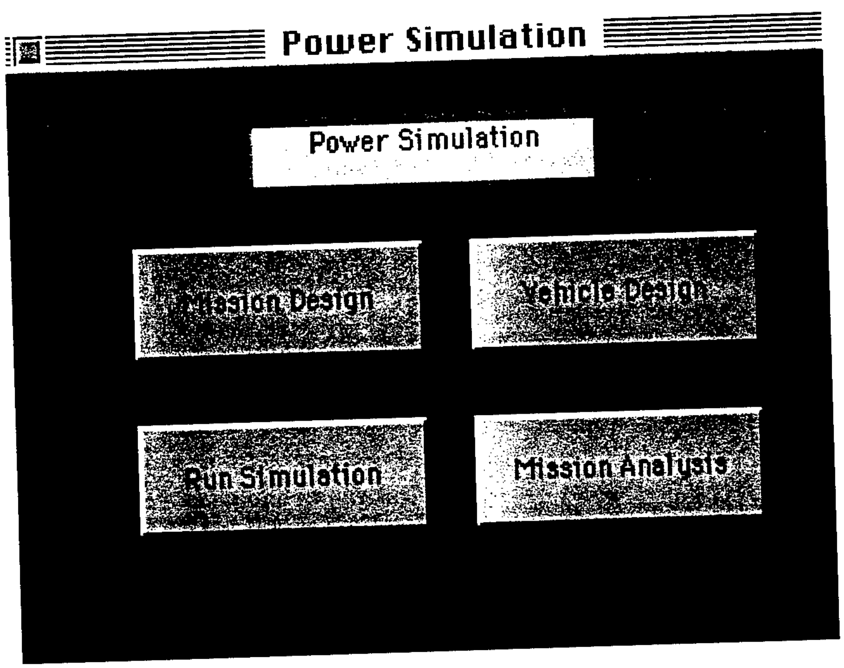

Figure A-1. Main Menu of the Power Simulation

A.2 Mission Input Steps

Step 1 - Input Terrain Filename

In the majority of the test cases, a default (i.e., made-up) terrain map named hlsmooth was used. Be sure to leave the .mat ending off when you type the name of the terrain map file.

Input filename for terrain [hlsmooth]:

<hit return to accept default>

Step 2 - Input Mission Filename

This is basically self-explanatory. One item to note is that there is currently no check for files previously designed with this filename, so the user must insure that the filename is an original name, unless of course they intend to revise a mission file.

Input filename for mission data files:

tempmiss *<return>*

The terrain plot figure now comes up with the terrain file chosen in the last step. Figure A-2 contains a plot of the default terrain file hlsmooth.mat.

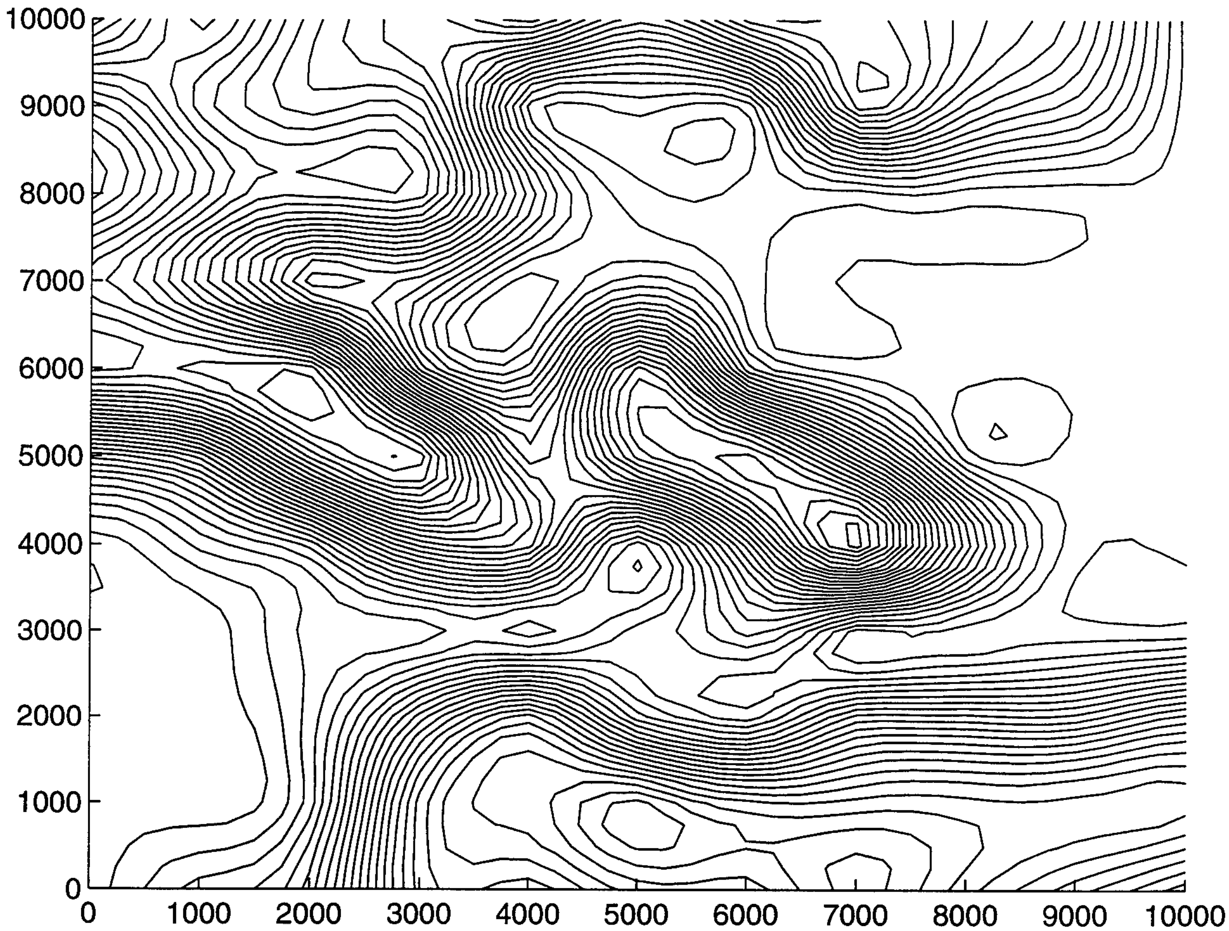

Fig. A-2. Terrain Plot

Step 3 - Retrieve Waypoints from File or Interactively

The user should note that this question is asked even though the file may not exist. If it does exist and the user chooses to use existing waypoints, then program controls switches to Step 9, Mission Firing Sequence. Otherwise, the user should type 'y' and hit enter to acknowledge a new interactive waypoint definition process.

Hit return to load existing waypoints file.

- OR -

Type a "y" and ENTER to interactively select waypoints on the contour map:

y *<return>*

Step 4 - Beginning and End Points

The user should click on a beginning point and an end point with the mouse. The Matlab command window will be showing:

Click mouse on the start and end points for the whole mission:

After the user clicks on both points, the figure is redrawn with the points now appearing.

Step 5 - Waypoint Definitions

Besides the beginning and end points, the user can also select up to 25 waypoints in between. It is recommended that the user select at least 2 waypoints so that the interpolations processes utilized in the code will function properly.

The Matlab command window will display:

Click mouse on the start and end points for the whole mission:

Select 2 – 25 intermediate waypoints, and press return when done:

(Keep the contour window on top)

Thus, the user selects 2 - 25 points to serve as waypoints. Two things to note:

1) If you select all 25 points, then do not hit return, as it will cause the program to skip the next step. Matlab automatically senses that you have reached the maximum input.

2) If you select less than 25 points, you must have the mouse's pointer in the terrain map window when you hit return in order for it to register with Matlab.

Step 6 - Waypoint Descriptions

This step in the mission definition is the most crucial, and may require redefinition of the mission if not properly accomplished. Basically, for each waypoint entered in the last section, several parameters about the vehicle must be entered, including the current velocity, the time since the last waypoint, and the direction of travel.

The Matlab command window will be showing:

You are now at the latest waypoint marked in yellow
Distance from last waypoint is 2.91899 km.
Elevation change since last waypoint is -25.1569 m.
Mission time at last waypoint: 0 min.
Velocity at last waypoint: 0 km/hr.

Input vehicle velocity at this point: [vlast] km/hr

The user should now enter a velocity. Although a default is given for the last velocity, an error will result at the first waypoint if a new velocity is not entered. Likewise, the user should insure that the last waypoint has zero velocity to allow the interpolation routines to run properly. Thus, for the first instance, the user could enter 15, meaning 15 km/hr.

15 *<return>*

The Matlab command window will be showing:

Your average speed over this interval is 7.5 km/hr.
Some sample transit times at average speeds:

@ 5 km/hr	35.0279 min.
@ 10 km/hr	17.5139 min.
@ 20 km/hr	8.75696 min.
@ 30 km/hr	5.83798 min.
@ 40 km/hr	4.37848 min.
@ 50 km/hr	3.50279 min.
@ 7.5 km/hr	23.3519 min.

Input time from last waypoint to this one: (minutes) [vavg value]

Here, the program is prompting the user to enter a time since the last waypoint. The default in this case would be the time associated with the average value, shown on the bottom line of the list. Hitting return here will select the time associated with the average velocity. However, if the user had intended for the vehicle to accelerate rather quickly up to speed and then hold constant, a shorter time based on the given times should be entered. In this instance, 15 minutes is chosen, indicating an average velocity higher than the average of the velocity at previous and current waypoints.

15*<return>*

Next, the user is prompted for a direction of travel on the terrain map figure. In other words, the program is requesting the velocity vector on which the vehicle is pointed at that waypoint. This allows an interpolation of the track rather than having sharp corners, unless such corners are intended. In that instance, the velocity should be fairly low at the waypoint so that the direction of travel could be directly to the next point rather than curved around. The Matlab command window will show:

Click on point in direction of travel:

The process then continues for the remaining points. The final point requires the velocity be stated as zero rather than accepting the default velocity of that of the previous waypoint. For any waypoint with a zero velocity (stop point), a stationary time is requested. Hitting return at this point accepts the default two minutes, or a different value can be entered, as can be seen in the Matlab window:

You are now at the latest waypoint marked in yellow
Distance from last waypoint is 1.8119 km.
Elevation change since last waypoint is -14.8824 m.
Mission time at last waypoint: 23.2877 min.
Velocity at last waypoint: 15 km/hr.

Input vehicle velocity at this point: [vlast] km/hr

0*<return>*

Input time stationary at this position: [2] min.

<return>

This completes the waypoint definition section of the mission input program. The next section will be used to roughly define the vehicle so that accelerations can be determined.

Step 7 - Vehicle Inputs

(The Matlab command window now shows:)

Total mission time 2266.98 sec or 37.7829 min.

Now computing path distances and slopes.

{ this takes a while }

Plotting, be patient.

Ready to compute power? [return=yes] :

<return>

Input estimated mass of vehicle (kg) [50000] :

<return>

Estimated frontal area = 7.560000e+01 ft^2
Typical frontal areas for other vehicles:
M1A1 80 ft2
M2/M3 90 ft2
LAV 47 ft2
M109 83 ft2

Input track separation in meters [3]:

<return>

Step 8 - Firing Sequence

You will now select firing positions, specifying the number of rounds fired at that point and the time between those shots.

The Matlab command window shows the following request from the program, even at the first instance:

```
click on next firing position:

Stored Energy for this shot sequence? [10] (MJ)
```

<return>

```
How many shots taken here? [1]:
```

<return>

```
More firing positions? [return=yes]:
```

<return>

If multiple shots are selected at one point, a time between shots is requested as well

```
click on next firing position:

Stored Energy for this shot sequence? [10] (MJ)
```

<return>

```
How many shots taken here? [1]:
```

3*<return>*

```
Time between shots (seconds) [5]:
```

<return>

```
More firing positions? [return=yes]:
```

n*<return>*

Step 9 - Mission Input Program Complete

The Matlab command window will show:

```
Mission Input Program complete.
```

Also, a dialog box will come on screen that will close all the figures associated with mission input program.

Appendix B: Tutorial on use of Vehicle Systems Input GUI

B.1 Introduction

The majority of this Graphical User Interface (GUI) is self-explanatory on screen, but it does help to have some insight about the various nuances of this GUI and the Matlab environment in general. This GUI is broken down into 5 major parameter sections: the prime mover, drive system, armament system, protection system, and the pulsed-power system. At the top and bottom of the window are various buttons and editable text boxes that calculate other parameters such as energy requirements or mass totals, link this GUI to those with higher detail in a specific area, and those interfaces used for file management. A scaled down version of the GUI is shown in Figure B-1.

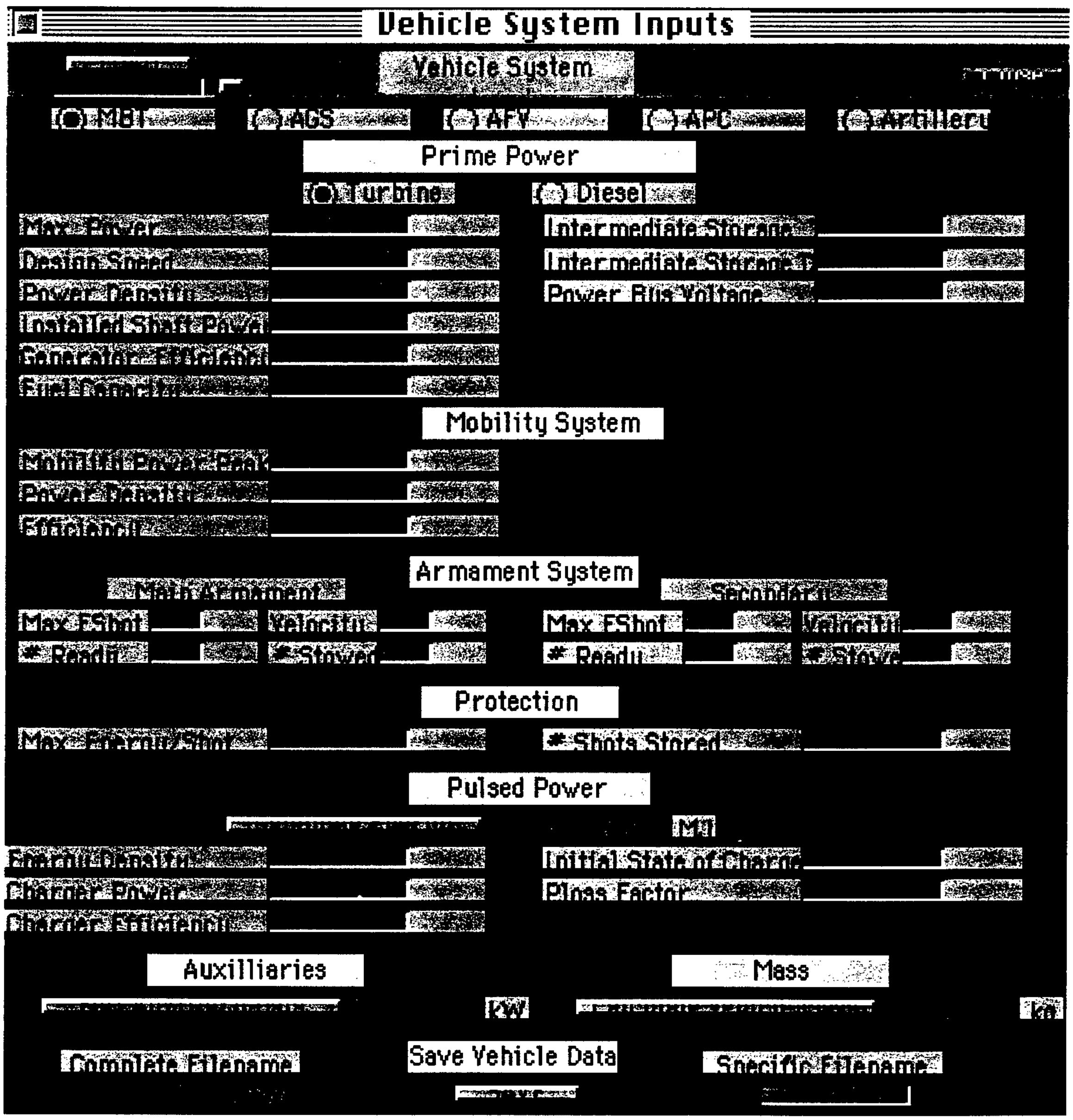

Figure B-1. Vehicle Systems Input GUI Thumbnail

B.1.1 Starting the POWERSIM Program

To start the POWERSIM program, change to the directory with the powersim files in it (the default is psim), then type 'psim' at the prompt. To get to the Vehicle Systems Input GUI, use the mouse to click on the button labeled 'Vehicle Design' under the main menu.

B.2 Variable Descriptions

B.2.1 Prime Power

Maximum Prime Power	(MW)	- rated power of prime mover
Design Speed	(rpm)	- design speed of prime mover
Power Density	(kW/kg)	- power per unit kg of prime mover
Installed Shaft Power	(%)	- subtracted power loss with shaft installed
Generator Efficiency	(0-1)	- efficiency of generator between 0 and 1
Fuel Capacity	(gal)	- amount of fuel for just the prime mover on board
Intermediate Storage	(MJ)	- total capacity of the energy storage device driven by the generator via the power bus
Intermediate Storage Density	(kJ/kg)	- mass required for the device versus amount of energy stored
Power Bus Voltage	(V)	- nominal voltage for the power bus

B.2.2 Drive System

Mobility Power Peak	(MW)	- Maximum Drive Power for this vehicle
Power Density	(kW/kg)	- amount of drive power available per kilogram of system equipment (motors, power electronics...)
Efficiency	(0-1)	- efficiency of drivetrain (from main bus to sprocket or wheel)

B.2.3 Armament

B.2.3.1 Main Armament

Max Eshot	(MW) - Maximum energy per shot
Velocity	(km/s) - Maximum velocity of a shot (not used yet)
# Ready	(#) - Number of shots available in the energy storage device currently
# stowed	(#) - Number of shots available on board (for weight calculation)

B.2.3.2 Secondary

Max Eshot	(MW) - Maximum energy per shot
Velocity	(km/s) - Maximum velocity of a shot
# Ready	(#) - Number of shots available in the energy storage device currently
# stowed	(#) - Number of shots available on board

B.2.4 Protection

Max Eshot	(MW) - Maximum energy per shot (stored energy in pulsed power device)
# shots stored	(#) - Number of defense shots available

B.2.5 Pulsed Power

Energy Density	(kJ/kg) - amount of useable energy per unit kg of pulsed-power energy storage
Charger Power	(MW) - Maximum Pulsed-Power Charger Power
Efficiency	(0-1) - Efficiency of the charger (main bus to energy stored)
Initial State of Charge	(0-1) - Fraction of energy capacity at start of simulation
Ploss Factor	(#) - Factor for windage / bearing loss or current leakage (not used yet)

B.2.6 Auxiliaries

The value shown in the window is the total of the subsystems considered constant for current uses. Clicking the edit button brings up a separate window where any of these values, as well as those of the powered turret and automatic loader, can be changed. Figure B-2 shows the Auxiliary Systems GUI.

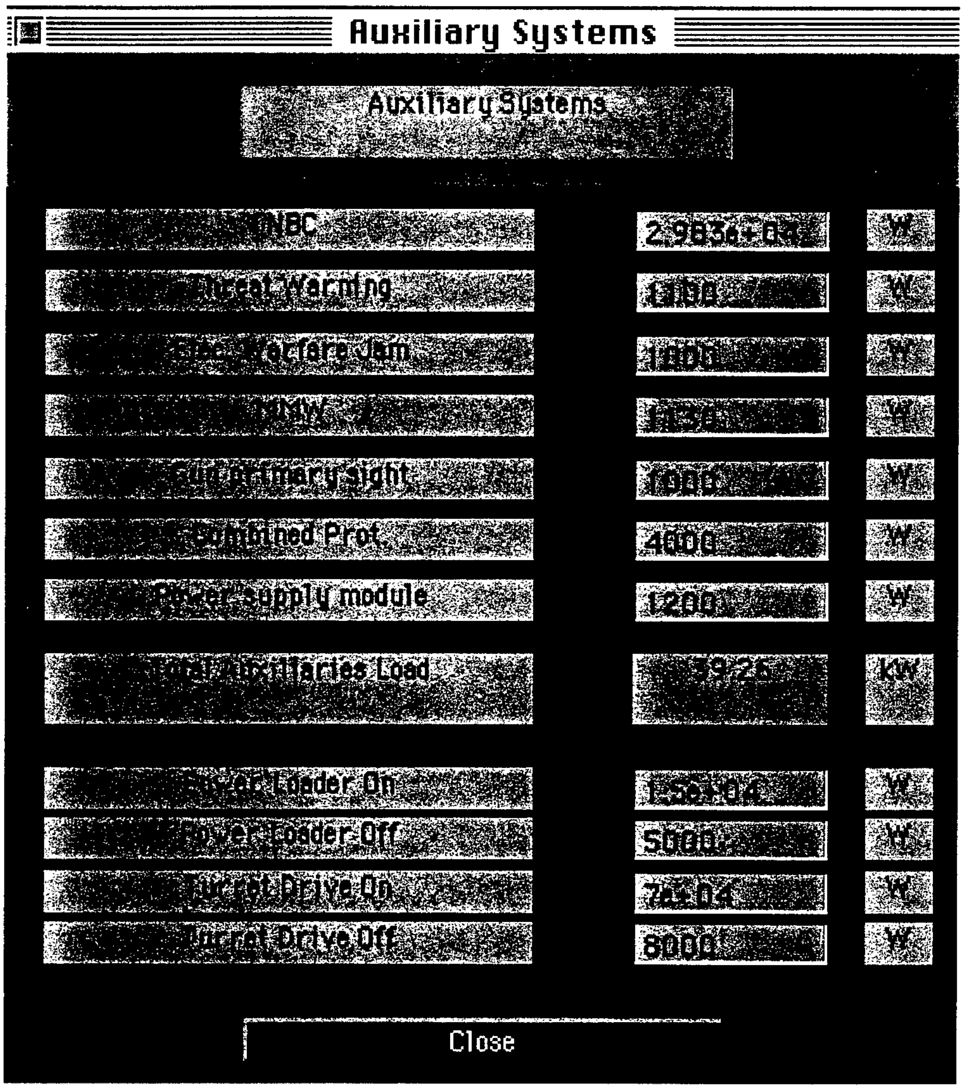

Figure B-2. Auxiliary Systems GUI

B.2.6.1 Constant Auxiliary Parameters

NBC (W)	- Auxiliary Power devoted to Nuclear, Biological, Chemical Warfare protection
Threat Warning (W)	- Auxiliary Power devoted to the Threat Warning System
Elec. Warfare (W)	- Auxiliary Power devoted to Electronic Warfare Jamming
MMW (W)	- Auxiliary Power devoted to the Millimeter Wave radar
Gun Primary Sight (W)	- Auxiliary Power devoted to the Gun's Primary Sight
Combined Prot .(W)	- Auxiliary Power devoted to Combined Protection System
Power Supply Module (W)	- Auxiliary Power devoted to the Power Supply Module

B.2.6.2 Variable Auxiliary Parameters

Power Loader On (W)	- Auxiliary Power Demand of Autoloader with Power On
Power Loader Off(W)	- Auxiliary Power Demand of Autoloader with Power Off
Turret Drive On (W)	- Auxiliary Power Demand of Turret with Power On
Turret Drive Off (W)	- Auxiliary Power Demand of Turret with Power Off

B.2.7 Mass Calculation (Back to Figure B-1)

The value shown in this text box is the mass as calculated by current procedures. This area is certain to undergo radical changes as soon as baseline data becomes available. Current procedure scales from an M1 by subtracting the weight associated with prime power, transmission, and ammunition and then adds estimated weights for new prime power, pulsed power, drive system and ammunition. The resulting figure can be overridden by the user with better mass estimates from other sources. This calculation method will be improved in the future.

B.3 File Management

Six areas should be noted in the Vehicle System Inputs window in regard to file management. They include the three boxes in the upper left, and the three boxes across the bottom.

B.3.1 Save File

In order to save a vehicle file, the first step is to type a specific name for this vehicle in the specific filename box in the lower right corner. Upon doing this, the user will notice that the complete filename box in the lower left, and the load box in the upper left have changed appropriately. The complete filename shows the actual filename the vehicle will be saved under. This filename consists of the vehicle type, appended by the specific filename, and then appended by '_veh' for use in determining whether a file describes a vehicle or a mission. If the specific filename was incorrect, or the user simply wished to change it, it needs only be changed in the lower right corner box. Once the user is content with that vehicle name, then the 'SAVE' button in the bottom middle should be clicked in order to actually save the program.

B.3.2 Load File

If the user desires to load a different vehicle into the GUI, there are three areas to look at in the upper right, including an editable box with the name of the vehicle file to be loaded, the button above that called 'LOAD' that actually loads that file, and a button to the side labeled '...' which brings up a second Matlab GUI simply to determine a filename. Once a filename is chosen by either of the two methods, then the 'LOAD' button must be pressed to actually load the file.

Appendix C: Tutorial on use of Simulation GUI

C.1 Introduction

This appendix describes in detail how to actually conduct a simulation and what occurs in this process. The Run Simulation GUI seems quite straightforward, since there are only a few pushbuttons and or editable-text boxes, but it is beneficial to the user to note some subtleties noticed in the Matlab environment (see Figure C-1). These items will be approached in the following sections.

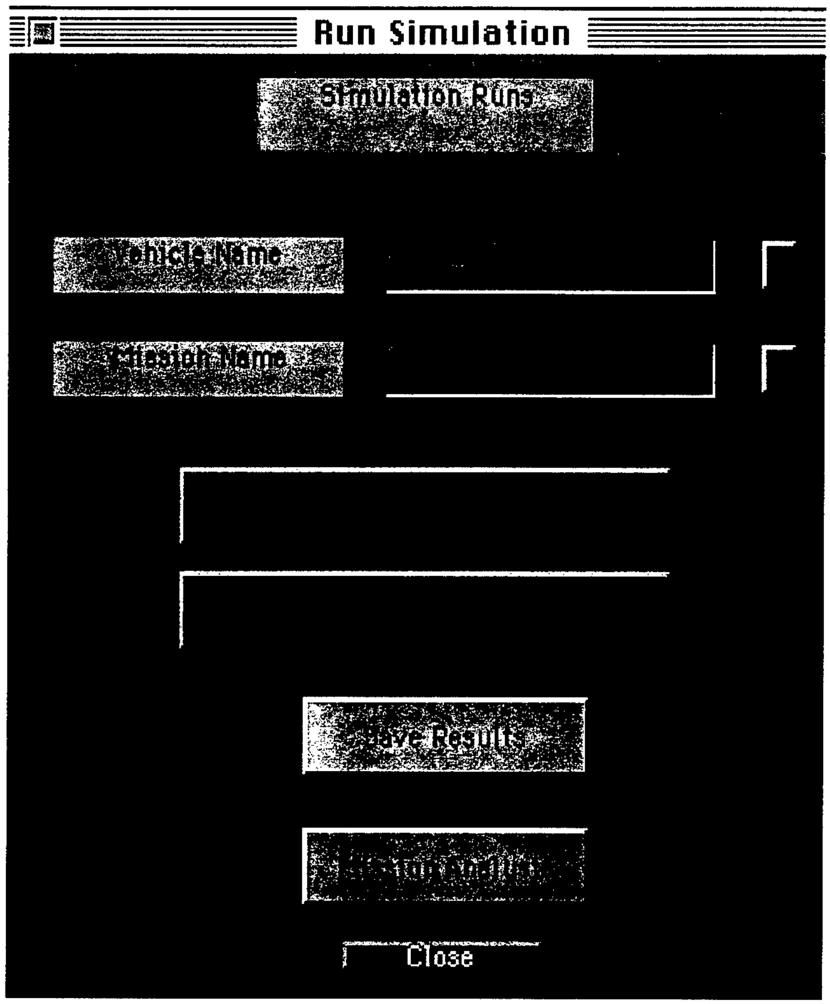

Figure C-1. Run Simulation GUI

C.1.1 Starting the POWERSIM Program

To start the POWERSIM program, change to the directory with the powersim files in it (the default is psim), then type 'psim' at the prompt. To get to the Simulation GUI, use the mouse to click on the button labeled 'Run Simulation' under the main menu.

C.2 Vehicle File

This interface allows the user to decide the vehicle with which to conduct the simulation. As with the mission file determination, the user can either type the vehicle name in the provided space or click the button labeled '...' to bring up a secondary GUI that actually displays the files available on the disk from which to choose.

C.3 Mission Name

This interface allows the user to decide the mission with which to conduct the simulation. The user can type the mission name in the provided space or click the button labeled '...' to bring up a secondary GUI that actually displays the files available on the disk from which to choose.

C.4 Run Mobility Simulation

This button initializes the workspace and conducts the simulation of just the mobility. This step is necessary to determine baseline data which will be used in post-processing. The breakdown between the two steps of simulation are incorporated to allow later revision of the shot sequences and conduct further simulations without having to reconstruct the baseline data. On UNIX systems, the simulation process takes about 1 minute real-time to simulate 7 minutes worth of mission, with or without a shot sequence included.

Tip: Only click this button once rather than double click like in standard windows. Double clicking is interpreted by Matlab as two distinct clicks, and therefore the simulation will be run twice. Although no erroneous data will appear, it will take twice as long, which is significant for longer missions or slower machines. An upgrade is being developed by Matlab that will alleviate this problem and will be implemented in POWERSIM when available.

C.5 Run Full Simulation

This button re-initializes the workspace and conducts the full simulation of the vehicle power systems.

Tip: Again, only click this button once rather than double click like in standard windows. Double clicking is interpreted by Matlab as two distinct clicks, and therefore the simulation will be run twice. Although no erroneous data will appear, it will take twice as long, which is significant for longer missions or slower machines. . An upgrade is being developed by Matlab that will alleviate this problem and will be implemented in POWERSIM when available.

C.6 Save Results

Choosing this option simply tells the program that the user is satisfied with the current runs and wants to save them. The filename associated with this is a conglomeration of the vehicle name followed by the mission name followed by the word '_results'. The reason this step was included was due to the lack of file overwrite messages. In other words, if another mission with same name and vehicle were already run and saved, choosing this button will write the current results directly over the old results since the name of the vehicle and mission are the same. Although future improvements may allow such niceties, a current workaround would be to simply return to the host environment and manually copy the results of the old file to a new name if the user wanted to save the old results.

C.7 Mission Analysis

Selecting this button links the user to the next interface window that allows him to analyze the parameters output by the simulation. This interface is described in detail in Appendix D.

Appendix D: Tutorial on use of Mission Analysis GUI

D.1 Introduction

The majority of the post-processing of the simulation results in the form of plots of various output parameters versus time. Four plots are predefined and made available to the user (see Figure D-1). These plots are listed in the next few sections. The user should note, however, that many different parameter values are available in the Matlab workspace and can be easily plotted if desired. Consult the Matlab documentation for plotting these other parameters. A list of parameter definitions is included in Appendix E.

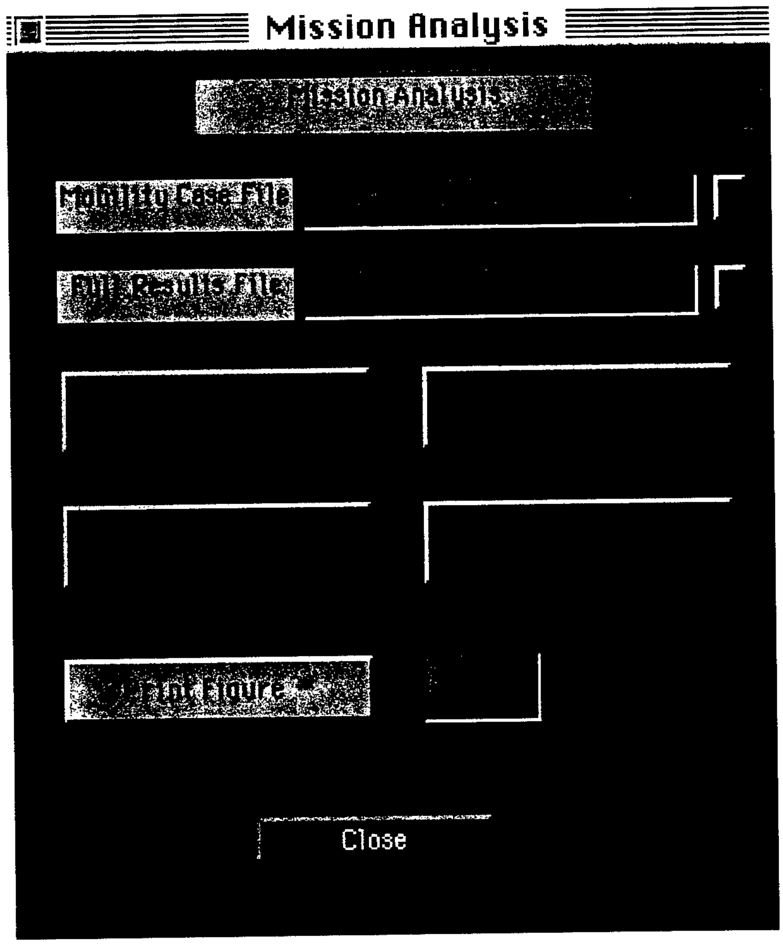

Figure D-1. Mission Analysis GUI

D.1.1 Starting the POWERSIM Program

To start the POWERSIM program, change to the directory with the powersim files in it (the default is psim), then type 'psim' at the prompt. To get to the Mission Analysis GUI, use the mouse to click on the button labeled 'Mission Analysis' under the main menu.

D.2 Main Systems Plot

The Main Systems Plot shows several plots over three separate axes (see Figure D-2). In each plot, time is the x-axis while the y-axis value changes from energy store (in MJ) for the first plot to power consumed by various systems (in W) for the second plot, and finally intermediate energy storage as a percentage of maximum energy store.

This figure window has several key points to note. First, the top line represents the overall energy available, so it is useful to consider how many shots are available versus drive power desired and see if they have driven the energy in the energy stores to zero. Likewise, if power were diverted from the mobility subsystems to be used in the weapon systems, how much power was taken and what was its effect on the energy storage as well as the vehicle dynamics.

The second plot shows the power relationship between several subsystems throughout the time history of the mission. In the color plots on screen, the user should notice that the green line is the amount of power available from the prime mover, whereas the red line is the power demand of the charger and the blue line is the power demand of the drivetrain from the energy stores. This further demonstrates how the vehicle was utilizing power when, looking at the upper plot, the energy stores came down below a critical level.

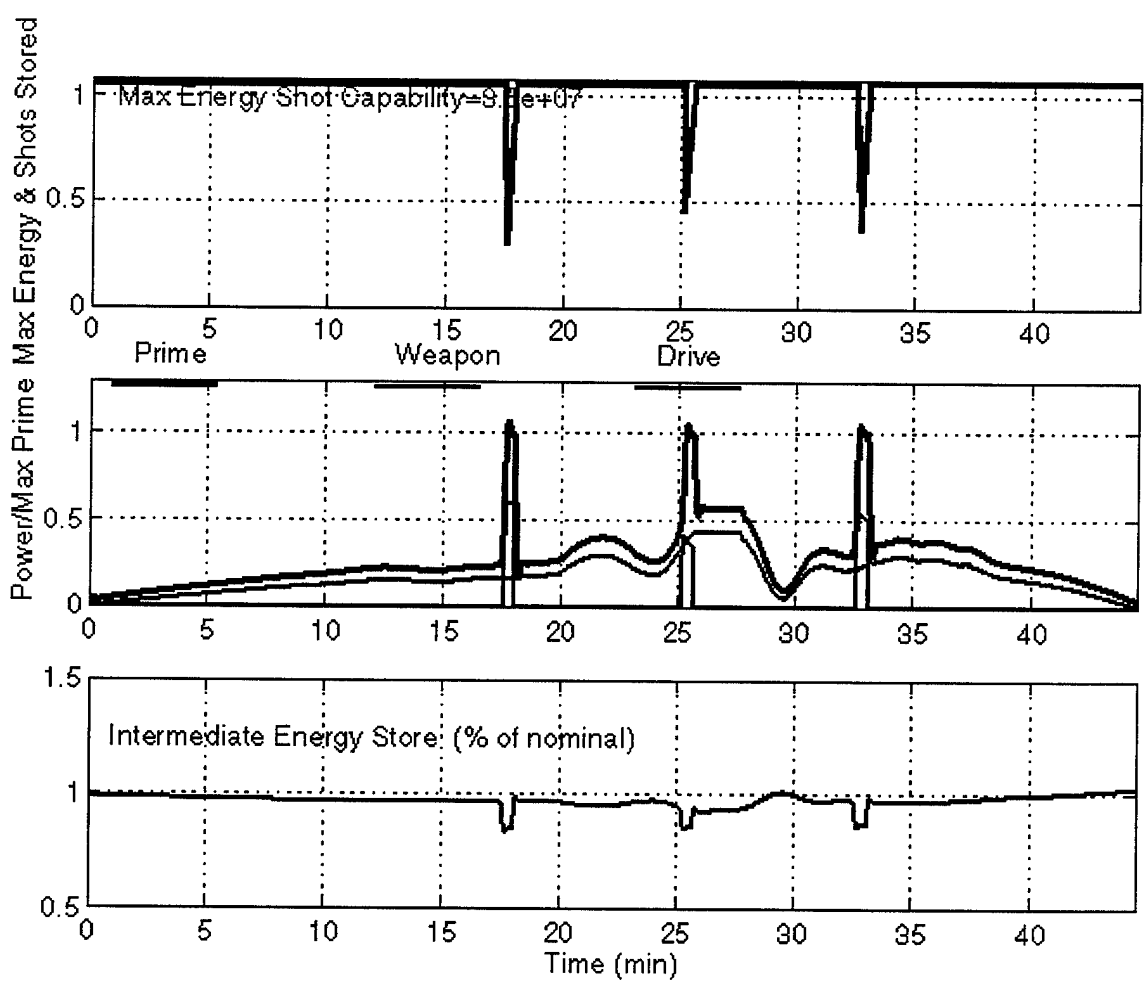

Figure D-2. Main Systems Plot

D.3 Prime Mover Plot (Figure D-3)

This plot simply shows the power required by the drive system both in the mobility-only simulation as well as in the full shot sequence simulation. In this instance, if the energy available in the system drops below a designated minimum energy, then energy would have been diverted from the drivetrain to enable the shot. This plot would demonstrate that by showing that the line from the plot of the full simulation case drops below that of the mobility only.

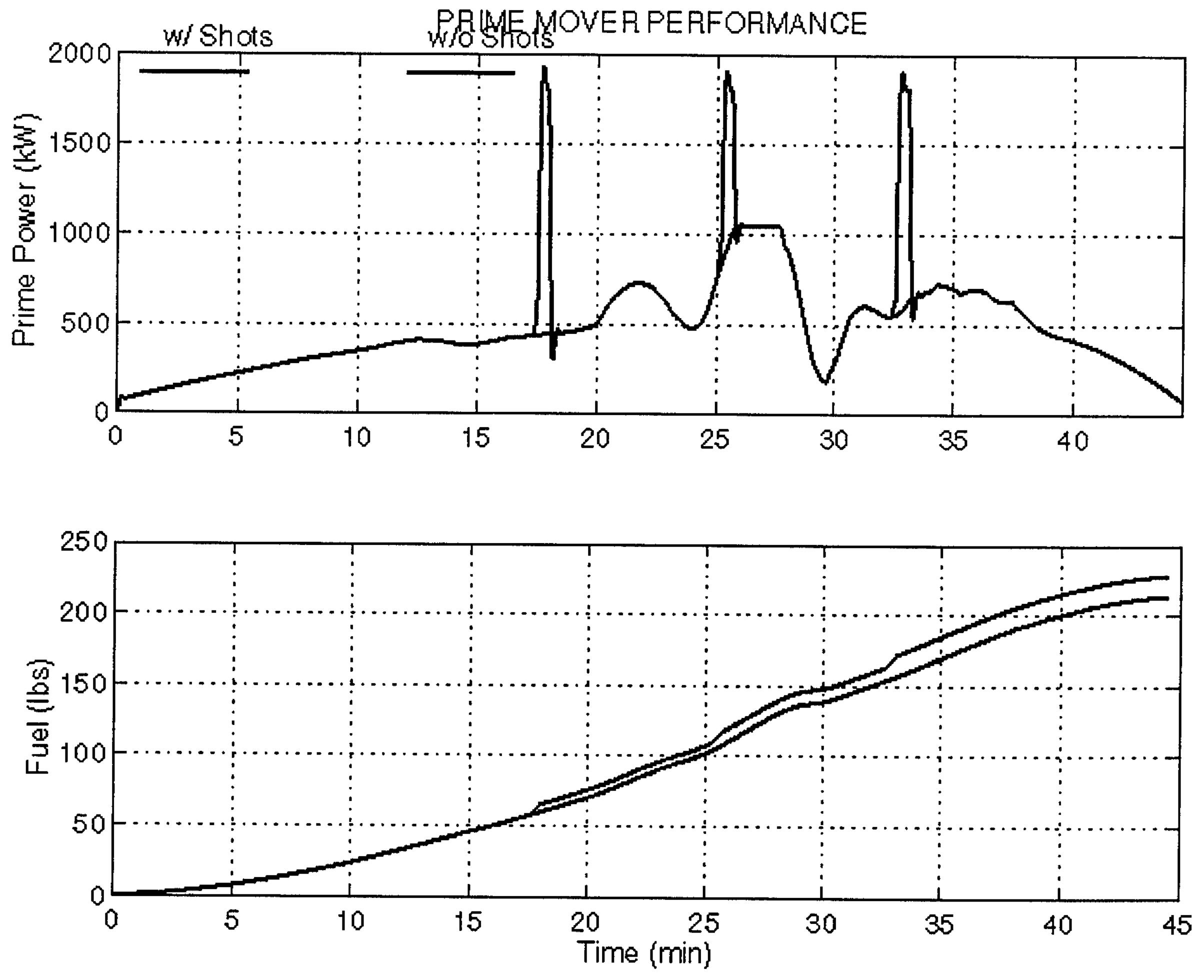

Figure D-3. Prime Mover Plot

D.4 Auxiliaries Plot (Figure D-4)

This plot simply shows the power required of the auxiliaries both in the mobility-only simulation as well as in the full shot sequence simulation. Currently, these values are simply constants except when a shot is being prepared, and in that case they are at some higher level for five seconds to enable the power loader and turret drive to aim the weapon.

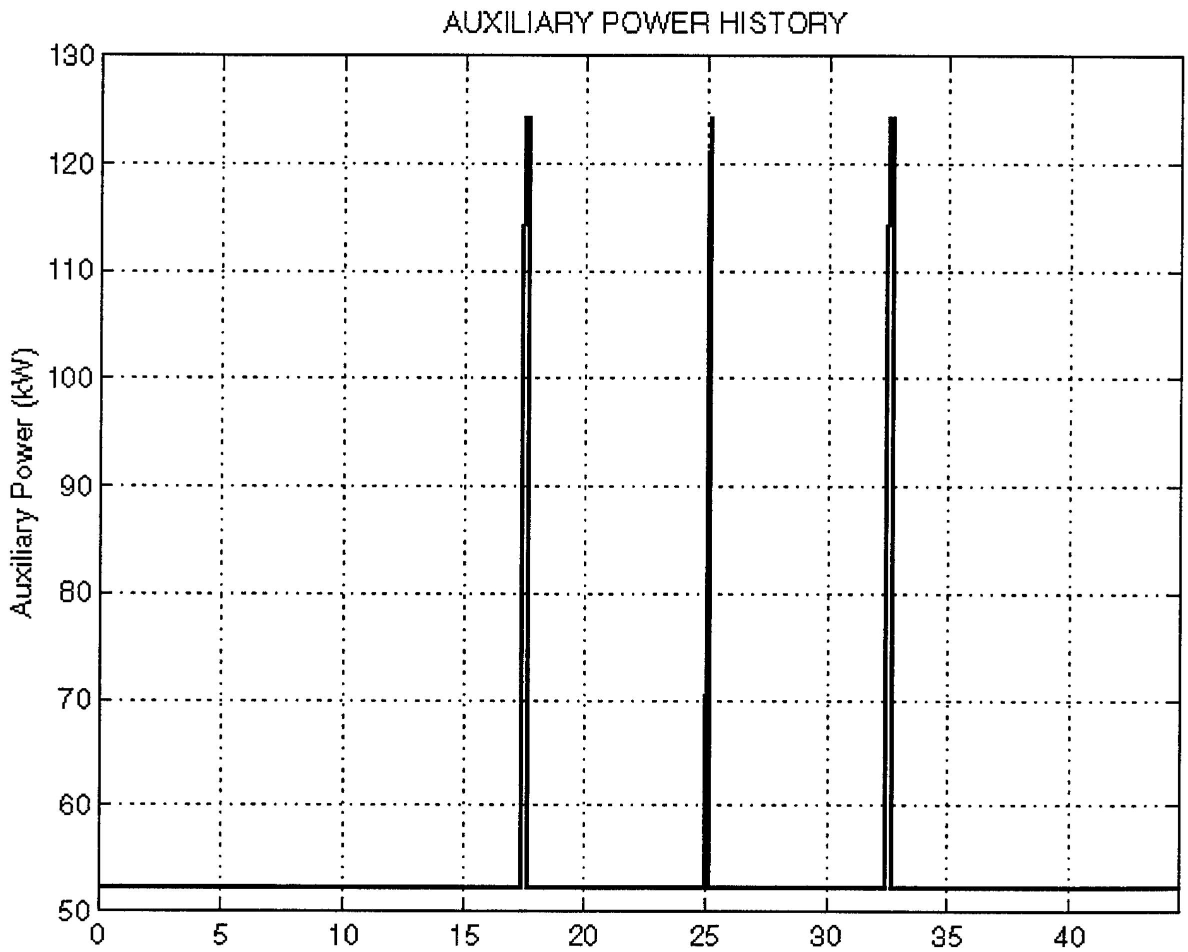

Figure D-4. Auxiliaries Plot

D.5 Shot Analysis Plot

This plot is primarily used to trace the shot delays that occur when the energy available in the main weapon block is less than the energy demand of the shot. This results in the shot not occurring for several time steps in the simulation process, until that point that enough energy is restored to the energy storage system, either by diverting mobility power or just letting the charger run without shooting. This plot thus indicates flaws in the vehicle design since the energy balance was too low to handle the desired mission. Shot delays are evident in this plot by any line extending above the bold horizontal line, with their height indicative of their respective delay (see Figure D-5).

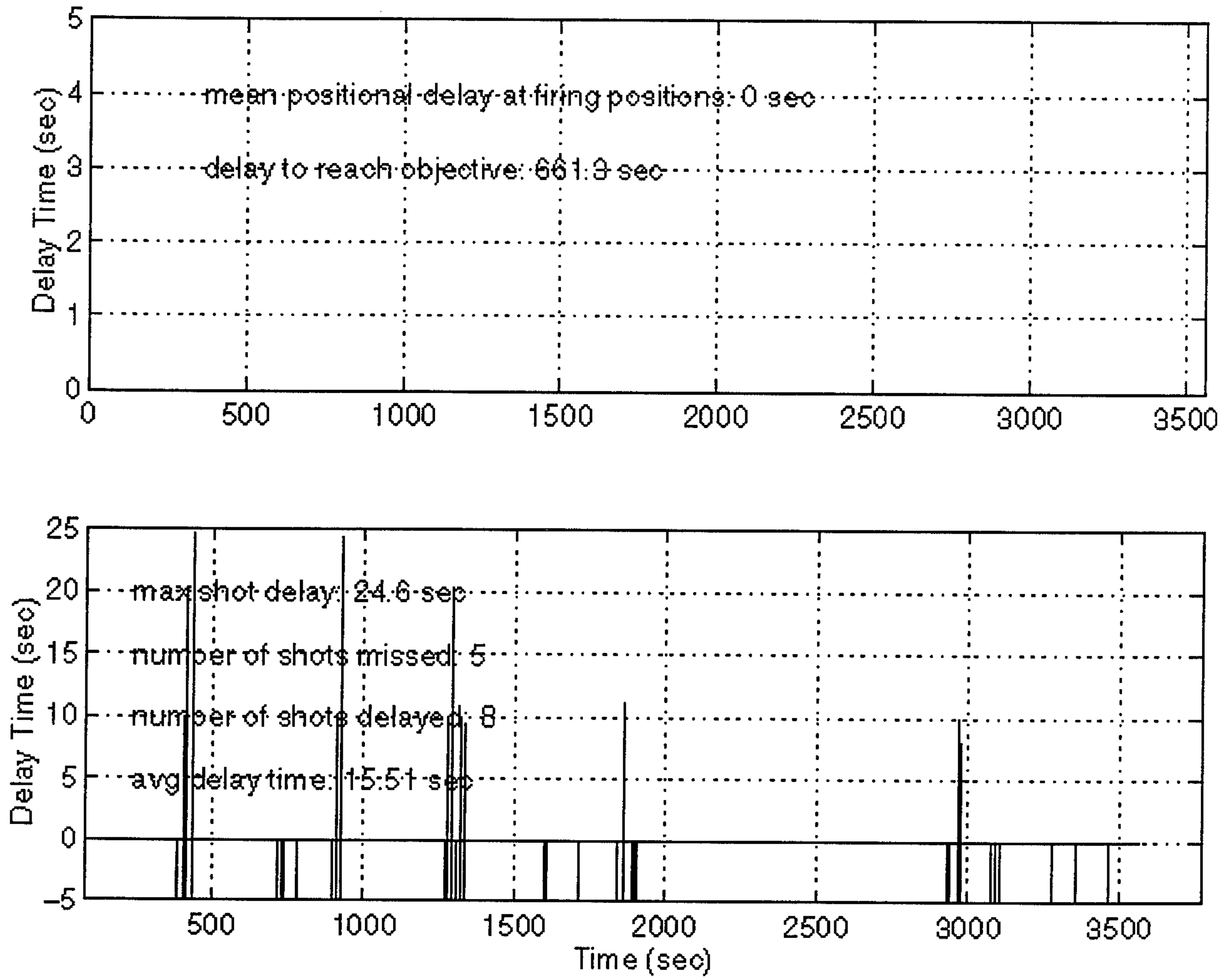

Figure D-5. Shot Analysis Plot

Appendix E: Simulation Output Parameter Definitions

E.1 Introduction

This section simply lists and explains each value output at each timestep by the simulation.

E.2 Parameter List

t	(sec)	Time Counter
x	(m)	Position of Vehicle
v	(m/s)	Velocity of Vehicle
e	(J)	Energy Available in Energy Store
ee	(J)	Current Shot Energy Desired
trig	(integer)	Trigger signal
DELAY	(integer)	Shot Delay Tracker
SHOT_DELAY	(sec)	Shot Delay Tracker
SHOT_OVERLAP	(int.)	Counter for Shots Delayed over Next Shot
TSHOT	(sec)	Actual Time of Shots
SSHOT	(m)	Actual Position of Shots
Te	(N)	Actual Torque Output of Generator
vgen	(V)	Actual Voltage Output of Generator
igen	(A)	Actual Current Output of Generator
iload	(A)	Current Load
Edc	(J)	Actual Energy Available on Bus
Vdc	(V)	Bus Voltage
Paux	(W)	Actual Dynamic Power Demand from Auxiliaries
PPload	(W)	Pulsed Power Energy Load
Tic	(N)	Torque of the IC Engine
Pic	(W)	Power of the IC Engine
recpid	(non-dim.)	Generator PID output
fuel	(gal)	Fuel consumed
throt	(non-dim.)	Actual Throttle Output after Lag
throt_com	(non-dim.)	Actual Throttle Output
spd	(rad/s)	Generator Shaft Speed
Fwind	(N)	Aerodynamic Forces on Vehicle
werr	(non-dim.)	Error tracker in PID driver controller
Pdr1	(W)	Actual Drive Power
Pdr3	(W)	Desired Throttle to engine controller
Pdrive	(W)	Drive Power
Pcharge	(W)	Actual Charger Output

APPENDIX F: Matlab Workspace Definition

F.1 Program Control Variables

fignum (integer array) index of different figures on the screen
- each element of the array indicates what figure is what GUI, since they won't always be open in the same order
  1 - Main Menu's figure number
  2 - Vehicle System Input's figure number
  3 - Run Simulation's figure number
  4 - Mission Analysis's figure number
  5 - Dialogbox's figure number
  6 - Edit Auxiliaries's figure number
  7 - Main Systems Plot's figure number
  8 - Prime Mover Plot's figure number
  9 - Auxiliaries Plot's figure number
  10 - Analysis Plot's figure number
  11 - Terrain Map's figure number
  12 - Mission Input Plot's figure number

There are 1 or 2 other GUI's that have not had this simple housekeeping scheme implemented

homedir (string)	path of psim.m, assumed to be prefix to vpath and mpath as well until otherwise set.
vpath (string)	path to vehicle directory
vehname (string)	name of vehicle in the form [veh_type,'_',specific filename,'_veh']
mpath (string)	path to mission directory
missname (string)	name of mission, associated files have fp, pwr, or wp appended to name
terpath (string)	path to terrain directory
ternane (string)	name of terrain
respath (string)	path to results directory
resname (string)	name of results file
noshotpath (string)	path to no-shot results directory
noshotname (string)	name of no-shot results file

F.2 Variables / Units Definitions and Descriptions

I/D/O* = Independent/Dependent Input or Output, whether the value is specified in the GUI or the GUI defines them based on inputs, or they are output from simulation

**** Units: the following units are assumed on the global variables used in this function:

Variable Name	Units	I/D/O*	Description
Pprime_max	(W)	I	Prime Power Max, according to manufacturer
ref_speed	(rad/s)	I	Generator Shaft Speed
prime_loss	(%)	I	Installed Shaft Power Loss, reduces Pprime_max
PPRIME_MAX	(W)	D	Pprime_max with shaft loss, actual working variable
prime_mdense	(W/kg)	I	Prime Power Energy Density
EFFG	(non-dim.)	I	Main Generator Efficiency
Pgen	(W)	D	Actual Generator Power
fuel_cap_gal	(gallons)	I	Fuel Capacity
int_sto	(J)	I	Cbus, Intermediate Storage

int_sto_mdense	(J/kg)	I	Intermediate Storage Density
int_sto_qic	(A)	D	Cbus Initial value
int_sto_c	(farads)	D	Capacitance of bus
Vdc_nominal	(V)	I	Nominal Bus Voltage
dbusV	(V)	D	Bus Voltage
T_gen_nom	(N)	D	Desired Torque Output of Generator
i_gen_nom	(A)	D	Nominal Current from Generator
Kvg	(N/V)	D	Torque/Voltage Proportionality Constant for PM Generator
Ktg	(N/A)	D	Torque/Current Proportionality Constant for PM Generator
max_drive_power	(W)	I	Maximum Power of Drive System
maxdrive_thrust	(N)	I	Maximum Thrust of the Drive System
drive_mdense	(W/kg)	I	Energy Density of Drive System
EFFDRIVE	(non-dim.)	I	Drive System Efficiency (prime power output to ground)
main_arm_emax	(J)	I	Maximum Energy Capability of the Main Armament
main_arm_n	(#)	I	Number of Shots Ready on Main Armament
v_main	(km/sec)	I	Maximum Velocity of Main Armament Shot
n_rnds_main	(#)	I	Number of Shots Stowed for Main Armament
second_arm_emax	(J)	I	Maximum Energy Capability of the Secondary Armament
v_second	(km/sec)	I	Maximum Velocity of Secondary Armament Shot
second_arm_n	(#)	I	Number of Shots Ready on Secondary Armament
n_rnds_second	(#)	I	Number of Shots Stowed for Secondary Armament
defense_arm_emax	(J)	I	Maximum Energy Capability of the Protection System
defense_arm_n	(#)	I	Number of Shots Ready on Protection System
ESTORE_MAX	(J)	D	Total Power Available
Estorei	(J)	D	Initial Power Available
estore_mdense	(J/kg)	I	Power density for Pulsed-Power System
EFFCHARGE	(non-dim.)	I	Efficiency of Charger (Turbine Output to Estore)
Pcharge_max	(MW)	I	Maximum Power of the Pulsed-Power Charger Rated
PCHARGE_MAX	(MW)	D	Maximum Power of the Pulsed-Power Charger Actual
pcharge_maxrate_up	(W/s)	D	Maximum Rate of Charger Power Rise
pcharge_maxrate_dn	(W/s)	D	Maximum Rate of Charger Power Fall
Ploss_max	(W)	I	Maximum rate of loss of Pulsed Power
Ploss_factor	(non-dim)	I	Energy Storage Power Loss as a Function of Energy Storage Value
Init_PP=1.0	(non-dim)	I	Initial SOC of Pulsed Power System
Mveh	(kg)	D	Total Mass of Vehicle
frontA	(m^2)	D	Total Frontal Area
torqtable	(Nm)	D	2-D Lookup Table of throttle Versus ref_speed for torque
spdindx	(rad/s)	D	Index of Speed Settings for Lookup Table
throt_indx	(0-1)	D	Index of Throttle Settings for Lookup Table
sfctable	(gal/s)	D	2-D Lookup Table of Power Versus ref_speed to get fuel
speedindx	(rad/s)	D	Index of Speed Settings for Lookup Table
pwrindx	(W)	D	Index of Power Settings for Lookup Table
jnet	(Nm/s^2)	D	Net Inertia
Paux_misc	(W)	D	Total of all auxiliary power demands
Paux_NBC	(W)	I	Auxiliary Power devoted to NBC systems
Paux_threat_warn	(W)	I	Auxiliary Power devoted to the Threat Warning System
Paux_EWjam	(W)	I	Auxiliary Power devoted to Electronic Warfare Jamming
Paux_mmw	(W)	I	Auxiliary Power devoted to the Millimeter Wave Radar
Paux_prim_sight	(W)	I	Auxiliary Power devoted to the Gun's Primary Sight
Paux_comb_prot	(W)	I	Auxiliary Power devoted to Combined Protection System
Paux_psm	(W)	I	Auxiliary Power devoted to the Power Supply Module
Paux_smoke	(W)	I	from TARDEC *(not using yet)*

Paux_active_sig (W)		I	25-190kW from TARDEC *(not using yet)*
Paux_noise_cancel (W)		I	from TARDEC *(not using yet)*
Paux_active_protec (W)		I	from TARDEC *(not using yet)*
PTUR_ON	(W)	I	Power Demand with Turret Drive On
PTUR_OFF	(W)	I	Power Demand with Turret Drive Off
PLOADER_ON (W)		I	Power Demand with Autoloader Power On
PLOADER_OFF (W)		I	Power Demand with Autoloader Power Off
EMIN	(J)	I	min energy for diversion of mobility power
sshot	(m)	I	vector of positions for each shot (3 shots=three element array) each element of the array has the position of that shot
tshot	(sec)	I	vector of times for each shot (3 shots=three element array) each element of the array has the time of that shot
eshot	(J)	I	energy storage required for each shot
xshot	(m)	I	vector of x positions for each shot
yshot	(m)	I	vector of y positions for each shot
misss	(m)	I	vector of positions for entire mission
missv	(m/s)	I	vector of velocities for entire mission
misst	(sec)	I	vector of positions for entire mission
Pdrive	(W)	I	vector of drive power required for entire mission
Pturn	(W)	I	vector of turning power required for entire mission
slop	(rad)	I	vector of gradient for entire mission
CdRR	(N)	I	vector of drag due to rolling resistance for entire mission
Cd	(N)	I	Coefficient of Drag for Vehicle
JSHOT	(integer)	O	shot counter (first shot is jshot=1)
DELAY	(integer)	O	Shot Delay Tracker
SHOT_DELAY (sec)		O	Shot Delay Tracker
SHOT_OVERLAP (int.)		O	counter for shots delayed over next shot
TSHOT	(sec)	O	Actual Time of Shots
SSHOT	(m)	O	Actual Position of Shots
Te	(N)	O	Actual Torque Output of Generator
vgen	(V)	O	Actual Voltage Output of Generator
igen	(A)	O	Actual Current Output of Generator
iload	(A)	O	Current Load
Edc	(J)	O	Actual Energy Available on Bus
Vdc	(V)	O	Bus Voltage
PPload	(W)	O	Pulsed Power Energy Load
Tic	(N)	O	Torque of the IC Engine
Pic	(W)	O	Power of the IC Engine
fuel	(gal)	O	Fuel consumed
throt	(non-dim.)	O	Actual Throttle Output after Lag
throt_com	(non-dim.)	O	Actual Throttle Output
spd	(rad/s)	O	shaft speed of generator
Fwind	(N)	O	Aerodynamic Forces on Vehicle
Paux	(W)	O	Actual Dynamic Power Demand from Auxiliaries
recpid	(non-dim.)	O	Generator PID output
werr	(non-dim.)	O	Error tracker in PID driver controller
Pdr1	(W)	O	Actual Drive Power
Pdr3	(W)	O	Desired Throttle to engine controller
Pdrive	(W)	O	Drive Power
Pcharge	(W)	O	Actual Charger Output
t	(sec)	O	Time Counter
x	(m)	O	Position of Vehicle
v	(m/s)	O	Velocity of Vehicle

e	(J)	O	Energy Available in Energy Store
ee	(J)	O	Current Shot Energy Desired
trig	(integer)	O	Trigger signal

F.3 Variables used by programs

This section includes those variables not previously mentioned but still available in the Matlab workspace. These variables are other extraneous variables that should not be tampered with since they control program flow and error detection.

loadfile	(string)	D	Variable holding the actual filename of the vehicle file to be loaded
complete_file	(string)	D	Variable holding the actual filename of the vehicle file
filesaved	(integer)	D	Flag set to 1 if file was saved
veh_type	(string)	D	Type of Vehicle selected
engine_type	(string)	D	Type of Engine selected
default	(string)	D	String with Specific Filename

Appendix G: Glossary

editable text box - a prompt at which a new value for a parameter can be typed in, reached by clicking the mouse in the box first.
global parameters - these are parameters available throughout a software package
global variable - the actual variable names that are linked from file to file to pass global parameters
GUI - Graphical User Interface: The standard in software engineering today for premier user-friendly inputs to a program using a combination of mouse and keyboard.
m-file - a Matlab file, so named because of the .m extension. An m-file is essentially a script that keeps the user from having type repetitive commands over and over at the Matlab prompt.
Matlab - A high level mathematical programming language optimizing matrices to gain efficiency in speed in computation. A perfect solution to the repetitive nature of dynamic system simulation.
PID Controller - the standard proportional, integral and derivative controller used to simulate throttles and any other control system driving an error signal to zero.
powersim1.0 - the name of the actual Simulink file that the simulation model is stored
prime power - the power produced by the internal combustion engine and delivered to the generator, also termed prime mover.
pushbutton - a pushbutton is a Graphical User Interface(GUI) method of input done by clicking the mouse over the simulated button to begin some process.
Simulink - A Matlab program utilized to simulate dynamic systems of all types, linear and nonlinear, in order to analyze certain parameters. The program maximizes ease of use for the designer, allowing graphical block-diagram programming and a multitude of numerical integration techniques.
two-dimensional lookup table - a table that takes two indices and a given table, and utilizes that to interpolate a lookup for two inputs based upon the two indices.
workspace - the workspace is a term describing virtual storage areas for variables in Matlab. For instance, the main workspace is the only workspace accessible from the command line prompt, but other files can call functions that are given their own variable workspaces. One of the highlights of Matlab and Simulink is that values can be sent from any point in the Simulink graphical model to the main workspace of Matlab and then be plotted if the user desires.

Appendix H: Distribution List

Director
US Army Research Lab
ATTN: AMSRL OP SD TP
2800 Powder Mill Road
Adelphi, MD 20783-1145

Joe Schroeder
Cdr. USAARMC
Attn: ATZK-CDM
Bldg. 1109B
Fort Knox, KY 40121

Director
US Army Research Lab
ATTN: AMSRL OP SD TL
2800 Powder Mill Road
Adelphi, MD 20783-1145

Director
US Army Research Lab
ATTN: AMSRL OP AP L (305)
APG MD 21005-5066

MAJ Hardin
Cdr. USAARMC
Attn: ATZK-CDM
Bldg. 1109B
Fort Knox, KY 40121

Director
U.S. Army Research Laboratory
Attn: AMSRL-WT-PB (Edward Schmidt)
APG, MD 21005-5066

Director
US Army Research Laboratory
ATTN: AMSRL-WT-P (Pam Kaste)
APG, MD 21005-5066

Director
US Army TACOM-TARDEC
Attn: AMSTA-TR-D, MS 207
(Munira Tourner)
Warren, MI 48397-5000

George Chryssomalis
SAIC
3800 W. 80th St., Suite 1090
Bloomington, MN 55431

Director
US Army Ballistics Research Laboratory
ATTN: SLCBR-IB-B (W. Oberle)
APG, MD 21005-5066

Director
US Army TACOM-TARDEC
Attn: AMSTA-TR-D, MS 207
(John Lewis)
Warren, MI 48397-5000

Robert J. Taylor
Lockheed Martin Vought Systems
M/S: WT-21
P.O. Box 650003
Dallas, TX 75265-0003

George Solhan
MARCOR SYSCOM (AWT)
Amphib. Warfare Tech
Quantico, VA 22134-5030

Director
US Army TACOM
Mobility Technology Center
Attn: AMSTA-TR-M, Bldg 212
(Jennifer Hitchcock)
Warren, MI 48397-5000

Administrator
Defense Technical Information Center
Attn: DTIC-DDA
8725 John J. Kingman Road, Ste 0944
Ft. Belvoir, VA 22060-6218

John H. Gully
ARPA-EASTO
3701 N. Fairfax Drive
Arlington, VA 22203-1714

Commander
U.S. Army Armament Research,
Development and Engineering Center
Attn: SMCAR-FSE (Dennis Ladd) Bldg. 382
Picatinny Arsenal, NJ 07806-5000

Directorate of Combat Development
Attn: ATZK-CD (COL E. Bryla)
U.S. Army Armor Center and School
Fort Knox, KY 40121-5000

Bill Jones
HQ, US Army TRADOC
Attn: Mr. William Jones
DCSCD
Ft. Monroe, VA 23651-5000

Director
US Army Research Lab
ATTN: AMSRL OP SD TA
2800 Powder Mill Road
Adelphi, MD 20783-1145

Directorate of Combat Development
Attn: ATZK-CD (John Butler)
U.S. Army Armor Center and School
Fort Knox, KY 40121-5000