Phase I SBIR Final Report
“A Comprehensive HEV Design Tool for Dual-Use Applications”

Security Classification: Unclassified
Contract: DAAE07-98-C-X009
COTR: Harold Pangilinan / TACOM
Performance Period: November 1997 – May 1998

Submitted by:

Prime Contractor: ThermoAnalytics, Inc.
Principal Investigator: Keith R Johnson
Lead Research Engineer: David M Less

94X Airport Road
PO Box 66
Calumet, MI 49913
(906) 482-9560

and

Subcontractor: Boeing
Principal Research Engineer:
Wayne S Hammond
Research Engineer:
Jeff M Hansen

Seattle, WA 98014
(425) 393-8904

Subcontractor: Michigan Technological University / Keweenaw Research Center
Lead Research Engineer:
Mark D Osborne
1400 Townsend Dr.
Houghton, MI 49931
(906) 487-2696

May 19, 1998

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 19 May 1998	3. REPORT TYPE AND DATES COVERED Final: November 1997 – May 1998

4. TITLE AND SUBTITLE
A Comprehensive HEV Design Tool for Dual-Use Applications

5. FUNDING NUMBERS
Contract DAAE07-98-C-X009

6. AUTHORS
David Less, Keith Johnson, Mark Osborne

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
ThermoAnalytics, Inc.
94X Airport Road
PO Box 66
Calumet, MI 49913-0066

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Tank-Automotive and Armaments Command
ATTN: AMSTA-TR-R (MS 121), Mr. Harold Pangilinan
Warren, MI 48397-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

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

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

New design tools configured specifically for hybrid electric vehicle (HEV) applications are needed to achieve multiple design objectives. In Phase I the emphasis was on the development of a fully functional prototype HEV design tool. This tool served as the test bed to investigate the effects of simulation complexity on modeling accuracy and to evaluate the user interface in a design environment. The primary goal of the Phase II program is to produce an engineering design tool that will enable the user to design the power and drive trains, mobility, and thermal management of the vehicle. The HEV Design Tool will require only those inputs consistent with a preliminary design effort; detailed knowledge of the component substructure will not be required. The code interface will be devised to provide the user with the information that he needs - when he needs it - during the design process. The code will be capable of modeling both wheeled and tracked vehicles, in both series and parallel configurations, for dual-use military vehicles and commercial automotive applications.

14. SUBJECT TERMS
SBIR, hybrid electric vehicle, power systems, mobility, simulation, batteries, design tool, thermal analysis, infrared signature

15. NUMBER OF PAGES

16. PRICE CODE

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

NSN 7540-01-280-5500 Computer Generated STANDARD FORM 298 (Rev 2-89)
Prescribed by ANSI Std 239-18
298-102

TABLE OF CONTENTS

1.0 INTRODUCTION

The Hybrid Electric Vehicle (HEV) technology is being explored both in the military and automotive industries. The future combat vehicles will require increased electrical power for their weapons systems, better mobility for the mission, and increased signature suppression for survivability. The automobile requires increased fuel efficiency and decreased emissions. The HEV has the potential to meet these dual-use needs.

The advantages that hybrid electric systems offer for both commercial and military vehicles are quite impressive and include the following:

1. The physical size and weight of the engine are reduced because it is designed to the average load rather than the peak load.
2. By running the engine at a steady optimal design load, fuel consumption and exhaust emissions are both reduced.
3. Through regenerative braking, energy is recaptured electrically rather than dissipated thermally.
4. A HEV can be operated with a variety of fuels thus reducing logistics problems and dependence on fossil fuels.
5. The reduced noise and waste heat generation will decrease acoustic and thermal signatures resulting in enhanced capabilities for silent mobility and extended silent watch.
6. Increased mobility is achieved through greater acceleration, speed, and range. Augmenting the APU with high power devices such as batteries, flywheels, and ultracapacitors results in quicker accelerations and faster dash speeds.
7. The power source in the HEV can be used for emergency applications in the field.
8. For the military the additional electrical capacity adds a powerful electronic punch that can result in greater lethality.
9. Since engines in a series configuration need not be physically connected to the axle a greater flexibility in vehicle design is possible. This flexibility can lead to lower silhouettes that reduce visual signature, and lower centers of gravity that improve stability, handling, and agility.
10. The HEV, especially one in the parallel configuration, provides redundant modes of operation for greater robustness.

In terms of power systems architecture, a HEV can be designed in one of several configurations. There are commonalties in these different design architectures. In all cases there will be an electric motor connected to at least two different sources of power. One power source will be an energy storage device that could be a battery, flywheel, and/or ultracapacitor. The other source, an auxiliary power unit (APU), will charge the energy storage device(s). The APU can be an internal combustion engine (diesel, gas, or alternative fuel), a gas turbine, or a fuel cell. As shown by the simplified configuration illustrated in Figure 1, the power system of the HEV has several interconnections and may be tied to other electrical components in the vehicle.

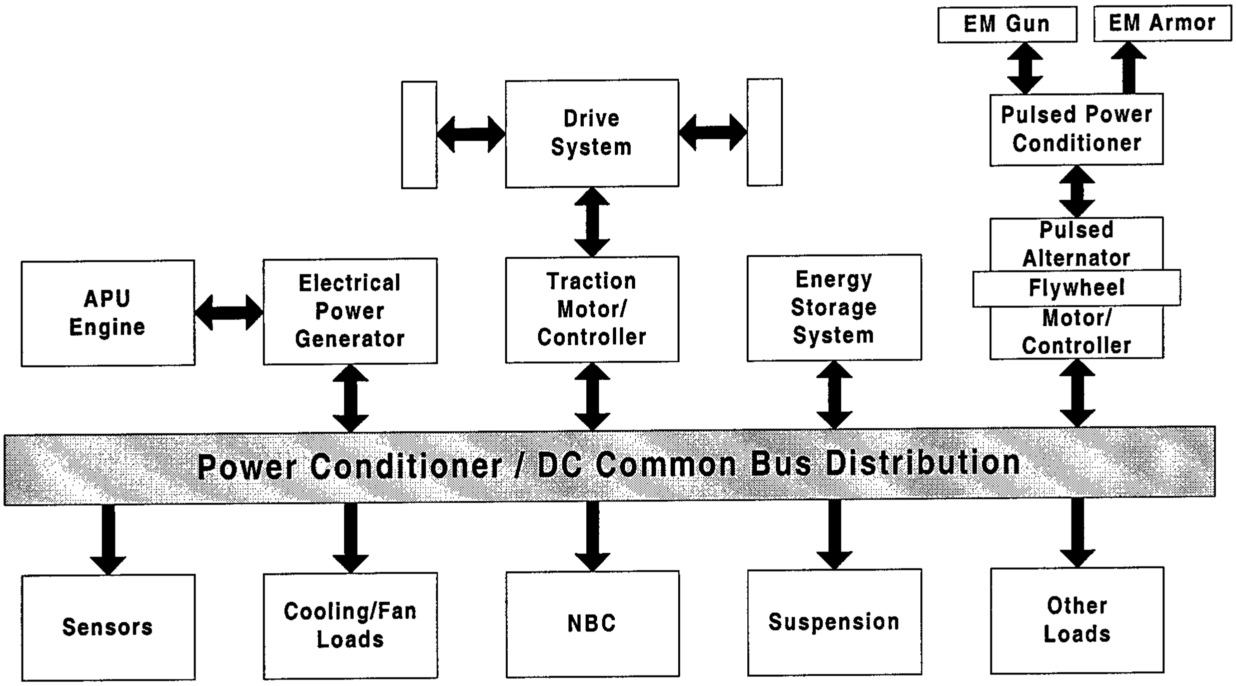

Figure 1. Series Configuration for a Hybrid Electric Vehicle

The two power sources can be designed in either a series (as in Figure 1) or parallel configuration. The series design uses the APU to exclusively generate electricity for the electric motor while the parallel configuration uses a power control strategy in which the APU can either charge the energy storage and/or directly power the drive system. An additional design distinction is whether the hardware configuration and control strategy is charge-sustaining or charge-depleting. Charge-sustaining refers to a system that sizes the APU to the average power load and can thus run indefinitely under those conditions. Charge-depleting designs do not have the capability to recharge at the same rate that they are discharged under average conditions.

Additional design variants include the electrical-assist parallel, the APU-assist parallel, the thermostat series, and the load-leveler series hybrids. The electrical-assist parallel (or power-assist) uses the APU as the primary means of propulsion and the electric motor only for peak power needs (and to capture regenerative braking energy). The APU-assist parallel is the opposite in that it uses the electric motor as the primary and the APU for higher torque or speed needs. The thermostat series hybrid operates entirely on electrical power with the APU operating on a thermostatic cycle that charges the batteries according to a high and low set-point state of charge (SOC). The fourth hybrid variant uses the load-leveler series design that matches the average power requirement of the vehicle with the electrical power generated by the APU. The SOC fluctuates around a mid-level charge causing the APU to change its power output based on load changes.

With all of these possible variations in the design of hybrid vehicles, the old assumptions used on traditional vehicle design are no longer valid and therefore the designer needs a new set of tools to facilitate design choices. A tool especially configured for HEV design is needed that will allow the designer to consider the full set of tradeoff concepts. In addition to considering all of the hardware and power control strategies identified above, the tool must also be able to include optimization techniques for achieving multiple design objectives.

As with most engineered systems, optimal design conditions often conflict with one another. The military has a need for its future combat vehicles to require increased electrical power. The electrical power consumers include sensors, pulsed power weapons, active defense system, electromagnetic armor, electric suspension systems, as well as electric propulsion. At the same time, there is a drive to augment mission capability while reducing logistics requirements, reduce overall costs (fuel, maintenance, and production), and reduce armor weight and volume. Not only must all of these desired features be individually optimized, they must be optimized when integrated into a single system. Whether the application is military or commercial automobile, the design technology must be able to integrate power generation, energy storage, and power distribution.

It is the objective of this program to produce a HEV design tool that will result in military vehicles that are stealthy, fast, easy to deploy, highly maneuverable, and mission effective while combining the key goals of the civilian vehicle which are fuel efficiency and reduced emissions.

2.0 HEV DESIGN TOOL TECHNICAL OBJECTIVES

The main objective of the Phase I work was to produce a prototype engineering software tool for hybrid electric vehicle design that would be accessible to all designers and consistent with manufacturing databases. This tool had to be dual-use, capable of handling both military vehicles (tracked and wheeled) and commercial automotive applications. To meet this goal we determined that the software design tool development should follow these objectives:

1. Focus algorithm development on power systems analysis.
2. Concentrate the analysis on thermal heat management, mobility aspects (road loads, speed requirements, gradients), and component weight and size.
3. Use databases containing manufacturer specifications of components and other system data as inputs to the algorithms.
4. Write the code in FORTRAN so that future engineering development and expansion could occur readily as hybrid systems develop. Avoid platform-dependent coding so that the software could be ported easily to a wide variety of engineering platforms including PC's and workstations. Use simple ASCII formats for output files to permit easy importation into spreadsheet programs.
5. Employ a computationally fast and simple mobility analysis that would permit more detailed analysis in future work.
6. Provide the vehicle designer with the information that he needs, when he needs it. During the component selection process, estimate the vehicle performance with that component installed, and provide an overall performance estimate once all components have been selected. Include all simulation variables (power, torque, voltage, current, speed, state of charge, etc.) in the output file. Summarize the vast amount of data produced by the simulation to a few key pieces of information that will identify which components limited the vehicle performance and under what conditions.

3.0 PHASE I PHYSICS AND CODE ARCHITECTURE

The Phase I HEV Design Tool uses a scope and sophistication in its physics that is similar to that used by the more popular existing HEV design codes such as DOE's SIMPLEV[1]. What sets the HEV Design Tool from the other HEV simulation programs is that it includes the following features: database of component specifications, mission profile input, thermal modeling of heat flows and component temperatures, and an interface that is specially designed for design trade analysis.

The Phase I research was broken in several different tasks:

1. Researching the mobility model;
2. Designing the code architecture to best meet the needs of the vehicle designer;
3. Determining how to model the power and drive system;
4. Creating the database of vendor specifications;
5. Verifying the operation of the prototype Phase I code.

The following sections discuss the results of this research, task by task. The discussion of the power and drive system is divided into two sections; the first (Section 3.3) describes the underlying physics and logic flow while the second (Section 3.4) discusses the simulation on a subroutine basis. A final section (Section 3.7) outlines the conclusions of the Phase I work.

3.1 Phase I Mobility Research

The Phase I HEV code employed a rudimentary mobility model as illustrated in Figure 2. Two rolling/motion resistance factors, simulating a tracked vehicle traveling on paved and cross-country roads, formed the basis of the model

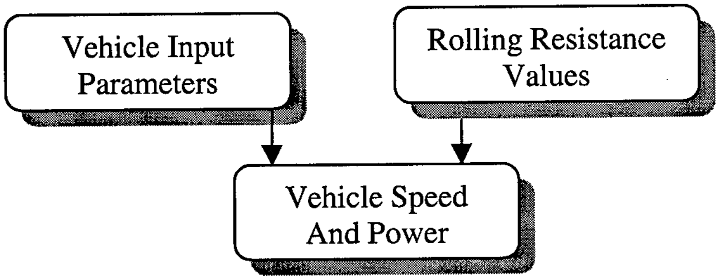

Figure 2. Phase I Mobility Model

The purpose of the mobility portion of the overall code was to determine a number of factors relating to an HEV type of vehicle. These factors include:

1. Power required traversing a certain terrain;

---

[1] Department of Energy (DOE), 1991, SIMPLEV: A Simple Electric Vehicle Simulation Program – Version 1.0, DOE/ID-10293, Idaho Falls, Idaho, June 1991.

2. Power available/remaining for the vehicle at any point during a particular traverse;
3. Mobility analysis.

The simplifications used during the Phase I effort reduced the terrain definition to only rolling/motion resistance values. These were 90 #/ton for concrete and 125 #/ton for cross-country terrain. Technically, these resistance values and their constancy imply:

1. No slope; the road is flat and level;
2. No curves;
3. Dry road; no slippage due to wet conditions;
4. No obstacles or vegetation to cross;
5. No other extraneous conditions affecting traversing requirements.

As part of the Phase I effort, KRC examined a full-up mobility code, the NATO Reference Mobility Model (NRMM), as a means of planning how the mobility model would be expanded during Phase II. A code such as NRMM requires very detailed information about the vehicle – information that would be unavailable or inappropriate during a preliminary design process for which the HEV Design Tool is tailored. Consequently, a subset of these inputs and analyses will be integrated into the HEV Design Tool during Phase II. The emphasis will be on mobility factors that affect or are affected by the vehicle's power system. Appendix A provides an overview of the NRMM inputs, denoting which ones would be appropriate for inclusion in a Phase II code.

3.2 HEV Design Tool Code Architecture

Based on our extensive experience in vehicle design exercises, we devised the code architecture for the HEV Design Tool diagrammed in Figure 3. The code consists of two primary components: the Component Selector and the Simulator.

When selecting components, the vehicle designer needs to know the specifications of the components and how installation of those units would effect the overall performance of the vehicle. The Component Selector displays two tables for the major components: a table of key component specifications and a table that lists vehicle performance parameters such as maximum continuous speed and acceleration times with that component installed. For minor components that are meant to work in conjunction with other components (e.g. controller for the motor), the specifications that need to be matched between the components are displayed.

In order to predict vehicle performance during the component selection process, the Component Selector needs access to information about the vehicle and the conditions under which it is being designed to operate. Consequently, the Component Selector needs the VAT (Vehicle dATa – vehicle mass, aerodynamic drag data, wheel radius, etc.) and SCN (SCeNario – weather, rolling resistance, driving cycle, etc) files. Specifications for the components are read in from the component database. For ease of maintaining the database, the main component database is delivered as an EXCEL spreadsheet file. For use with the HEV Design Tool the database must be exported as an ASCII file and named COMPONENT.DAT.

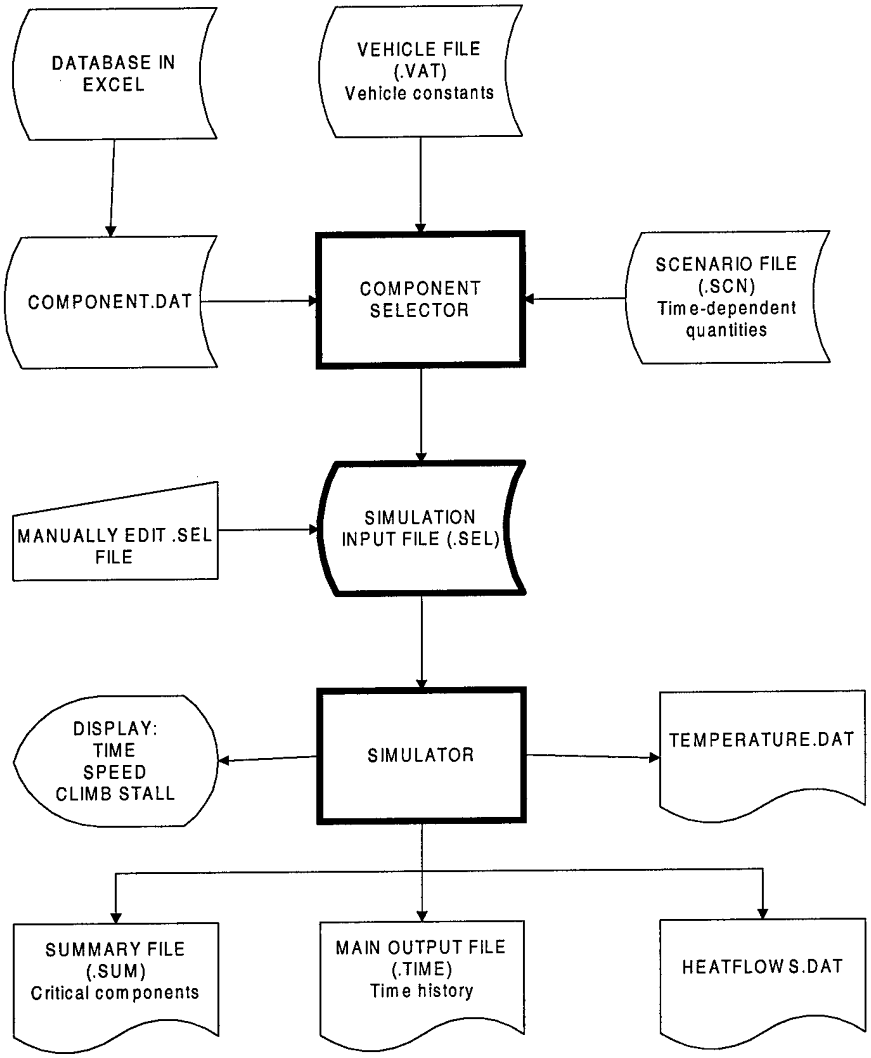

Figure 3. Phase I Code Architecture

By having access to the vehicle's VAT and SCN files in addition to the component database, the Component Selector knows everything that is required for the simulation. The Selector creates a SEL file that contains all the data that the Simulator needs. The SEL file is an ASCII text file and all parameters are labeled. This arrangement allows the vehicle designer the ability to quickly edit the SEL file and try out variations of the vehicle and/or driving cycles.

The Simulator produces several outputs. A screen display informs the user of the progression of the simulation. The main output file (TIME) lists all simulation variables at every time step. The summary file (SUM) tabulates which components limited vehicle performance. The thermal simulation produces two file outputs - one lists the heat generated by each component and the other the temperature of each component assuming that the only means of cooling is a low speed forced air flow of ambient air.

3.3 Phase I Power and Drive Train Research

Our research into power and drive train models led us to devise the following HEV simulation scheme. The overall simulation scheme illustrated by Figure 4, starts at the interface between the road and wheel/track and progresses up the drive and power trains.

Road Load Power: The power required at the interface between the wheel/track and road are defined by the power needed to overcome the forces of inertia, gravity (road grade), aerodynamic drag, rolling resistance, and wheel bearing drag. To obtain the required power, the code multiplies each road load force by the vehicle velocity. The velocity used in this calculation is one that is averaged between the velocities at the start and at the end of each time step. The rolling resistance is input directly by the user; the choice of this value reflects the road condition and whether the vehicle is wheeled or tracked.

Wheel/Track Slippage: The first check on vehicle mobility is whether or not the wheels/track slip. The code multiplies the static friction by the weight and, in the case of vehicle acceleration, by the fraction of wheels directly powered by the drive train, and compares this against the force at the wheel/track-road interface. If this, or any subsequent, check fails, the desired speed is revised for the current time step and the calculation restarted – new road load powers are computed and the code proceeds back up the drive and power train.

Transmission: The gear ratio listed for the transmission should be the overall gear ratio. This ratio is the only multiplier for shaft speed between the motor output and the wheel/sprocket. The torque is modeled as the power divided by the product of shaft speed and 2π. The code checks to see that the maximum output power of the transmission is not exceeded. When the power at the motor-transmission interface is computed, the code considers the efficiency of the transmission.

Motor: The motor can be operated at either the continuous or "5 minute" power rating. If the output of the motor exceeds its maximum torque rating, the code attempts to fix the problem by shifting to a higher gear ratio. The code then checks the motor power load, and if the motor has not been operating at its higher power rating for more than 5 minutes, then the motor is set to its "5 minute" power rating. The Phase I code assumes a 50% duty cycle for the motor – once the motor drops from its "5 minute" rating to its continuous rating it must wait 5 minutes before operating again at its "5 minute" level. If the check on motor power still fails, or if the maximum motor torque is exceeded, the desired vehicle speed for that time step is revised. Since the Phase I HEV Design Tool does not model the motor power and torque as functions of shaft speed, this calculation fails to truly model the performance of a motor. When the required power output of the motor controller is calculated, the efficiency of the motor is considered.

Motor Controller: The motor controller enters the calculation only as a power efficiency factor.

Inverter for the Motor: The inverter for the motor enters the calculation only as a power efficiency factor.

Auxiliary Power Load: At this point in the calculation, the power required by the auxiliary load (lights, communication equipment, and electromagnetic gun) is added to the power required by the inverter to power the drive train.

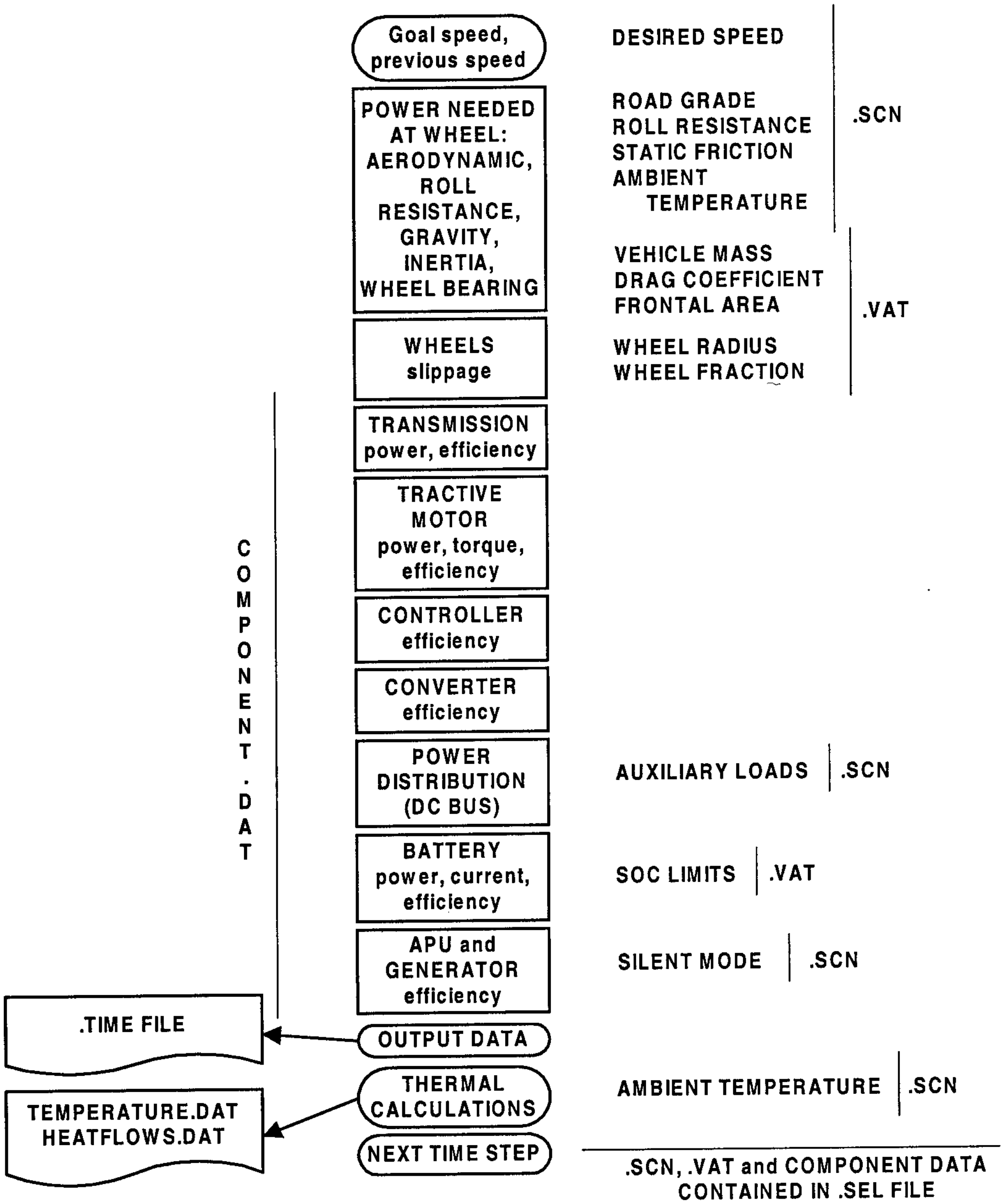

Figure 4. Phase I Code Calculation Flow Showing Input and Output from Files

Power Distribution: The power distribution scheme is constrained by the fact that the Phase I code only handle vehicles designed in the series configuration. If the APU is switched on at this point in the simulation, then its power is subtracted from the combined power requirements of the drive train and auxiliary load. On the first pass through the algorithm for each time step, the APU is switched on *only* if the APU is presently charging the battery back up to its high state of charge (SOC) limit. In other words, the code assumes that the APU is switched off unless the battery is being recharged or if the power storage devices (battery, etc) cannot supply enough power to meet the power demands. At this point in the simulation, the code predicts the power expected out of the battery based on the power supplied (if any) by the APU and the demand load.

Battery: The power supplied by the battery is modeled as $I*R^2 + VOC*I$ where I = battery current, R = internal resistance of the battery, and VOC = open circuit voltage of the battery. Checks are made if the battery maximum power and current specifications are exceeded. If either check fails, the code first checks to see if the APU is on or off. If the APU is off the APU is switched on and the power distribution recalculated. If the checks fail with the APU switched on, the code calculates a new vehicle speed and restarts the calculation for that time step. Once the code passes these final checks, the state of charge of the battery is increased or decreased depending on the sign of the battery power load. A positive battery power load denotes that the battery is supplying power to the system and a negative load denotes that the battery is being charged by the APU. Separate efficiencies are used for battery charge and discharge conditions.

Flywheel: The operations of flywheels and ultracapacitors are not included in the simulation. The Selector, however, does predict the vehicle performance obtained from using a flywheel.

APU: If designated switched on by the previous power subroutines, the auxiliary power unit (APU) supplies power at its continuous rating. The code multiplies this power by the efficiency of the generator. The efficiency of the APU is only considered in the calculation of the heat generated by the APU. If the APU is active during a time step, the fuel in the fuel tank is decreased based on the fuel consumption of the APU (BSFC * APU power * time interval). The APU is not allowed to be active during silent mode. If the APU is active and the battery is fully charged to its high SOC limit, then any excess power is wasted by the system (i.e., Phase I code does not allow throttling back of APU but Phase II code will).

3.4 Subroutine Description

The code is divided into a dozen subroutines; each supplied as a separate source code file. Each of these subroutines will be individually discussed in the following sections. File names presented here and are described in detail in Section 4.

HEV_ALL_MAIN: The main routine for the HEV Design Tool defines the debug level and file units, and calls the Selector, Simulator, and Post Processor as needed.

HEV_SELECTOR: The Selector must perform several tasks: 1) it must read in the specification database, 2) estimate vehicle performance based on those specifications, and 3) provide the specifications and performance data in a manner that will aid the user in the design of his vehicle. First, the routine reads in the VAT file to learn basic information about the vehicle under consideration. From the SCN file the routine grabs the first line of weather data and uses that data (air temperature and density) in the vehicle performance calculations. On the first pass through the Selector the code estimates the mass and efficiencies of components; these estimates are required to permit the calculation of vehicle performance when not all components have been selected. For each component the Selector reads in the specifications from the database, converts the specifications to kgs units, and displays the most important specifications to the user. The code calls the subroutines CONST_SPEED and CONST_ACC to predict maximum speeds and accelerations. For the transmission selection the code calls CONST_TORACC and GRADE_MTORQUE to predict the maximum acceleration and the maximum road grade that the vehicle can ascend at a specified speed based on motor torque. Once all components have been selected, the code updates its calculation of the overall mass of the vehicle (it has been continually updating the component efficiencies during the selection process) and displays a summary table of vehicle performance based on each component's specifications. The Selector also tabulates the voltage requirements of the power components before saving all the information to a SEL file.

CONST_SPEED: Used by the Selector, this subroutine finds the velocity such that the sum of aerodynamic, wheel bearing, gravity, and roll resistance powers match the available power. Inertia is not considered since the routine assumes that the vehicle is moving at a constant speed.

CONST_ACC: Used by the Selector, this subroutine finds the time to reach a given velocity from a dead stop by balancing the sum of inertia, aerodynamic, wheel bearing, gravity, and roll resistance powers with the available power.

CONST_TORACC: Similar to CONST_ACC, this subroutine is also called by the Selector to determine the time to reach a specified speed from a dead stop. CONST_TORACC balances available torque with the forces that the vehicle needs to overcome.

GRADE_MTORQUE: Used by the Selector, this subroutine finds the road grade such that the sum of the aerodynamic, wheel bearing, gravity, and roll resistance forces match the available torque.

SIMULATOR_MAIN: This routine drives the simulation. It begins by initializing the simulation variables, including hardwiring heat capacity values for all components (all heat capacities are set to 460 J/kg-C, a typical value for metal). The main output file (TIME), heat flow and temperature files are created, opened, and headers written. The Simulator then reads in the SEL file. It reads in the time-dependent data (driving cycle and weather) as needed. For the thermal calculation it calculates the area over which each component will be subjected to a convective flow of ambient air. It assumes that this area is half the wetted area of a bounding box having the same dimensions as each component. For the power distributor/DC bus the code creates a circular-cross-section bar of copper whose length matches the longest dimension of the power pack and whose diameter is selected based on the maximum current of the motor controller. For each time step, the routine calls SERIES_STEP. After each call, the routine calculates the heat generated by each component as the product of the power passing through each component by one minus the efficiency of that component. The exception to this is the DC bus whose heat is calculated as the square of the battery current multiplied by DC bus's resistance (hardwired as being that of a copper bar). The routine employs an elementary thermal model that considers the heat generated internally, a convective flow of ambient air (the coefficient of convection is hardwired to be 30 W/m^2-C, a value consistent with a low speed forced air flow), and the heat capacity of the component (C_p*mass). All variable information is output and SERIES_STEP is called for the next time step.

SERIES_STEP: This subroutine contains most of the physics of the simulation. Section 3.3 describes the process that SERIES_STEP uses to predict vehicle motion and performance. The highlights of the process, along with the prominent database information, are diagrammed in Figure 4.

NEW_SPEED: Called by SERIES_STEP to calculate what speed the vehicle can move given a specified power at the wheel. This is called each time the desired speed needs to be revised due to lack of power. The new desired speed is calculated by determining at what speed will the power that the vehicle must overcome equals the available power.

NEW_MTORQUE: Similar to NEW_SPEED, this subroutine is called by SERIES_STEP when the desired speed needs to be revised due to insufficient motor torque.

POSTPROC: The post processor reads the POST.POS file created by SERIES_STEP. Whenever SERIES_STEP must adjust the desired speed due to a limitation of a component, it appends this information to the POST.POS file. POSTPROC reduces this data to a design aid summary.

HEV_COMMON: This include file lists the common blocks by which the variable data is transferred between the calculation subroutines.

3.5 Component Database

For use with the HEV Design Tool, we created a database of vendor specifications for HEV components. The database contains specifications beyond those required by the Phase I Selector and Simulator to allow rapid upgrading of the HEV Design Tool. Table 1 lists the specifications stored in the database for each component.

The database is in two forms. The version delivered as an EXCEL spreadsheet file contains the specifications as obtained from the vendors. We created the second version by exporting the EXCEL spreadsheet to an ASCII file. Later we edited the ASCII file and inserted missing specification required for the simulation. The inserted data was obtained through a mix of interpolating from other components and through engineering judgement.

3.6 Phase I Code Verification

We confirmed the operation of the Phase I HEV Design Tool using four test cases: the MARCAV HMMWV[2], the SCAT HMMWV with Unique Mobility motors[3], the DARPA hybrid electric Bradley Fighting Vehicle Demonstrator[4], and the SCAT bus with a Westinghouse power train[5]. For each of these four test cases we had a number of different vehicle performance parameters and most, but never all, of the component specifications required by the HEV Design Tool. We estimated the missing component specifications by interpolating the data from other components within the HEV database. We approximated the values of missing vehicle parameters by comparing the hybrid vehicles with their conventional counterparts.

---

[2] Kathy C Bearden and Daniel R Markiewicz, 'Development of a Hybrid Electric HMMWV', pp. 155 – 166, Second International Conference on All Electric Combat Vehicle (AECV), June 1997.

[3] Unique Mobility press release 97-19

[4] Tom Ferro, 'Hybrid Electric Drive BFV Demonstrator', pp. 189 – 216, Second International Conference on All Electric Combat Vehicle (AECV), June 1997.

[5] Frank A Lindberg, 'Large Power Trains for Electric and Hybrid Buses', 95ATS039, ISATA 95.

Table 1. Specifications Contained in the Component Database

APU	TYPE [DIESEL, TURBINE], CONTINUOUS POWER [HP], TORQUE [N*m] & RPM AT CONTINUOUS POWER, 5 MINUTE POWER [HP], TORQUE [N*m] & RPM AT 5 MINUTE POWER, PEAK POWER [HP], TORQUE [N*m] & RPM AT PEAK POWER, MAX RPM, BSFC [g/kw/hour, EFFICIENCY [%], MASS [kg], DIMENSIONS [m]
MOTOR	TYPE [DC, AC], CONTINUOUS POWER [HP], TORQUE [N*m] & RPM & VOLTS & CURRENT AT CONTINUOUS POWER, 5 MINUTE POWER [HP], TORQUE [N*m] & RPM & VOLTS & CURRENT AT 5 MINUTE POWER, PEAK POWER [HP], TORQUE [N*m] & RPM & VOLTS & CURRENT AT PEAK POWER, MAX TORQUE [N*m], EFFICIENCY [%], MASS [kg], DIMENSIONS [m]
TRANSMISSION	MAX POWER [kW], EFFICIENCY [%], # GEARS, MASS [kg], DIMENSIONS [m], GEAR RATIOS
BATTERY	SPECIFIC ENERGY CAPACITY [W*hour/kg], SPECIFIC POWER [kW/kg], MAX CURRENT [A], OPEN CIRCUIT VOLTAGE [V]. CHARGE EFFICIENCY [%], DISCHARGE EFFICIENCY [%], CYCLE LIFE [# CYCLES], RESISTANCE [OHMS], MASS [kg], DIMENSIONS [m]
FLYWHEEL	ENERGY CAPACITY [J], MAX POWER [kW], CHARGE EFFICIENCY [%], DISCHARGE EFFICIENCY [%], MASS [kg], DIMENSIONS [m]
GENERATOR	MAX POWER [kW], OUTPUT VOLTAGE [V], MAX EFFICIENCY [%], MASS [kg], DIMSENSIONS [m]
CONVERTER	TYPE [DC-AC, DC-DC], MAX CURRENT [A], MAX EFFICIENCY [%], MIN INPUT VOLTAGE [V], MAX INPUT VOLTAGE [V], OUTPUT VOLTAGE [V], MASS [kg], DIMENSIONS [m]
CONTROLLER	TYPE[AC, DC], MAX POWER [kW], MAX CONTINUOUS CURRENT [A], MAX 5 MINUTE CURRENT [A], EFFICIENCY [%], MIN VOLTAGE [V], MAX VOLTAGE [V], MASS [kg], DIMENSIONS [m]

The MARCAV HMMWV reference provided details about the vehicle and the performance goals for the vehicle. The motor torque specification seemed unusually low when compared with other motors in its power range. When tested in the simulation the specified motor torque severely limited the performance of the vehicle. As a result of these observations we created a new database entry in which the max motor torque was upgraded to be consistent with other motors. As shown in Table 2 the HEV Design Tool matched the acceleration and max speeds – on level and sloped roads – very well. Variance in the range predictions were primarily due to uncertainties about the conditions under which the performance parameters were derived (e.g. what roll resistance/type of road?). Since the Phase I HEV Design Tool assumes that the APU is always operated at its optimal design point, the vehicle speed is not a priori determined by stating that the APU is on. In other words, when powered by the APU, the hybrid electric HMMWV could travel at 10 or 20 or 40 or 60 mph and use the same amount of fuel per time in each instance. The observed fuel efficiency (miles per gallon) varies directly with vehicle speed. As previously noted, the Phase II code will allow more sophisticated APU controls (e.g. throttling and cycling on/off).

In contrast, the performance data for the SCAT HMMWV was measured data (Table 3). The acceleration and max speed on level ground all matched very well between simulation and measurements. When the simulation was run at the predicted max speed of the vehicle, the range using APU matched almost exactly.

Table 2. MARCAV HMMWV Performance Data

Specifications	Performance Goals	Performance Predictions
Time to 50 mph	< 10 sec	8.5 sec
Max speed on 60° grade	6 mph	8 mph
Max speed	60 mph	74 mph (continuous rating) 82 mph (5 minute rating)
Range using battery at 25 mph	20 miles	34 miles
Range using APU	300 miles	214 miles at 25 mph

Table 3. SCAT HMMWV Performance Data

Specifications	Measured Performance	Performance Predictions
Time to 50 mph	8sec	7 sec
Max speed on 60° grade	17ph	10 mph
Max speed	60+mph	74 mph (continuous rating) 89 mph (5 minute rating)
Range using battery at 25 mph	20 miles	33 miles
Range using APU	300 miles	110 miles at 25 mph 308 miles at 70 mph

There is some controversy surrounding the component specifications used in the HEV simulation for the electric Bradley. The data presented in Table 4 does not reflect improvements made after-the-fact with regards to the accuracy of the specifications. The predictions match the actual performance parameters very well with the exceptions of performance on grades (due to an incorrect gear ratio) and max speed of the vehicle. This latter discrepancy disappeared when an improved set of motor specifications was input.

Table 4. DARPA Electric BFV Performance Data

Specifications	Actual Performance	Performance Predictions
Time to 20 mph	7.7 sec	6 sec
Max speed on 60° grade	3.1 mph	2.7 mph at 50° grade
Max speed on 10° grade	17.1 mph	9.9 mph
Max grade at 5 mph – battery only - hybrid mode	17° 33°	1.4 mph at 10° grade 4.0 mph at 30° grade
Max speed	42 mph	7 mph (continuous rating) 41 mph (5 minute rating)

The overall gear ratio was not available for the SCAT bus; thus the simulation does not match the performance for grades very well, as shown in Table 5. The acceleration, max speed, and fuel efficiency (all on level ground) match extremely well. There is a discrepancy in the acceleration performance that is best illustrated when plotted. In Figure 5, the HEV Design Tool (speed_hev) predicted an exponential acceleration whereas the Westinghouse data (speed_west) showed a more linear acceleration from a dead stop. The exponential acceleration is to be expected from the physics of a hybrid electric vehicle, but one would assume that for a bus one would want the less sudden acceleration profile as shown by the Westinghouse data. The bus presumably had a mechanism to prevent such rapid accelerations. Alternatively, the acceleration profile may been due to the relationship between motor power/torque and shaft speed. In the Phase I HEV Design Tool, shaft speed and motor power/torque are not coupled together.

Table 5. SCAT Electric Bus Performance Data

Specifications	Actual Performance	Performance Predictions
Time to 18.6 mph (30 km/hr)	10 sec	3 sec
Time to 37 mph (60 km/hr)	20 sec	16 sec
Time to 56 mph (90 km/hr)	>40 sec	47 sec
Max speed on 10° grade	25 mph	10 mph
Max speed on 5° grade	40 mph	19 mph
Max speed	55+ mph	60 mph (continuous rating) 72 mph (5 minute rating)
Fuel efficiency at 42 mph	10.65 mpg	9.62 mpg

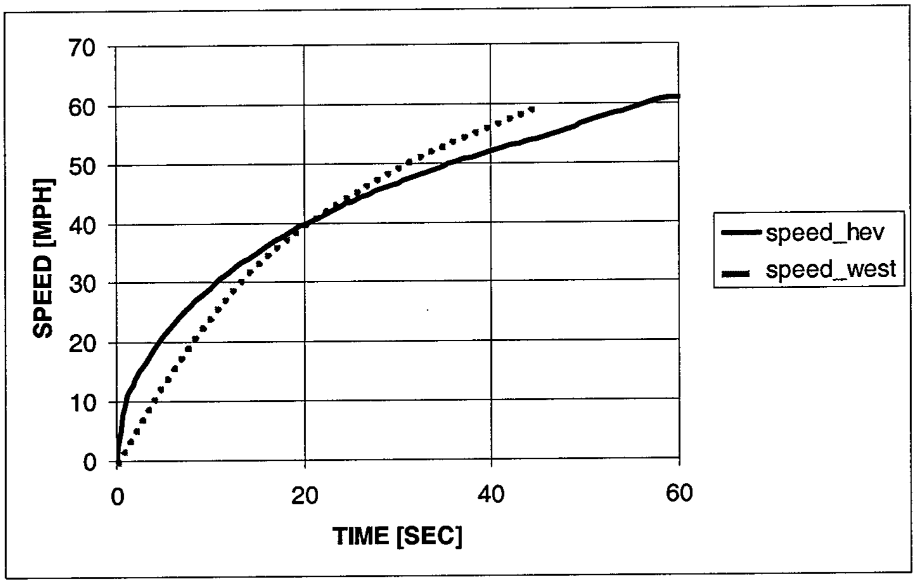

Figure 5. Acceleration of SCAT Electric Bus

The simulated acceleration profile for the DARPA electric BFV, shown in Figure 6, matches nearly exactly the data provided by UDLP for operation with the APU alone. In the simulation, there is no mechanism to tell the code not to tap the battery for power, but in this case almost all of the power was provided by the APU. Both acceleration curves for the electric BFV reveal the exponential profile predicted by the HEV Design Tool for the electric bus (Figure 5).

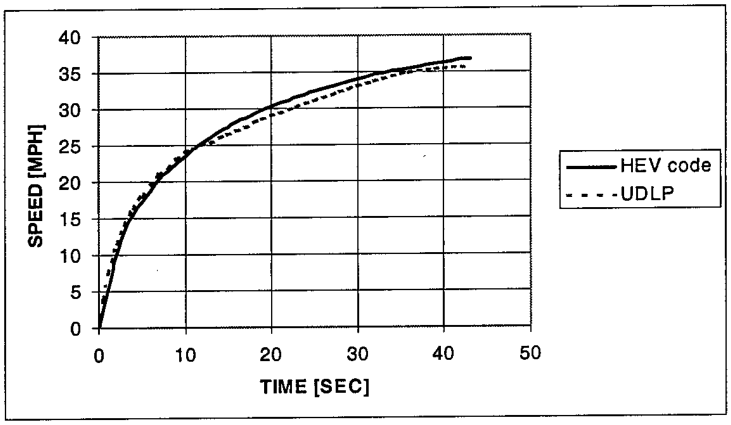

Figure 6. Acceleration of Electric Bradley Fighting Vehicle

Figures 7-9 are examples of the outputs that the Phase I HEV Design Tool can generate. In this simulation, the SCAT HMMWV was driven up a series of increasing road grades from 0° to 60° as fast as the vehicle could go. At the two-thirds mark through the simulation the road leveled off and the speed of the HMMWV was set at 50 and then 25 mph (road grades and speeds are plotted in Figure 7).

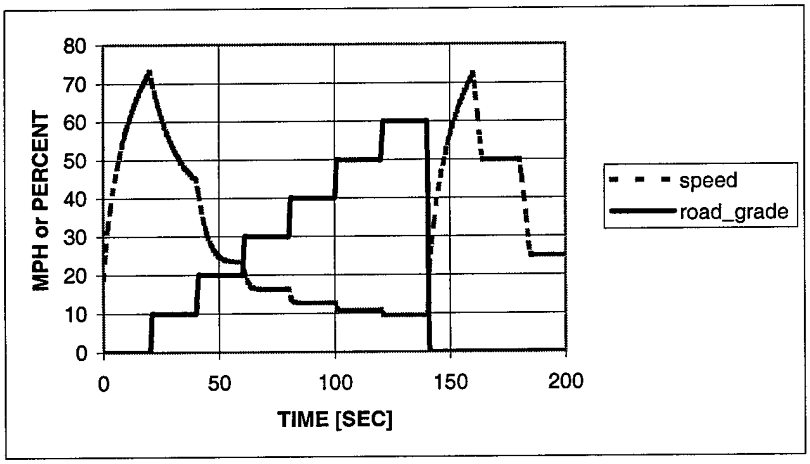

Figure 7. Speed and Road Grade for HMMWV Power Simulation

In the Simulator output of the forces and powers that the vehicle must overcome, plotted in Figure 8, several processes can be identified. Until the end of the simulation when the speed of the vehicle is set at a value less than maximum, the vehicle is power-limited. Consequently, there is a balance of power. At the start of the simulation, inertia is the largest player; in time, the inertia trades off with aerodynamic drag as the vehicle speed increases. As soon as the vehicle begins to ascend a grade, gravity forces dominate the power balance. At each increment of road grade, the Simulation accurately predicts that the vehicle will use its inertia to propel the vehicle up the incline at a faster speed that it can maintain on that slope. When the vehicle hits level ground inertia and aerodynamic drag once again trade off one another. At each decrement in speed the vehicle's inertia prevents an instantaneous reduction in vehicle speed.

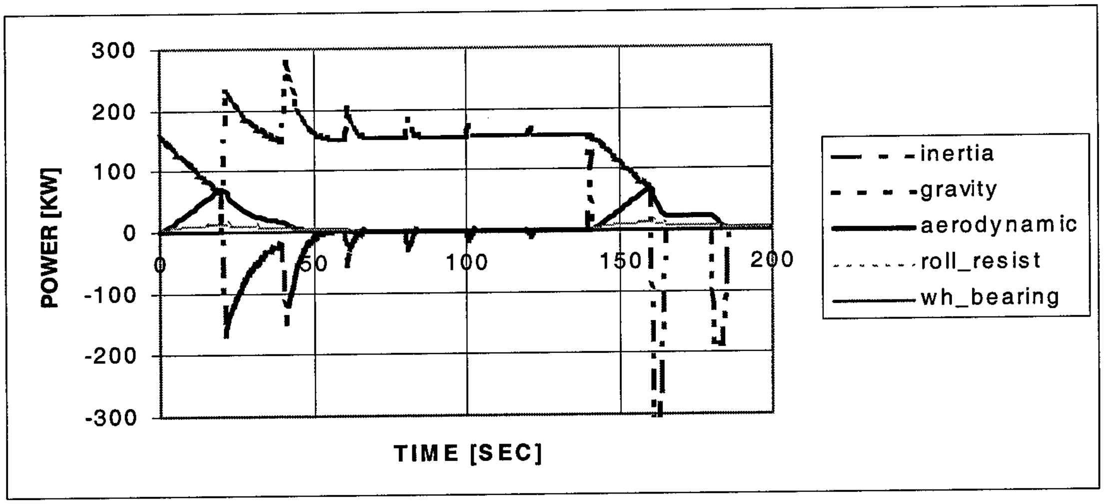

Figure 8. Mobility Power Levels the HMMWV Must Overcome in Simulation

The Simulator output also lists the source of the power driving the vehicle as plotted in Figure 9. The top two lines follow the power requirements at the output of the motor controller and at the wheel. Due to the efficiencies of the motor and transmission, the motor controller output is always larger than the power at the wheel during times of acceleration. The two lower curves reveal how much power is being obtained from the battery and the APU. Since the APU is assumed to operate at its optimal design condition its output is constant throughout the simulation. The battery output fluctuates a little during the simulation until near the end of the simulation at which point the vehicle is no longer power-limited and the APU is capable of supplying all the power needs.

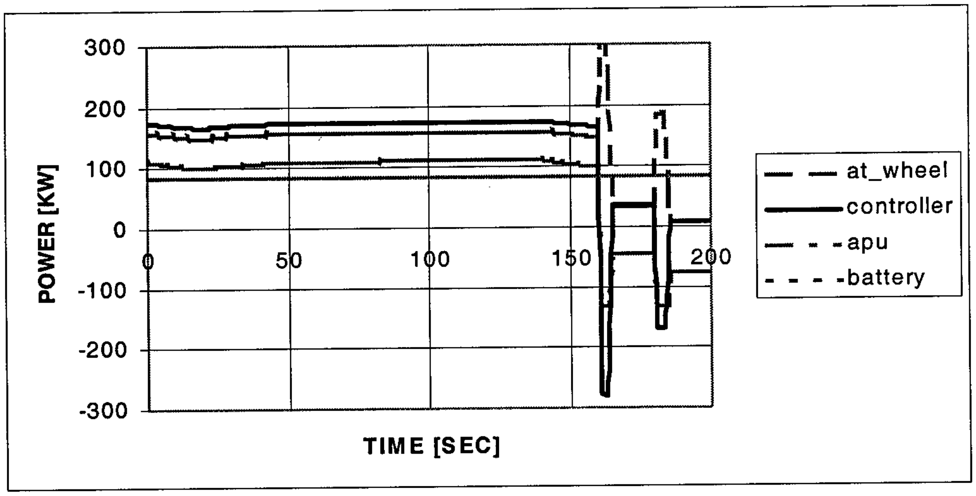

Figure 9. Power and Drive Train Simulation of the HMMWV

An example of the thermal analysis capability of the HEV Design Tool, shown in Figure 10 was generated using the electric BFV simulation. In the simulation, high speeds and road grades, both in standard and silent mode, placed a substantial load on the BFV. These high loads produced heat flows which raised the temperatures of the components, especially that of the APU. After these high loads, the vehicle was stopped which permitted component temperatures to fall. Due the heat capacities of the components, temperatures fell slowly. The APU exhibited the highest temperatures all components followed by the power components, the controller (pc_t) and DC bus (pd_t). Due to the short period of vehicle exercise, the transmission (r_t) and motor (tm_t) never became very hot.

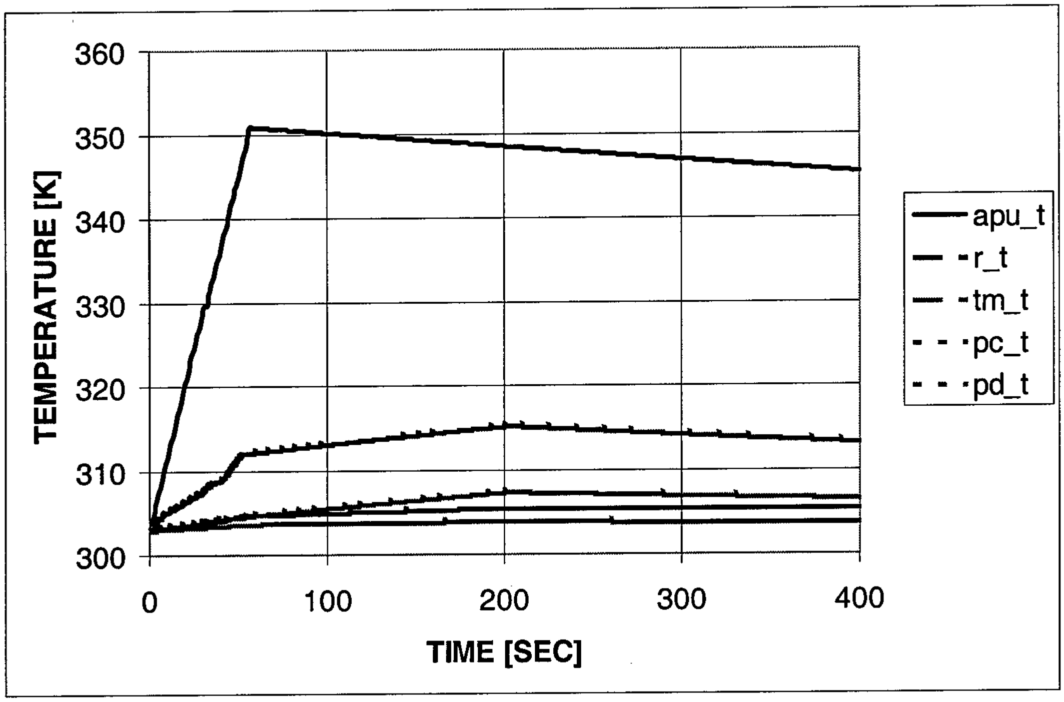

Figure 10. Predicted Component Temperatures for the BFV

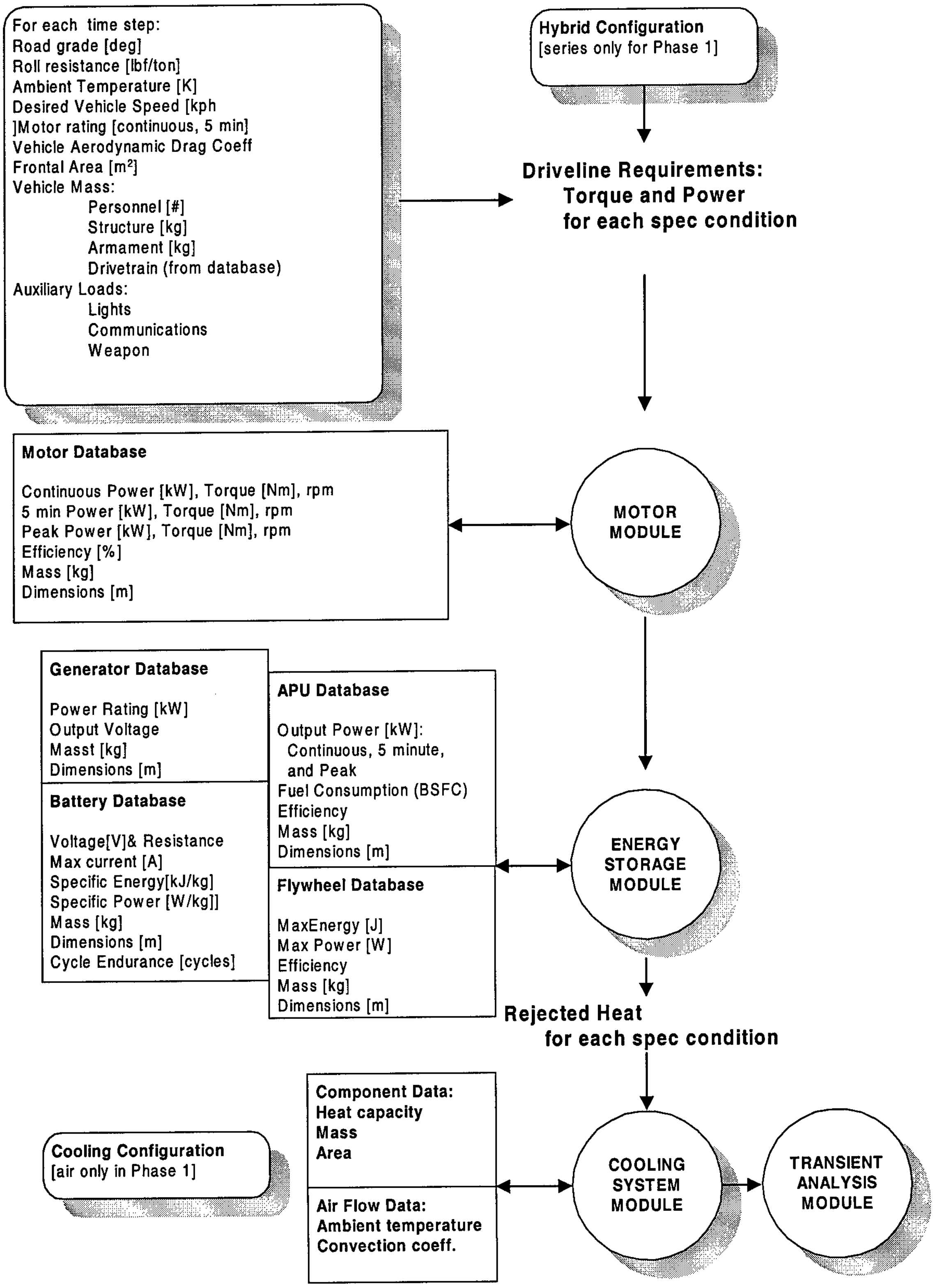

Figure 11. The Database Input Flow for the HEV Design Tool

3.7 Conclusions from the Phase I Work

There are two measures of success for the HEV Design Tool: accuracy of the simulation and utility as a design aid. The verification tests proved the accuracy of the Simulator, and also indicated the need for accurate specification data. We deliberately set the complexity of physics models employed in the Phase I tool, outlined in Figure 11, to be as simple as possible to see how far such a simple model could be used. Initially the Simulation assumed that the motor operated only at its continuous rating. During development we determined the need to add the "5 minute" rating to the simulation of the motor. Without this addition, the predictions did not match the verification data for accelerations from a dead stop – a condition under which the real-life motor would be run at a level above its continuous capacity. The trials also indicated areas where the simulation lacked fidelity:

1. To accurately predict performance on road grades, the Simulator needs to know how motor power and torque track with shaft speed;
2. To accurately predict component temperatures – and the effect that these temperatures have on component performance – a complete thermal model including radiation, convection, and radiation is required; additionally, a means of modeling and designing cooling loops is necessary for vehicle design purposes;
3. To predict the performance of components whose specifications are not totally known (a situation which proved to be the norm rather than the exception), the HEV Design Tool needs a robust method for interpolating missing specification data;
4. To model a variety of vehicles, a means of employing different power distribution strategies is needed (parallel and some series configurations will require throttling back and other APU power controls);

All of these factors are addressed in the Phase II plan.

The true test of the HEV Design Tool as a design aid would be its use within a design environment. Time did not allow for extensive testing during Phase I. Evaluation of the code by vehicle designers was positive, especially for the interactive feedback during the component selection process. To facilitate user tailoring and processing of the Simulator data, the main output file contained all simulation variables. While providing great flexibility, this approach is not time-efficient. Further work is needed to provide the Simulator with the ability to produce interactive outputs. Such feedback would yield a timely assessment of how well the vehicle design performed during the simulation. Better reduction of the output data into useful design information is addressed in the Phase II plan as well.

4.0 CODE USAGE INSTRUCTIONS

The HEV Design Tool is divided into two main sections: the Component Selector and the Simulator. In terms of installation and use these two sections are blended together seamlessly.

4.1 Code Installation

The HEV Design Tool consists of a number of Fortran source code files, a Fortran include file, and several sample input files including an ASCII form of the component database. To compile the code, place all Fortran source code files and the include file in a single directory, and link or "add to project" all the Fortran source code files (but not the HEV_COMMON include file). The code contains no graphical commands and uses only standard Fortran commands, thus making the code as platform independent as possible. The HEV Design Tool was developed and tested under DEC Visual Fortran 5.0.B using Microsoft Developer Studio 97.

4.2 Code Inputs

The code contains two main modules: the Component Selector and the Simulator. The Simulator can be run directly using a SEL file or can be run in conjunction with the Component Selector in which event both a SCN and VAT file will be needed.

All time-independent information about the vehicle is contained in the VAT (Vehicle dATa) file. The VAT file contains 8 lines of data in free format. The first line lists the vehicle's aerodynamic drag coefficient (non-dimensional) and the frontal vehicle area (m^2). The frontal area is used in the aerodynamic drag calculation. The second line contains the radius of the drive wheel or sprocket (m) and the wheel bearing torque drag (N*m). Mass (kg) of the personnel, vehicle body, armaments, and fuel are listed in the third line of the file. The next three lines contain the power requirements (W) and the mass (kg) of the lights, communications, and gun. The power control strategy charges the battery until the battery reaches its high state of charge (SOC) and allows the battery to drain until its low SOC is reached. The seventh line lists these low and high SOC limits (non-dimensional fraction) for the battery. The final line of the file lists the fraction of wheels that are directly connected to the drive shaft; this is the fraction of wheels available for regenerative braking.

The SCN (SCeNario) file, an ASCII free format file, lists all time-dependent input variables needed for the simulation. The first line of the file is a header which lists the variables. The variables are:

1. Time (seconds);
2. Desired speed (mph);
3. Road grade (degrees);
4. Roll resistance coefficient (lb_f/tons);
5. Coefficient of static friction between the tires/track and the road (non-dimensional);
6. Ambient temperature (K);
7. Air density (kg/m^3);
8. Wind speed (mph);
9. Wind direction relative to the vehicle (degrees);
10. Silent mode indicator (0 = APU can be on; 1 = APU must be off, silent mode).

The SEL (SELected components) file is created by the Component Selector of the HEV Design Tool. Although the user can generate this file, this procedure is not recommended. The usre can edit this column-delimited file, and its use in this way is recommended for it allows rapid perturbation of the component specifications and/or the driving cycle. The file is fully commented. Details of the file structure can be gained by viewing the Simulator_Main.For source code. Basically the file contains the complete listing of specifications of all selected components and echoes of the VAT and SCN files.

4.3 Running the Code

Running the code is easy since an explanatory prompt precedes all interactive user inputs. The first question the HEV Design Tool asks is if the user wishes to select components. Answering Y (Yes) activates the Component Selector while a N (No) answer bypasses the Component Selector and launches the Simulator in which case the user is asked for the name of the previously created SEL file which the user wishes to run.

Whenever the user inputs a filename, the complete filename, including the extension, must be typed in. The code will indicate a suggested file extension.

Since the Component Selector needs information about the vehicle being designed, such as its mass, and the scenario for which it is being designed for (ambient temperature and air density), the user is asked to input the names of the VAT and SCN files. At this point, the code starts to read the ASCII form of the database that is hardwired to be COMPONENT.DAT. The code displays information about each of the components in turn. Always the pertinent specifications of each component are displayed in a table. For the major components a second table is generated which contains predictions of how the vehicle will perform with the components installed. Necessarily these predictions involve numerous assumptions such as efficiencies of components not yet selected. Also, on the first iteration through the Selector, the code artificially increases the mass of the vehicle as contained in the VAT file to approximate the mass of the vehicle once all components have been added to it. On subsequent iterations through the SELECTOR (before a SEL file is created), the mass used in the performance calculations is the mass of the vehicle plus the mass of the components selected during the previous pass through the Selector.

The Selector predictions are calculated by equating the power required in performing the action with the power delivered to the shaft by the power and drive train components. The one exception to this rule is the selection of the transmission that deals instead with torque – the torque of the motor. By dealing with torque rather than power, the gear ratio comes into play. The maximum constant speed predictions are straightforward; the minimal time to accelerate from a dead stop to a specified speed adds the inertia term to the relation. The calculation of the steepest slope climb at a specified velocity balances the gravity term against the power available minus the other power terms (aerodynamic, wheel bearing, roll resistance). The range for the APU divides the fuel tank capacity by the fuel consumption rate and multiplies the resulting time by the speed. Since the APU is assumed to be operated at full power whenever it is switch on (its power cannot be varied), operating the vehicle at a velocity smaller than the maximum vehicle speed results in a waste of energy. As a result, the achievable range scales directly with vehicle speed.

Some of the components are selected by their capability with previously selected components. After the major drivers have been selected, such as the APU, motor, batteries, and transmission, the selection of inverters and motor controllers are based on matching the voltage, current, and power specifications with those of the major components. To facilitate this matching, the

Selector displays the pertinent specifications of the major components above the table of specifications for these components.

Each selection is preceded by an integer. Typing that number selects that vendor model for that component. Typing a zero bypasses the selection of that component. If the user types in an illegal number (less than zero or greater than the number of models for that component) the code will not accept the input and will ask the user again for his selection.

Once the Selector has run through the entire database, it then displays a performance summary. For this summary the vehicle mass is accurate – within each iteration it adds the masses of the components to the masses listed in the VAT file. The performance predictions listed depend only on the power limits of that component (transmission included). To predict the performance achievable if all power sources are utilized at the same time, summation entries (e.g. Battery + APU + Flywheel) appear at the bottom of the performance table. Also, the voltage range of each component is listed in a separate table.

At this juncture, the user can decide to go back through the Selector. If the user re-enters the Selector, the code does not remember anything about the selections already made, except that the mass sum of the previously selected components will be used in the performance calculations. Alternatively the user can elect to write the selected components, along with the corresponding VAT and SCN file data, to a SEL file and proceed on to the simulation. The code asks the user t type in a one-line description that will be used as the top line in the SEL file.

To run the simulation, the user must interactively input several variables. First the user must input the name of the main output file. We suggest using a TIME extension for this file since it is a time history of the simulation.

The SCN file contains a time history of desired speeds along with road conditions. The user is given the option of running any segment of that time history. Accordingly, the user must input the start and stop times for the simulation, along with the time step. All times are input in seconds. Due to the nature of the rapid acceleration of electric vehicles, a time step of one second is recommended.

The next two required initial conditions concern the initial state of charge (SOC) of the battery and flywheel (even though there is no flywheel in this version of the HEV Design Tool). The amount of fuel (kg) aboard the vehicle is input next. For convenience sake the code reports to the user the mass of fuel as recorded in the VAT file.

The next required input is the initial speed. If the first speed in the driving cycle is 20 mph, the user may intend that the vehicle begin the simulation at 20 mph or he may want the simulation to begin with the vehicle at a dead stop and then accelerate to 20 mph. To instill this flexibility of starting conditions, the user is asked to input the speed at which the simulation will begin. Instantaneously the code will begin to accelerate the vehicle to the speed specified in the driving cycle (if different from the input speed).

During the simulation, the code reports the time and speed at regular intervals. This display is intended to assure the user that the code is working and is not stuck in an infinite do-loop. The only occasion that will cause the code to abort the simulation is in the event that the vehicle begins to coast back down a hill that it is trying to climb. If this occurs, the code will report the grade of the road that it failed to successfully climb. The two other cases that would cause the failure of a real vehicle are if the vehicle runs out of battery charge and/or fuel. Since this is a design tool, the simulation continues, allowing the battery's state of charge and amount of fuel remaining in the tank to become negative. Through this means, the code informs the user how much more battery power and/or fuel is required to complete the assigned mission, as opposed to aborting and telling the user that an unknown amount of fuel needs to be added.

Lastly the user must input the name for the summary file. This file lists which components kept the vehicle from following the desired driving cycle.

4.4 Code Outputs

The code generates five output files. According to the debug level hardwired in the main routine (HEV_ALL_MAIN.FOR), the code generates a copious amount of information about the simulation in a file called DEBUG.OUT. The key feature to this file is that one can follow the course the code takes during each time step as it reiterates on the resulting vehicle speed for that time step based on available and requested power levels. Due to the potentially enormous size that this file can be, it is recommended that the debug level be set to zero. If the user does want some debug information, we suggest that the user customize the code to output those variables that he is interested in. For this reason, we hardwired the debug level – to activate this option the user must edit the source code and recompile, and hopefully he will take the time to edit the source code so that the resulting debug file is tailored to his needs.

The main output file (.TIME), whose name is specified by the user at the beginning of the Simulator, lists all the major simulation variables (power, torque, voltage, vehicle speed, shaft speed, state of components, etc) as a function of time. Each line in the file represents the state of the vehicle at that time step. The variable name and unit are listed in the top two lines of the file. Care must be taken when this file is imported into EXCEL because it is a column-delimited file; one must scroll throughout the file to insure that the column borders are correctly set.

The variable names are terse. The following lists all output variable names and what they represent:

1. **time:** Time (sec) – driving cycle time;
2. **des_sp:** Desired Speed (mph) – speed requested for that time step;
3. **speed:** Vehicle Speed (mph) – achieved vehicle speed at the end of that time step;
4. **accel:** Acceleration (mph/sec) – achieved acceleration during that time step;
5. **lapu:** APU state (zero/nonzero) – if = 0 then APU was off during that time step;
6. **w_pow:** Power at Wheel (kW) – power at the wheel/sprocket;
7. **r_pow:** Power at Transmission (kW) – power at the transmission-wheel interface;
8. **tm_pow:** Power at Motor (kW) – power at tractive motor-transmission interface;
9. **pc_pow:** Power at Controller (kW) – power output by motor controller;

10. **g_pow:** Power at Generator (kW) – power output by the generator – in this version of the Simulator this is the same as the power output by the APU;
11. **apu_pow:** Power at APU (kW) – power output by the APU;
12. **b_pow:** Power Out of Battery (kW) – if positive, power supplied by battery – if negative, power used to charge battery;
13. **f_pow:** Power Out of Flywheel (kW) – there is no flywheel in this version of the Simulator;
14. **a_pow:** Auxiliary Power Load (kW) – power required by the lights, communication equipment, and electromagnetic gun (as specified in VAT file);
15. **pd_pow:** Power Distribution (kW) – power out of the DC bus = power supplied by APU and battery – power used by the power converter supplying the drive train;
16. **s_tor:** Sprocket Torque (N*m) – power at sprocket – not used, for sprocket systems this torque is listed as w_tor;
17. **w_tor:** Wheel Torque (N*m) – torque at the wheel/sprocket;
18. **tm_tor:** Motor Torque (N*m) – torque required at motor output;
19. **b_vol:** Battery Voltage (V);
20. **b_soc:** Battery State of Charge (fraction);
21. **b_cur:** Battery Current (Amps) – if positive, battery is supplying current – negative while battery is being charged;
22. **fuel_kg:** Fuel Remaining in Fuel Tank (kg);
23. **milesgone:** Distance Traveled (miles) – distance traveled since start of simulation;
24. **rd_grade:** Road Grade (deg) – echo of road grade from driving cycle;
25. **roll_res:** Roll Resistance (lb_f/ton) – echo of roll resistance between wheel/track and road from driving cycle;
26. **stat_fric:** Static Friction – echo of static friction between wheel/track and road from driving cycle;
27. **gear #:** Transmission Gear – indicates which gear vehicle was in during time step, as numbered in database;
28. **pow_accel:** Inertia Power (kW) – power of inertia that vehicle must overcome;
29. **pow_grav:** Gravity Power (kW) – power caused by gravity while climbing hills that vehicle must overcome;

30. **pow_aero:** Aerodynamic Drag Power (kW) – power caused by aerodynamic drag that vehicle must overcome;

31. **pow_roll:** Roll Resistance Power (kW) – power caused by roll resistance that vehicle must overcome;

32. **pow_bear:** Wheel Bearing Drag Power (kW) – power caused by wheel bearing drag that vehicle must overcome;

33. **mass_veh:** Vehicle Mass (kg) – includes all masses, including the changing fuel mass;

34. **l5motor:** State of Motor (zero or nonzero) – if = 0 then motor is running at CONTINUOUS power level – if <> 0 then motor is temporarily operating at 5 MINUTE power level;

35. **isilent:** Silent Model Flag (zero or nonzero) – if = 0 then vehicle is not in silent mode – if <> 0 the vehicle is in silent mode and APU must be off.

Due to the overwhelming size of the main output file, a summary file is also created. This file attempts to guide the user to those components that should be beefed up. Every time the Simulator had to reduce the desired speed for a time step because a component limit was exceeded, it records that information. At the end of the simulation, the code processes this information and determines how often a component specification limited the vehicle performance. For each component that limited performance, the fraction of the scenario during which the limitation arose is listed (as a percentage of time) along with statistics about the vehicle motion when the limitation occurred (to cue the user if an excessive speed or steep grade triggered the limitation). Each line ends with the magnitude of that specification which would have completely avoided any reduction in requested vehicle speed.

Although this file is extremely useful, caution must be exercised in its use. The magnitude listed at the end of each line may be a momentary requirement; for instance, the power required at the first time step to accelerate the vehicle instantaneously from a dead stop to a high speed. Additionally, the user may have set up his driving cycle with an unreasonable speed, which is a convenient way of triggering the vehicle to go as fast as it can. This condition would result in a summary file that would indicate that the vehicle was extremely under-powered when it may not be at all. Lastly, one must remember the sequence that the Simulator steps through for each time step. It first calculates the power required at the wheel and then proceeds up the drive train. Consequently, the transmission can receive more "limitation hits" than the motor or APU that lie farther up the drive/power train.

Two additional files are created with the hardwired names of HEATFLOWS.DAT and TEMPERATURE.DAT. These two files contain the heat flows (W) and temperatures (K) of each of the components: r = transmission, tm = tractive motor, pc = power controller, pv = power converter for the motor, pd = power distributor/DC bus (modeled as bar of copper), g = generator, apu = APU, b = battery, and f = flywheel.

5.0 PLANS FOR PHASE II

The activities for Phase II are detailed in the Proposal. A list summarizing those activities that were derived from the Phase I study is outlined below.

5.1 Cooling Loop Design

Based on the thermal loads of the engine, drive train, and power components, the HEV code will lead the user through the design of the cooling system. The user will be able to select both air and liquid-based cooling loop and manifold designs. A 1-D fluid flow model currently being developed for WinTherm^2 will link the fluid transport with the heat transfer calculation. During the design process, the code will calculate pressure losses, heat exchange and dissipation efficiencies, fan loads, and cooling system mass and volume requirements.

5.2 APU Modeling

Accurate thermal and fuel consumption predictions of the APU are required for HEV design. Often the user does not have access to a complete set of specification data for the engines he wishes to analyze. To predict engine performance given a minimal set of specification data, the code will refer to a special APU database from which it will interpolate the required engine specification data not provided by the user. Based on this data the code will model engine power, torque, efficiency, exhaust gas mass flows and temperatures, fuel consumption, and heat loads for diesels, gasoline, and turbine engines.

5.3 Mobility

Vehicle motion will be computed using a complete mobility analysis. The code will also predict cornering speed/curvature, max side slope, max drawbar pull, and max climbing performance. In the case of mobility failure, the code will guide the user through the various solution options.

5.4 Additional Databases

We will expand the existing component database. We will create additional databases containing driving cycle, weather, terrain, and vehicle data. To facilitate handling varying user needs, the code will be revised to handle driving cycles based on either time or distance, and the driving cycle file format extended to include the loading and unloading of personnel, cargo, and munitions.

5.5 Motor Model

Through consultation with industry, we will upgrade the motor model so that the power and torque will be functions of shaft speed. Another feature of this enhanced motor model will be accurate thermal load predictions.

5.6 Vehicle Thermal Model

The user will be able to plug-in thermal links through a database listing of options. The links detail how heat generated by the components are transferred to cooling loops, dissipated through convection and radiation via the casing, and conducted to the vehicle chassis through the component mounts. Based on user input the components will either be contained within closed air volumes, with or without forced airflow, or be exposed to the outside environment.

5.7 *Code Validation*

The contractor will participate in the electric vehicle tests being conducted at Aberdeen Proving Ground. The data collected through this series of tests will form the basis of the validation of the HEV code.

5.8 *Improved Selector*

The Component Selector Code Module will be upgraded to:

1. Edit the .SEL files (as opposed to only creating them);
2. Adapt the performance criteria (e.g. roll resistances, speeds, and grades used for performance predictions) automatically for various types of vehicles;
3. Modify all input and output files, including the databases, to be ASCII comma-delimited to facilitate the import and export of the files to spreadsheet programs and other applications.

5.9 *Battery Modeling and Selection*

We will upgrade the battery model to include the dependence of battery performance on temperature. The code will allow for both the selection of both pre-arranged battery packs and the creation of packs from individual batteries.

5.10 *Power Control Module*

Currently only one power strategy (e.g. should the power be obtained from the battery, APU, or both?) is available in the HEV code. With this new feature, the user will select the desired control strategy from a collection of different plug-ins. Additionally the user will be able to write his own power strategy subroutine and hook it into the HEV code.

5.11 *WinTherm Interface*

The existing HEV code will be upgraded in its present stand-alone FORTRAN form; then, to improve and expand its capability, it will be interfaced with the WinTherm[6] thermal solver. The link with WinTherm will provide several key capabilities:

1. Visualization of the vehicle design – the 3D vehicle and component shell descriptions will permit the rapid calculation of mobility inputs such as center of gravity (cg) location, and will serve as a visual cue to the volumes that the components take up relative to space available in the vehicle;
2. "Plug-in" cooling loops – the GUI will permit the graphical insertion, scaling, and linking together of cooling loops;

---

[6] WinTherm/MuSES is being developed under a US Army TACOM Phase II SBIR. The commercial version is WinTherm and is used for heat management design. The military version is MuSES and is used for the related signature management design and analysis purposes. This SBIR program is being cost-shared by Ford Motor Company in which common features are being added to Ford's version of the code as well as to the government funded programs. ThermoAnalytics is also cost sharing the development of WinTherm for commercial use.

3. "Plug-in" and visualization of thermal links – the user will have complete control and flexibility in creating the thermal links, and will be able to verify the placement of the thermal links graphically;
4. Visualization of temperatures – WinTherm displays the temperature distribution over the vehicle and components as a function of time in an animation that can be played using VCR-like controls;
5. Continual improvements in thermal modeling – since WinTherm is a commercial thermal solver continually being upgraded (see note 2 below), and since the HEV will be a "plug-in" to WinTherm, the resulting HEV-WinTherm code will automatically benefit from the upgrades to WinTherm's thermal solver and design features;
6. File imports and exports – WinTherm can import and export vehicle geometries in a number of different file formats including PATRAN, NASTRAN, DXF, STL (stereo-lithography), Wavefront OBJ, and VRML for input into CAD and visual simulators. WinTherm can also create the necessary geometry-related files for infrared and radar analysis codes;
7. Easy customization of the interface– including the graphical display (e.g. plotting) of both input and output data, and user-customization of variables included in the output files;
8. Unlimited thermal modeling – whereas the stand-alone code will be limited to a set of simplistic thermal links and models, the HEV-WinTherm combination will have access to the full thermal modeling capability of the commercial thermal design code, WinTherm. The thermal vehicle model within WinTherm can remain simple and basic or be expanded to be a complete thermal model of the vehicle and subsystems;
9. Portability – the WinTherm GUI is cross-platform, available for both PCs (Win95 and NT) and UNIX workstations; the GUI's appearance and functionality is nearly totally identical between the platforms.

5.12 System Architecture

The Phase II code will consider other HEV system architectures beyond the series configuration considered in Phase I. Parallel configurations and variations thereof will be modeled in the Phase II code.

5.13 Option to Extend to Conventional Vehicles

As a possible option, the HEV code could be extended to model conventionally powered vehicles. This capability would allow for the direct comparison of hybrid electric vehicles with their conventionally powered counterparts – under identical simulation conditions.

APPENDIX A. PHASE I MOBILITY RESEARCH

A.1 Background

This report begins with an explanation of the NATO Reference Mobility Model (NRMM). It describes several parts of NRMM, shows some general flow diagrams, describes most of the vehicle parameters and discusses how the terrain data is input and processed. This is done in an attempt to explain how detailed NRMM is and how complex a mobility model for the HEV SBIR could become.

A.2 Introduction

The NATO Reference Mobility Model (NRMM) consists of three separate modules, VEHDYNII, OBS78 and NRMM. A simplistic flow diagram is shown below.

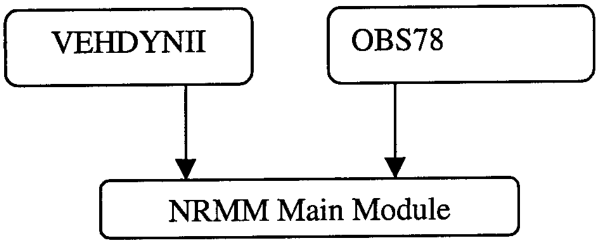

Figure A-1. Simplistic Flow Diagram of NRMM

VEHDYNII is a stand-alone ride and shock quality analysis program. Input for VEHDYNII consists of vehicle weight, suspension data and terrain data. The terrain data is divided into two sections. The first is a set of road roughness profiles for 6 watt absorbed power criteria for the driver. These profiles range from 0.19 to 4.0 inch rms surface roughness. The program calculates the maximum speed the vehicle can attain while traveling over each course before the driver is subjected to 6 watts of absorbed power. The second is a set of single discreet half-round obstacles ranging in height from 4 to 18 inches in 2 inch increments. The vehicle is run over each half-round separately and the speed is incremented from 5 to maximum vehicle rated speed until the 2.5 g vertical acceleration limit is reached. The output of VEHDYNII consists of two curves, one set of vehicle speeds and surface roughness data, and the other a set of vehicle speeds and obstacle heights.

It is recommended for the HEV Design Tool that a known set of curves be used. For example, data for each type and size of vehicle can be stored and when the vehicle type is entered the program could take the appropriate data from the data file. The suggested categories could be automobile, light truck (pickup-CUCV), medium truck (2 ½ ton to 5 ton), heavy truck (10 ton), light tracked vehicle, medium tracked vehicle, and heavy tracked vehicles. This data could then be directly input into running the mobility portion of the code. It is not felt that the ride quality will significantly affect the amount of engine power required to cross a certain terrain.

OBS78 is the obstacle-crossing module and is also a standalone program. It simulates the placement of the vehicle at a sequence of positions across the obstacle and for each position calculates the following:

1. The tractive forces under the running gear to maintain that position;
2. The clearances/interferences between the frame of the vehicle and the obstacle at that position;
3. Selects the maximum interference (or minimum clearance, if there is no interference) and the maximum tractive effort and calculates the average tractive effort across the various positions.

The obstacles are standard trapezoidal shapes. It is recommended that the Phase II HEV Design Tool calculate of the energy required to cross various shaped obstacles because this energy value is an important mobility design parameter.

The output table of OBS78 is used as input into the NRMM main module. The flow chart for running the main module, which is similar to what is recommended for the Phase II HEV Design Tool, is shown in Figure A-2.

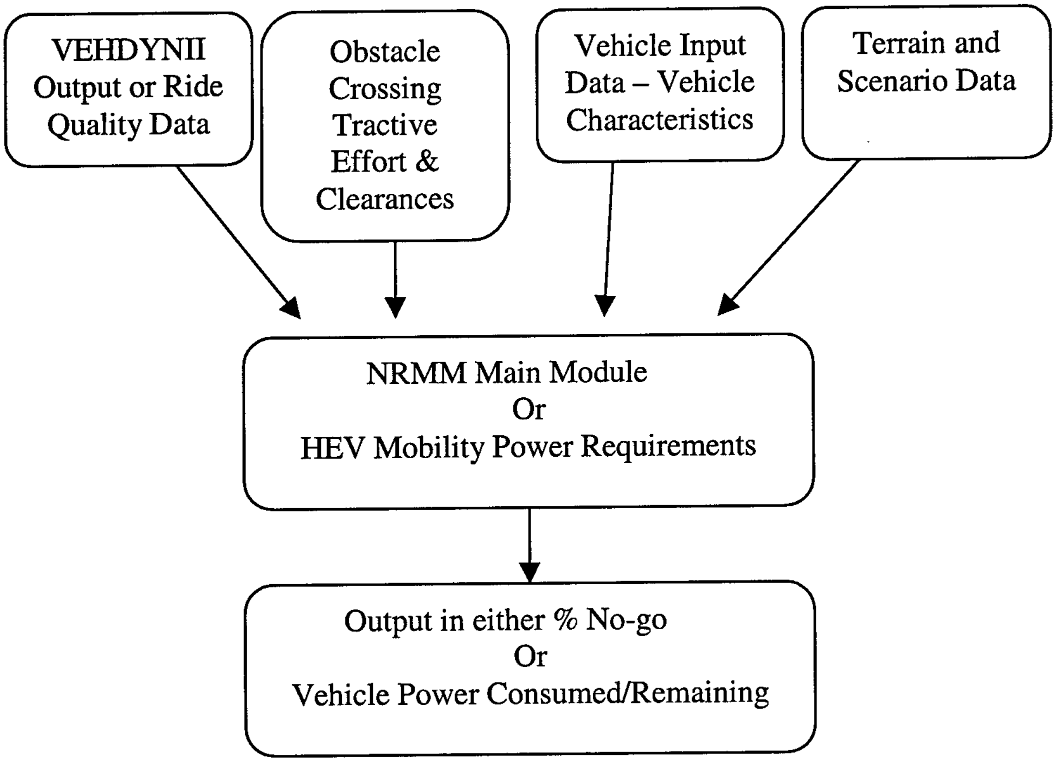

Figure A-2. Data Flow Between Mobility Modules in NRMM

NRMM is a two dimensional model that predicts vehicle mobility over certain on and off-road terrain. It considers each wheel station or axle assembly an "assembly". Therefore, for a 6 x 6 truck there are three assemblies, not six.

A.3 Vehicle Parameters Used in NRMM

1. Absorbed Power level for each ride limit curve (watts) [not used in Phase I and a modified version is recommended for Phase II]
2. Aerodynamic drag coefficient (lb/mph^2)
3. Area of 1 track shoe (in.^{2}) [for tracked vehicles only - may use track width multiplied by sprocket pitch if no particular track is identified]
4. Interaxle spacing (I to I+1)
5. Hydrodynamic drag coefficient if swimming the vehicle (lb/mph^2) [may be omitted from HEV analysis]
6. CG height above the ground of the loaded vehicle (in.)
7. CG lateral distance from vehicle centerline (in.)
8. CG horizontal distance from rear axle (in.)
9. Minimum ground clearance for each assembly (in.) [ground to bottom of axle housing]
10. Minimum ground clearance (in.) [everything but axle assembly which is usually the transmission, transfer case or vehicle hull]
11. Torque converter torque-multiplier versus input speed (rpm) [output engine speed]
12. Torque converter torque-multiplier versus speed – ratio
13. Tire deflection for each assembly and each deflection case (in.) [based on air pressure]
14. Undeflected tire diameter for each assembly (in.)
15. Engine speed (rpm) versus engine torque (ft-lb)
16. Driver's eye height above ground (in.) [may not be needed for HEV code]
17. Final drive gear ratio and efficiency
18. Track grouser height for each assembly (in.) [measured from track shoe/track pad interface to the outside of track pad surface]
19. Total net engine power for all engines (hp) [used for hp/ton calculations]
20. Vertical distance from vehicle roll center to the axle (in.) [may not be needed for HEV - used for side slope tipping analysis]
21. List of axle assemblies that are braked [or not]
22. Tire type construction code [radial or bias]
23. List of axle assemblies that have dual wheels/tires [or not]
24. List of axle assemblies that are powered [or not]
25. List of axle assemblies that are part of a tandem axle
26. Type of transmission [automatic or manual]
27. List containing obstacle height versus speed for 2.5 g's [for HEV Design Tool input a canned list based on type of vehicle and size instead of a VEHDYN-like calculation]
28. Transmission operating range for each type of terrain scenario, i.e., primary roads or cross-country, etc. [may not be needed for HEV]
29. Tire stiffness code [from no stiffness to very stiff]
30. Vehicle ride table of speed versus surface roughness course and 6 watts absorbed power [for Phase II input a canned list based on type of vehicle and size instead of calculating in VEHDYN-like code]
31. Does differential lock – each axle assembly?
32. Does torque converter lock?
33. Total number of track road wheels for each track assembly [both sides of vehicle)]
34. Does vehicle have tire chains? [do not need for HEV]
35. Number of engines [some vehicles have more than one]
36. Number of transmission gear ratio's for each range
37. Number of tire inflation/deflection cases
38. Does vehicle have track pads?
39. Number of suspension springs per side.
40. Number of transmission operating ranges.
41. Number of tractor trailer units in vehicle combination [NRMM is designed for vehicles that tow trailers and even vehicles that have a powered prime mover and powered trailer]
42. Is the vehicle wheeled or tracked?
43. Number of tires on each wheeled assembly
44. Maximum pushbar force vehicle can withstand overriding vegetation stems (lbs) [this is usually set equal to max vehicle drawbar pull force available for best terrain]
45. Height of pushbar above ground (in.).

46. Vehicle projected frontal area (ft^2)
47. Vehicle speed versus tractive force for each range (mph, lb) [this is of primary importance and is what is normally given on a scan sheet from engine or transmission company]
48. Maximum net torque available from engine
49. Mean stiffness of suspension springs
50. Tire rim diameter for each wheeled assembly (in.).Tire revolutions per mile for each assembly (rev/mi.)
51. Tire rim width for each wheeled assembly (in.)
52. Effective radius of track roadwheels (i.e., roadwheel radius + track thickness).
53. Vehicle swamp angle during egress and ingress (degrees)
54. Tire nominal undeflected section height (in.)
55. Tire nominal section width (in.)
56. Engine torque converter gear ratio and efficiency [usually use 1.0 for both if direct coupled]
57. Distance from center of first wheel to center of last wheel (in.)
58. Tire ply rating for each wheeled assembly
59. Tire inflation pressure (psi)
60. Torque input used for torque converter input speed versus speed ratio data relation
61. Length of track on the ground on one side (in.)
62. Track width (one side) for each track assembly
63. Transmission gear ratios & efficiencies for each transmission gear and range
64. Length of each vehicle unit (in.)
65. Maximum combination vehicle width (in.)
66. Weight beneath each vehicle assembly (lb.)
67. Tread widths (center to center across vehicle) (in.)
68. Minimum width between traction elements (in.)
69. Combination vehicle braking coefficient.

A.4 *Vehicle Parameters Used in Obstacle Crossing Module*

1. Number of vehicle units
2. Total number of suspension supports for entire vehicle
3. Tracked or wheeled vehicle
4. Track type – rigid or flexible
5. Height of hitch above the ground when empty vehicle is at rest (in.)
6. Vertical force on hitch at rest (tongue weight) (lbs.)
7. Type of suspension, [independent or bogie]
8. For each axle assembly is wheel powered?
9. For each axle assembly is wheel braked?
10. Effective loaded radius of wheels at each support, i.e., the distance from the wheel centers to the contact point (including track thickness for a tracked vehicle) (in.)
11. Horizontal coordinate (longitudinal distance) of suspension support point I with respect to hitch (in.)
12. Bogie swing arm width at each support (in.)
13. Limit of angular movement in counter clockwise direction of bogie arm at support I (deg.)
14. Limit of angular movement in the clockwise direction of bogie arm at support I - this angle is negative if the front wheel is below the rear wheel at the extreme position (deg.)
15. Equilibrium load on each assembly support when vehicle is empty and at rest (if the support is a bogie it is the sum of the loads on both axle assemblies) (lbs.)
16. Vertical position from ground to center of gravity for unloaded vehicle (in.)
17. Vertical position from ground to center of gravity for unloaded second vehicle (in.)
18. Horizontal coordinate (longitudinal distance) of the first unit payload cg with respect to hitch (in.)
19. Vertical distance to the cg of the payload of the first unit from the ground at rest (in.)
20. Horizontal coordinate (longitudinal distance) of the trailer payload cg with respect to hitch (in.)
21. Vertical distance to the cg of payload of the second unit from the ground at rest (in.)
22. Weight of the payload of the first unit (lb.)
23. Weight of the payload of the second unit (lb.)
24. Table of coordinates representing the height from the ground of lowest vehicle points and distance from the hitch of these "break" points. From hitch to front of vehicle is positive and from hitch to second vehicle is negative (in. and in.). This table thus contains the bottom profile of each unit, including the vehicle frame, transfer case and transmission but not the axle assemblies.

25. Indicate suspension type, powered or unpowered and braked or not braked for front and rear spridler
26. Horizontal coordinate (longitudinal distance) of center of front spridler with respect to hitch (in.)
27. Vertical distance from ground to center of front spridler (in.)
28. Effective radius (distance from wheel center to contact point including track thickness of front spridler (in.)
29. Horizontal coordinate (longitudinal distance) of center of rear spridler with respect to hitch (in.)
30. Vertical distance from ground to center of rear spridler (in.)
31. Effective radius of rear spridler (in.)

A.5 Terrain Data Input Format

The following describes the terrain data input formats for NRMM for both off-road and on-road terrain. The HEV Design Tool need not be as detailed as NRMM. Phase I may only require predicting wheel torque and speed on a hard road such as a paved road or a hard secondary gravel road.

For the current version of NRMM, vehicle (wheel) force and speed are predicted and used to generate the final outputs. The purpose of NRMM is to predict vehicle speed and go or no-go capability over a specifically described piece of terrain. The code that will be used for the HEV SBIR program is slightly different in that we want to predict the force and speed required to traverse a piece of terrain. The force can be converted to a torque at the wheel. The torque and speed can then be taken back to the power plant of the vehicle to determine the amount of power output required.

A.6 Off-Road Terrain Input Format

Shown below are the first three lines of a typical Off-Road Terrain Input File for NRMM. The first line is a description of the terrain quadrant. The first number (1879) represents the total number of terrain units in the terrain quadrant. The second number (5) means that it is in the MAP90 format, which is now used most of the time for NRMM II. The rest of the first line is an alphanumeric description of the terrain (in this case somewhere in the Middle East).

The next two lines are a series of numbers and characters describing the shape and characteristics of the first terrain unit.

First Line

The first number (1) is the number of the terrain unit; in this case it is the first terrain unit. The second number (4) denotes that this is off-road terrain. The third number (1) denotes the surface condition of the terrain, in this case a normal surface. The fourth number (0) denotes the surface cover depth in inches. In this case water or snow does not cover the surface. If the number in the third location is a 3 or 4 then a depth is normally indicated for water or snow, respectively. The fifth location (CL) denotes the USCS soil type code for the terrain unit. The sixth location (0) denotes the land use code, in this case it is unknown. The seventh location (4) denotes the "wetness" index code, which in this case means that it is saturated or flooded part of the year. The following eight numbers denote soil strengths for the 0-6 and 6-12 inch layers during the dry, average, wet and wet-wet seasons and are CI or RCI units probably determined through the use of a cone penetrometer. The sixteenth location (99) denotes the depth of soil to bedrock in inches, and anything over 12 inches is not significant. The 17th location (14) is the grade of the slope in percent in the terrain unit. The 18th location (24) is the surface roughness in the terrain unit in RMS-in multiplied by 10 (therefore, the surface roughness is 2.4 inches rms). The 19th location (0.42) is the amount of area in this particular terrain unit, in km^2.

Second Line

The first location (139) is the obstacle approach angle in degrees. The second location (12) is the obstacle height in inches. The third location (33) specifies the obstacle's base width in feet. The fourth location (3) is the obstacle's length, in feet. The fifth location (115) is the obstacle spacing in feet. The sixth location (1) specifies the obstacle spacing type, in this case it is random or potentially avoidable. The seventh through 14th locations describe the average stem spacing for various classes and sizes of vegetation and the units are in feet. The last four locations denote the visibility distance for each of the four quarters of the year, i.e., the first is January through March, etc., in feet.

Off-road Terrain Title Line and First Unit Descriptors

```
1879 5 QUAD-3254C (MAFRAQ) ( NRMM90 format )
1 4 1   0   CL  0 4 219 216 183 167 219 216 183 167  99  14  24     0.4200
139  12  33   3 115   1  31  31  66  66 328 328 328 328  21 273 275 133
```

A.7 On-Road Terrain Input Format

Shown below are the first three lines of a typical On-Road Terrain Input File for NRMM. The first line is a description of the terrain quadrant. The first number (602) represents the total number of road units in the terrain quadrant. The second number (6) means that it is in the MAP90R format, which is now used most of the time for on-road NRMM II analysis. The rest of the first line is an alphanumeric description of the terrain (in this case somewhere in the Middle East).

The next two lines are a series of numbers and characters describing the shape and characteristics of the first terrain unit.

First Line

The first number (1) is the road unit segment ID number, in this case the first road unit of a total of 602 road units. The second number (4) is the urban code, i.e., village, town, city, etc., in this case 4 indicates off-road (a trail). The third number (0) indicates the surface condition and in this case it is unknown. The fourth location (0) indicates the surface cover depth in inches if the third number is a 3 or 4, indicating water or snow depth, respectively. In this case there is nothing covering the terrain. The fifth location (SMSC) indicates the soil type by USCS code type. The sixth location (0) indicates the land use code, which in this case is unknown. The seventh location (3) indicates the wetness-index; in this case the roads and trails are wet, poorly drained soils and bottomlands. The eighth through 15th locations indicate the soil strength for different surface/scenarios such as dry, average, wet and wet-wet seasons, the first four indicating these from 0-6 inches and the last four for 6-12 inches below the surface. The numbers are in CI or RCI determined by a cone penetrometer. The 16th location (99) indicates the depth to bedrock in inches and anything over 12 is not significant. The 17th location (1) indicates the slope in percent grade. The 18th location (3) indicates the surface roughness in rms inches * 10 so in this case the surface roughness is 0.3 inches rms. Note that most on-road terrains do not have a high surface roughness. The 19th location on the first line (2.4224) is the road segment distance in miles, in this case it is 2.4224 miles long.

Second Line

The first location on the second line (4) displays the road type code of which there are eight choices. The 4 indicates a main road. The second location (2) is the same as the first only in the TRADOC definition of which there are four choices with the 2 indicating a primary road. The third location (1) indicates the type of material that covers the road surface, in this case it is soil. Because (1) is specified the next two numbers describe the soil condition, the (9) being a USCS type CH and the (1) meaning the soil is dry but there may be steep slopes such as those found in semi-arid regions. The seventh location (0) describes the radius of curvature in feet. The eighth location (0) describes the super-elevation slope in percent. The following four locations describe the visibility for each of the four quarters of the year, as described in the Off-Road Terrain section. The distance is in feet. The 13th location (2) describes the number of road lanes present in the terrain unit. The 14th location (10) specifies the road lane width in feet. The 15th location (10) specifies the shoulder width in feet. The 16th location (3) describes the drainage feature, in this case a ditch greater than 3 feet deep is present. The 17th location (0) describes if a bridge is present, in this case there is no bridge. The 18th location (0) describes if there is a tunnel present, in this case there is no tunnel.

On-road Terrain Title Line and First Unit Descriptors

```
602 6 STATISTICAL ON-ROAD DATA (PRI & SEC 3254III, TRAIL 3254IV) 2 (NRMM90R
format)
1 4 0    0 SMSC  0 3 300 300 300 300 300 300 300 300  99   1   3    2.4224
4 2 2 1 9 1 0 0 300 300 300 300 2 10 10 3    0    0
```

A.8 Terrain Preprocessor

In NRMM there are two separate subroutines that deal with the terrain, the terrain description and the scenario description. The scenario is used to describe the condition of the terrain, i.e., if it is dry, sandy, wet, wet-wet (saturated) or covered with snow. Note that the current version of NRMM only has a shallow snow model. As part of setting the model up to run, the user runs a file that sets certain conditions. For example, the user may specify an on and an off-road terrain in Germany. The user would also specify the scenario. Much of the time it is easiest to specify two or three different scenarios such as dry, wet and snow covered.

The terrain preprocessor combines the terrain module specified with the scenario information and determines a number of factors that are used in predicting vehicle mobility. It gives certain soil conditions in each unit a slipperiness factor, calculates the grade of slope, applies soil strengths, and sets a number of flags that affect the NRMM calculations.

Once these are determined the vehicle and terrain/scenario data are combined such that all of the vehicle calculations are computed to predict vehicle speed over each terrain unit. The result is a percent no-go and vehicle speed for a given vehicle and terrain module.