

**VALIDATION OF A VEHICLE PERFORMANCE MODELLING
SYSTEM**

Russell Lloyd Herbert

Submitted in partial fulfilment of the requirements
for the degree of MSc Engineering

School of Bioresources Engineering and Environmental Hydrology
University of KwaZulu-Natal
Pietermaritzburg

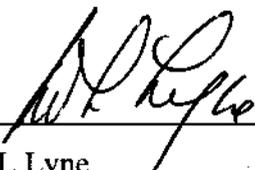
2007

DISCLAIMER

I wish to certify that the work reported in this dissertation is my own original and unaided work except where specific acknowledgement is made. Additionally I wish to declare that this dissertation has not been submitted for a degree in any other university.

Signed:  _____ Date: _____
R L Herbert

As the candidate's supervisor I have approved this dissertation for submission.

Signed:  _____ Date: 3/5/07
P W L Lyne

ACKNOWLEDGEMENTS

The author wishes to express his sincere and heartfelt appreciation for the assistance given by the following individuals and organisations:

Professor Peter W L Lyne, South African Sugarcane Research Institute (SASRI), for his supervision of this project, sage advice, encouragement and assistance throughout the project.

Mr Robert van Zyl, School of Bioresources Engineering and Environmental Hydrology, University of KwaZulu-Natal, for his advice, technical assistance and supervision during the development of the data logging system and the vehicle performance testing.

Mr Alan Hill, School of Bioresources Engineering and Environmental Hydrology, University of KwaZulu-Natal, for his help and technical assistance in the vehicle performance testing and the development of the data logging system.

The Department of Transport for funding my studies.

The University of KwaZulu-Natal for their financial assistance and the use of their laboratories and equipment for this project.

Unitrans Freight Sugar and Agriculture, Eston Depot, for the use of their driver and Mercedes Benz 2637 truck during the vehicle performance tests. Thanks in particular to Mr Richard Linklater and Mr Don Mackenzie, for their time and co-operation.

Mr James Wood of Greenbelt Mapping for surveying the two routes.

Mr Richard Borain, Forest Services, for the use of one of their haulage vehicles during the testing and verification phase of the data logging system.

My family and friends, especially my parents, for their encouragement and unending support during my studies.

ABSTRACT

Transport costs can account for a significant portion of the total production costs in an agricultural system. In order to ensure maximum efficiency of a transport operation, the transport manager is required to select a suitable vehicle and choose the best route. This is a complex process as vehicles should not be selected based solely on lowest capital cost, but with several variables taken into account.

In order to determine an optimal haulage system and to aid the decision making process, a computer model called SimTrans has been developed. SimTrans simulates a haulage vehicle travelling on any given route under normal vehicle operating conditions. The model produces a set of simulated vehicle performance results and a summary of trip details. The vehicle input parameters and route details can be modified to enable the user to simulate different vehicle configurations on alternate routes. The SimTrans model was used in this research project as it provides a comprehensive output of simulated results. However, under certain conditions there is variance between actual and the simulated vehicle performance in SimTrans, particularly with the gear selection routines. The primary objective of this research was to refine, calibrate and validate the current SimTrans model.

An initial investigation focused on diesel engine management systems for heavy-duty commercial vehicles and how these systems affect a vehicle's performance. This investigation was aimed at gaining an understanding of how performance data could be gathered directly from the onboard vehicle electronics of a haulage vehicle. A data logging system was developed to measure and record certain operating parameters of a haulage vehicle during operation. The system was installed on a Mercedes Benz 2637 and the vehicle was driven on two alternative routes between the Eston Sugar Mill and a loading zone near Mid-Illovo.

The Mercedes Benz 2637 used for the vehicle tests was simulated in SimTrans using identical operating conditions of route and speed limits. By comparing the observed and simulated data sets, errors particularly with gear selection routines were identified. The SimTrans model was modified such that these errors were corrected and a new gear skipping routine

was added. The modified SimTrans model was validated through a comparison of a new set of simulated results with the observed data. The modified model showed significant improvement in simulation accuracy over the previous version of SimTrans.

Keywords: SimTrans, vehicle performance modelling systems, vehicle performance logging, engine management systems, electronic fuel injection, diesel engine, electronic control, on-board systems, monitoring engine performance, emission control.

TABLE OF CONTENTS

| | |
|--|------|
| LIST OF TABLES | viii |
| LIST OF FIGURES | ix |
| LIST OF ABBREVIATIONS | xii |
| 1. INTRODUCTION | 1 |
| 2. ENGINE MANAGEMENT SYSTEMS | 4 |
| 2.1 Merits of Electronic Control Systems | 5 |
| 2.2 Powertrain Control Modules | 8 |
| 2.2.1 Map-Based Control Systems | 10 |
| 2.2.2 Real-Time Model-Based Control Systems | 11 |
| 2.3 Electronic Fuel Injection Systems | 12 |
| 2.3.1 Operation | 13 |
| 2.3.2 Fuel Pumps | 13 |
| 2.3.3 Time-Controlled Single-Cylinder Pump Systems | 14 |
| 2.3.3.1 Unit Injector System | 14 |
| 2.3.3.2 Unit Pump System | 15 |
| 2.3.4 Common Rail System | 15 |
| 2.4 Features of Engine Management Systems | 16 |
| 2.4.1 Engine Starting | 16 |
| 2.4.2 Electronic Governor | 17 |
| 2.4.3 Emission Control | 17 |
| 2.5 Sensors | 18 |
| 2.5.1 Smart and Integrated Sensors | 18 |
| 2.5.2 Chemical Sensors | 19 |
| 2.5.3 Optical Sensors | 19 |
| 2.5.4 Knock Sensor | 21 |
| 2.5.5 Inductive Speed and Position Sensors | 21 |
| 2.5.6 Temperature Sensors | 22 |
| 2.5.7 Speed Control Lever Position Sensors | 22 |
| 2.5.8 Inertia Switch | 23 |
| 2.6 Networking Vehicle Electronic Systems | 23 |
| 2.7 Discussion | 25 |

| | | |
|---------|--|----|
| 3. | VEHICLE PERFORMANCE MANAGEMENT SYSTEMS | 27 |
| 3.1 | TransSolve | 27 |
| 3.2 | SimTrans | 29 |
| 3.2.1 | User Interface | 30 |
| 3.2.2 | Model Structure | 32 |
| 3.2.2.1 | AirResistance Function | 33 |
| 3.2.2.2 | GradeResistance Function | 34 |
| 3.2.2.3 | RollingResistance Function | 35 |
| 3.2.2.4 | GetMaxEngTorque Function | 36 |
| 3.2.2.5 | GetTorqueLosses Function | 36 |
| 3.2.2.6 | GetBrakeTorque Function | 37 |
| 3.2.2.7 | SelectGear Function | 37 |
| 3.2.2.8 | GetFuel Function | 38 |
| 3.2.2.9 | Main Procedure | 39 |
| 4. | MEASUREMENT OF VEHICLE PERFORMANCE | 43 |
| 4.1 | Development of a Vehicle Performance Data Acquisition System | 43 |
| 4.1.1 | Fuel Flow Rate Measurement | 44 |
| 4.1.1.1 | Flow Meter Calibration | 44 |
| 4.1.1.2 | Flow Meter Digital Signal Conditioning Board | 45 |
| 4.1.1.3 | Fuel Flow Rate Measurement System Testing | 45 |
| 4.1.2 | Throttle Position Sensor | 46 |
| 4.1.3 | Global Positioning System Receiver | 47 |
| 4.1.4 | Speed Measurements | 48 |
| 4.1.5 | Monitoring System Software Development | 49 |
| 4.1.6 | Monitoring System Testing and Installation | 53 |
| 4.2 | Haulage Vehicle Performance Measurements | 55 |
| 4.3 | SimTrans Simulation | 56 |
| 5. | RESULTS AND DISCUSSION | 59 |
| 5.1 | Comparison of Simulated and Observed Data | 59 |
| 5.2 | SimTrans Modifications | 68 |
| 5.2.1 | Main Procedure | 68 |
| 5.2.2 | AccelerationCalc Procedure | 71 |
| 5.2.3 | SelectGear Function | 72 |
| 5.3 | Validation of Modified Model Accuracy | 75 |
| 5.4 | Further Proposed Solutions | 85 |
| 6. | CONCLUSIONS AND RECOMMENDATIONS | 87 |
| 7. | REFERENCES | 89 |

| | | |
|----|---|-----|
| 8. | APPENDICES | 94 |
| | APPENDIX A – DEFINITIONS | 94 |
| | APPENDIX B – FLOWCHARTS FOR THE ORIGINAL SIMTRANS MODEL | 95 |
| | APPENDIX C – FLOW METER CALIBRATION | 107 |
| | APPENDIX D – FUEL FLOW RATE MEASUREMENT SYSTEM TESTING | 109 |
| | APPENDIX E – ENGINE SPEED CALIBRATION | 110 |
| | APPENDIX F – VEHICLE SPEED CALIBRATION | 111 |
| | APPENDIX G – FLOWCHARTS ILLUSTRATING THE CR10X PROGRAM CYCLE | 112 |
| | APPENDIX H – SPECIFICATIONS FOR MERCEDES BENZ 2637 | 124 |
| | APPENDIX I – FLOWCHARTS FOR MODIFIED SIMTRANS MODEL | 127 |

LIST OF TABLES

| | | |
|-----------|--|-----|
| Table 3.1 | Equivalent percentage rolling resistance coefficients for radial and bias ply tyres on various travel surfaces (after Webster, 1995) | 35 |
| Table 4.1 | Observed vehicle trip details | 56 |
| Table 5.1 | Summary data for the simulated vehicle | 85 |
| Table D.1 | Observed and measured data from fuel flow tests on dynamometer | 109 |
| Table E.1 | Measured engine speed and pulses from the tachograph | 110 |
| Table H.1 | Mercedes Benz 2637 V Series specifications | 124 |

LIST OF FIGURES

| | | |
|------------|--|----|
| Figure 2.1 | Typical structure of an EMS (after Yotong <i>et al.</i> , 1998) | 9 |
| Figure 2.2 | Real time running mode (after Yotong <i>et al.</i> , 1998) | 10 |
| Figure 2.3 | Schematic of an optical exhaust particulate sensor (after Challen and Stobart, 1998) | 20 |
| Figure 3.1 | The SimTrans user interface | 32 |
| Figure 4.1 | The Throttle Position Sensor installed in the test vehicle | 47 |
| Figure 4.2 | The case containing the core instruments of the VPMS used for the vehicle performance tests | 54 |
| Figure 4.3 | The Mercedes Benz 2637 haulage vehicle used in the vehicle performance tests | 55 |
| Figure 4.4 | Vertical profile of Route One from the Eston Mill to Zone Three | 57 |
| Figure 4.5 | Vertical profile of Route Two from the Eston Mill to Zone Three | 58 |
| Figure 5.1 | Comparison of observed and simulated vehicle speed and the altitude on Hill Two via Route One | 60 |
| Figure 5.2 | Comparison of observed and simulated engine speed and gear changes on Hill Two via Route One | 61 |
| Figure 5.3 | Comparison of observed and simulated vehicle speed and the altitude on Hill Three via Route One | 63 |
| Figure 5.4 | Comparison of observed and simulated engine speed and gear changes on Hill Three via Route One | 64 |
| Figure 5.5 | Comparison of observed and simulated vehicle speed and the altitude on Hill Five via Route One | 65 |
| Figure 5.6 | Comparison of observed and simulated engine speed and gear changes on Hill Five via Route One | 67 |
| Figure 5.7 | Comparison of observed and simulated vehicle speed and the altitude on Hill Two via Route One using the modified model | 76 |
| Figure 5.8 | Comparison of observed and simulated engine speed and gear changes on Hill Two via Route One using the modified model | 78 |
| Figure 5.9 | Comparison of observed and simulated vehicle speed and altitude on Hill Three via Route One using the modified model | 79 |

| | | |
|--------------|---|-----|
| Figure 5.10 | Comparison of observed and simulated engine speed and gear changes on Hill Three via Route One using the modified model | 80 |
| Figure 5.11 | Comparison of observed and simulated vehicle speed and altitude on Hill Five via Route One using the modified model | 81 |
| Figure 5.12 | Comparison of observed and simulated engine speed and gear changes on Hill Five via Route One using the modified model | 83 |
| Figure B.1 | Flowchart illustrating the algorithm for the AirResistance function | 95 |
| Figure B.2 | Flowchart illustrating the algorithm for the GradeResistance function | 95 |
| Figure B.3 | Flowchart illustrating the algorithm for the RollingResistance function | 96 |
| Figure B.4 | Flowchart illustrating the algorithm for the GetMaxEngTorque function | 97 |
| Figure B.5 | Flowchart illustrating the algorithm for the GetTorqueLosses function | 98 |
| Figure B.6 | Flowchart illustrating the algorithm for the GetBrakeTorque function | 98 |
| Figure B.7.1 | Flowchart illustrating the algorithm for the SelectGear function | 99 |
| Figure B.7.2 | Flowchart illustrating the algorithm for the SelectGear function | 100 |
| Figure B.8 | Flowchart illustrating the algorithm for the GetFuel function | 101 |
| Figure B.9.1 | Flowchart illustrating the algorithm for the Main procedure | 102 |
| Figure B.9.2 | Flowchart illustrating the algorithm for the Main procedure | 103 |
| Figure B.9.3 | Flowchart illustrating the algorithm for the Main procedure | 104 |
| Figure B.9.4 | Flowchart illustrating the algorithm for the Main procedure | 105 |
| Figure B.9.5 | Flowchart illustrating the algorithm for the Main procedure | 106 |
| Figure C.1 | Flow rate distribution plot for flow meter 83014 | 107 |
| Figure C.2 | Calibration plot for flow meter 83014 | 107 |
| Figure C.3 | Flow rate distribution plot for flow meter 83015 | 108 |
| Figure C.4 | Calibration plot for flow meter 83015 | 108 |
| Figure D.1 | Error magnitude comparing observed and measured data | 109 |
| Figure E.1 | Engine speed calibration graph | 110 |
| Figure F.1 | Calibration plot of tachograph vehicle speed output | 111 |
| Figure G.1.1 | Flowchart illustrating the algorithm for the Main procedure of the CR10X program | 112 |
| Figure G.1.2 | Flowchart illustrating the algorithm for the Main procedure of the CR10X program | 113 |
| Figure G.2 | Flowchart illustrating the algorithm for Subroutine 1 of the CR10X program | 114 |

| | | |
|--------------|--|-----|
| Figure G.3 | Flowchart illustrating the algorithm for Subroutine 2 of the CR10X program | 115 |
| Figure G.4 | Flowchart illustrating the algorithm for Subroutine 3 of the CR10X program | 116 |
| Figure G.5 | CR10X program printout | 117 |
| Figure H.1 | Torque versus engine speed for ADE 443N engine | 125 |
| Figure H.2 | Power versus engine speed for ADE 443N engine | 125 |
| Figure H.3 | Plot of the dataset of observed engine speed versus data point number of the engine speed values, as they appear in the observed dataset | 126 |
| Figure I.1.1 | Flowchart illustrating the algorithm for the modified Main procedure | 127 |
| Figure I.1.2 | Flowchart illustrating the algorithm for the modified Main procedure | 128 |
| Figure I.1.3 | Flowchart illustrating the algorithm for the modified Main procedure | 129 |
| Figure I.1.4 | Flowchart illustrating the algorithm for the modified Main procedure | 130 |
| Figure I.1.5 | Flowchart illustrating the algorithm for the modified Main procedure | 131 |
| Figure I.1.6 | Flowchart illustrating the algorithm for the modified Main procedure | 132 |
| Figure I.2 | Flowchart illustrating the algorithm for the AccelerationCalc procedure | 133 |
| Figure I.3.1 | Flowchart illustrating the algorithm for the modified SelectGear function | 134 |
| Figure I.3.2 | Flowchart illustrating the algorithm for the modified SelectGear function | 135 |
| Figure I.3.3 | Flowchart illustrating the algorithm for the modified SelectGear function | 136 |

LIST OF ABBREVIATIONS

| | | |
|-----------------|---|--|
| ABS | – | Anti-lock braking system |
| ACC | – | Adaptive cruise control |
| ATS | – | Air temperature sensor |
| ASIC | – | Application specific integrated circuit |
| CAN | – | Controller Area Network |
| CAS | – | Crank angle sensor |
| CPU | – | Central processing unit |
| CRC | – | Cyclic redundancy check |
| CRS | – | Common rail system |
| CSMA/CD | – | Carrier Sense Multiple Access with Collision Detection |
| CTS | – | Coolant temperature sensor |
| DI | – | Direct injection |
| EGR | – | Exhaust gas re-circulation |
| EMS | – | Engine management system |
| FI | – | Fuel injection |
| GPS | – | Global Positioning Systems |
| HITCAN | – | Hitachi intelligent time triggered CAN |
| ISO | – | International Organisation for Standardisation |
| NO _x | – | Nitrogen Oxide |
| OBD | – | On-board diagnostics |
| PCM | – | Powertrain control module |
| ROM | – | Read only memory |
| TDC | – | Top dead centre |
| TPS | – | Throttle potentiometer sensor |
| TS | – | Throttle switch |
| VPMS | – | Vehicle Performance Monitoring System |
| VR | – | Variable reluctance |
| UKZN | – | University of KwaZulu-Natal |

1. INTRODUCTION

Transport costs can form a significant portion of the total production costs in the agricultural sector. In the current economic climate in Southern Africa with inflation, fuel prices and labour costs all increasing on an annual basis, transport managers are faced with a difficult task of ensuring that each transport operation is running at peak efficiency while at the least cost. Costs of a transport enterprise are comprised of fixed costs, such as the capital cost of a vehicle, insurance etc.; and variable costs such as fuel, tyres and maintenance. The choice of route for a transport operation can also impact significantly on the overall cost (Lyne *et al.*, 2005). At the outset, the transport manager is required to choose a suitable vehicle and best route to ensure maximum efficiency of the operation. This is a complex process as vehicles should not be selected based on lowest capital cost, but with several variables such as optimum route, fuel consumption and time requirements, all taken into account concurrently.

Several computer based models have been developed for the purpose of assisting transport managers in determining an optimal haulage system solution and thus aid the decision making process. SimTrans is one such model and was developed by the University of KwaZulu-Natal (UKZN). SimTrans is a mechanistically based model which can simulate any diesel vehicle travelling on any given route. SimTrans models the torque and power produced by the engine, vehicle speed, engine speed, gear changes and fuel consumption. The model provides the user with a comprehensive set of simulated vehicle performance results and a summary of trip details. The vehicle input parameters and route details can be modified to allow the user to simulate different vehicle configurations on alternate routes. Performing a separate costing exercise using the simulation results, the manager can make an informed decision as to the optimal transport solution in a particular situation.

Although other computer based transport decision support systems exist, these programs focus on driver training and vehicle dynamics. The other model which simulates vehicle performance is TransSolve. However, TransSolve does not provide the user with a comprehensive output of simulation results as SimTrans does. Having an output of simulation results enables the user to compare simulated performance with measured performance. TransSolve is also known to produce errors in the simulations and there is no

means of identifying where the programme is producing these errors as TransSolve is a “black box model” (Lyne, 2004). SimTrans on the other hand has accessible code which allows for modifications and useful additions to the programme. Thus it was decided to use SimTrans as the model to study for this research project.

It has been discovered that under certain conditions, particularly due to the gear selection, there is variance between actual and the simulated vehicle performance in SimTrans (Lyne, 2002). Thus the primary objective of this research was to refine, calibrate and validate the current SimTrans model.

The method employed in order to achieve the objective of this research was fourfold:

- To develop a data logging system and install it on a haulage vehicle to measure and record some operating parameters during operation;
- To simulate the performance of this same haulage vehicle operating under identical operating conditions;
- Identify the specific circumstances or reasons why the simulation results deviate from the observed data;
- Modify SimTrans code to correct these snags.

To gain a more in-depth insight into electronic engine management systems used with haulage vehicles and how they affect a vehicle’s performance, this document presents an overview study of diesel engine management systems for heavy-duty commercial vehicles. It was anticipated that performance data could be gathered directly from onboard vehicle electronics to form an observed data set. The powertrain control module, electronic fuel injection system and other features of engine management systems were considered, along with some types of sensors and actuators. Some specific sensors were examined in detail. A review of the Controller Area Network protocol for networking the vehicle electronic systems was also included.

The Mercedes Benz 2637 truck used for the performance tests was not fitted with an electronic management system suitable for direct data collection and thus a data logging system was developed. The data logging system consisted of a Campbell Scientific CR10X Data Logger which controlled all instruments on the system and stored data for download.

Instruments connected to the system included a GPS receiver, two fuel flow meters and a throttle position sensor. This data logging system generated a data set comprising of fuel consumption, engine and vehicle speed, vehicle position and throttle position.

The Mercedes Benz 2637 obtained for the performance tests was driven on two alternative routes between the Eston Sugar Mill and a timber compartment near Mid-Ilovo. The vehicle was driven on both routes in both directions unloaded and then loaded with timber. The vehicle with identical route and loading conditions was then simulated in SimTrans. By comparing the observed and simulated data sets, errors particularly with gear changing routines were identified. The SimTrans model was modified such that these errors were corrected and a new gear skipping routine was added. The model's performance after modifications were added was then validated through a comparison of the new simulated results and the observed data.

2. ENGINE MANAGEMENT SYSTEMS

The most important challenge facing vehicle manufacturers today is to offer vehicles that achieve high fuel efficiency, enhanced on and off road performance and lower emissions. Growing awareness that emissions produced by both petrol and diesel engines are a threat to the environment and contribute to global warming has resulted in a call for the reduction of greenhouse gases and other harmful products emitted from these engines. In order to meet these demands, vehicle manufacturers are now installing microprocessor controlled engine management systems in all types of vehicles. An engine management system (EMS) is an electronic control system whereby the injected fuel timing and quantity, idle speed, and other vehicle functions are controlled by a powertrain control module (PCM) (White, 1997). EMS's are used to refine the performance and efficiency of vehicles by improving parameters such as power and drivability, whilst achieving significant reductions in emissions.

Using electronic components in vehicles is not a recent innovation and electronics have been finding their way into vehicles for over 70 years (Jordan, 1987a). In the 1930's and 1950's attempts were made to use automotive electronic control systems (White, 1997). Many features of modern automobiles were investigated and then discarded by the early pioneers of automotive research, years before the technology became available to make their application possible. The high cost of components, harsh environment, and the primitive technology available proved insurmountable for many designers of electronic fuel injection systems up to the late 1950's (White, 1997).

Only in recent years with the increased ruggedness of electronic components and wiring harnesses, the improvement and increased reliability of semi-conductors and the introduction of integrated circuits and transistors, have microprocessor-controlled engine management systems started to make a more critical impact on engine control (Jordan, 1987a). The issue of pollution control and the attraction of increased engine power have driven the requirement for engine management systems to be used in most vehicles produced today. The first commercially produced vehicle with an engine management system was the 1978 BMW 732i – in the form of a Bosch system called Motronic (White, 1997). The PCM contained the circuits to control the ignition function as well as the fuel injection (FI) system (White, 1997).

EMS's are able to precisely control fuel metering and adjust it in response to changing operating conditions and to continually maintain the optimum air-fuel ratio (Probst, 1991). Additionally, EMS's may be used to control and communicate with other vehicle systems such as the air conditioning system, the antilock braking system, traction control and transmission-shift control system via the high speed Controller Area Network (CAN) data bus (Bauer, 2000, p551). One advantage of using an EMS is that it has a high degree of self-diagnostic capability with predictive fault detection; ensuring higher safety standards as well as aiding technicians locate engine problems faster and more efficiently. EMS's now come with built in redundancy measures, termed limited operation strategy, to ensure the entire system is not crippled by a minor failure such as a sensor malfunction.

This chapter presents an overview study of diesel engine management systems for heavy-duty commercial vehicles. Firstly the merits of electronic control systems are considered in order to gain some perspective on the relevance and benefits of EMS's for vehicles. Thereafter some specific components of EMS's are studied in some detail, i.e. the powertrain control module, electronic fuel injection systems and some relevant sensors and actuators. Some features of EMS's are presented to demonstrate the application of EMS's and electronics to vehicles. Finally a review of the CAN protocol for networking the vehicle electronic systems was incorporated as data transmission is a vital component of any electronic management system.

2.1 Merits of Electronic Control Systems

EMS's can provide a number of advantages over mechanical systems and in addition, several enhanced features can also be incorporated into vehicles using EMS's for improved safety, comfort and drivability (Amey, 1995). Drivability implying, the ability of the vehicle to provide smooth, trouble free performance under virtually any condition while delivering power, fuel economy, and controlled emissions (Probst, 1991). This section discusses some of the merits and drawbacks of using an EMS compared with mechanical systems. A few of the enhanced features of EMS's will also be discussed here.

All EMS's have self-diagnostic capability (Jordan, 1987b). If the PCM receives an out of range signal during vehicle operation or the "self-test" mode, a corresponding code will be generated (Anonymous, 1990). Service codes, for later inspection by a service technician,

may also be generated to indicate that servicing operations are required (Anonymous, 1990). Current engine operating parameters, such as fuel consumption and engine speed may be displayed on the dashboard to help the driver plan trips more economically and drive to meet fleet goals (Viljoen, 1996). The PCM has non-volatile memory to retain the self-diagnostic data in memory once the ignition is switched off (White, 1997).

Recently engine manufacturers have been designing EMS's so a computer with diagnostic software can connect to the PCM through the standard communication port (Ishiguro and Masui, 1999 and Viljoen, 1996). Connecting a computer to the PCM can be for diagnostic, control, optimisation or fleet management purposes (Ishiguro and Masui, 1999). Information concerning the vehicle's operation during a trip, such as fuel consumption and average time spent in one gear, can be stored by the PCM to be later captured by a computer (Viljoen, 1996).

EMS's incorporate a limited operation strategy (limp home facility), which allows the vehicle to be driven if a malfunction occurs in the EMS (White, 1997). If for example the crank angle sensor fails, the PCM can switch to use the camshaft sensor as a measure of the engine speed (Krebs, 1993). The PCM can also adopt factory pre-programmed data in place of a sensor that has failed. Vehicle performance will be negatively affected in limp home mode, but the vehicle can be driven for a limited time until the malfunction is attended to.

Some EMS's employ an "adaptive strategy" in which the PCM continually adapts the functioning of the system to changing engine characteristics, due to engine wear etc. (Anonymous, 1990, Jordan, 1987b and White, 1997). This keeps engine operation to within set tolerances at all times. The PCM applies corrected values to the basic engine maps to allow faster engine response to changes in the engine (White, 1997).

Other advantages of electronic systems over mechanical systems can be illustrated by considering the electronic governor. An electronic governor can monitor a number of engine and vehicle parameters and combine these to calculate the required injected fuel quantity, which is not possible with a mechanical unit (Bauer, 2000, p541). Electronics can correct for air and fuel temperatures and allow complete freedom in determining the full-load response of the engine (Bauer, 2000, p542). The PCM can precisely coordinate the maximum injected

fuel quantity with the boost pressure to keep the engine below the smoke limit (Bauer, 2000, p542).

The PCM is able to regulate and control the engine speed for purposes of maximum speed, consistent low idle speed, and auxiliary power take-offs (e.g. adjustment for air conditioner operation) (Bauer, 2000, p542). Other advantages of EMS's include a cruise-control facility, enhanced comfort (smooth-running control) and the option for intervention in the traction control system and automatic transmission (Bauer, 2000, p542).

Electronic fuel injection systems allow improved control of injected fuel quantity and more precise control of timing, which positively affects fuel consumption, engine power, and emissions. Fuel can be cut during deceleration or when the speed control lever is released to maintain smooth running of the engine and as a fuel economy measure (White, 1997). New technology pumps no longer need calibration or injectors to be serviced (Viljoen, 1996). Solenoid valve control of fuel injection pumps allows a rapid response of the pump, improved high pressure sealing and therefore offers greater efficiency (Bauer, 2000, p545).

EMS's can be used to shutdown the engine after a preset idling time to conserve fuel and reduce engine wear (Viljoen, 1996). The same feature can also be used to automatically start or stop the engine for air conditioner purposes, although the driver can override this system for safety and other reasons (Viljoen, 1996).

Cruise control systems are another feature of EMS's which allow the driver to set constant speeds. In some EMS's, the PCM may even select a higher power curve when cruise control is engaged (Viljoen, 1996). Cruise control systems automatically activate engine brakes and the engine fan to retard engine speed (Viljoen, 1996). Adaptive cruise control systems function similarly to conventional cruise control systems, but in addition can detect the distance of an object ahead using a radar sensor system and then give control to the powertrain and anti-lock braking system to control the vehicle and avoid the collision (Uno *et al.*, 2001). Fenton (1996) discusses vehicle stability control systems that can prevent a vehicle becoming unstable while cornering even if the driver takes no corrective action.

“Drive-by-wire” control systems can be used to control steering, the speed control lever, and braking without mechanical links between the driver's controls and the control actuators

(Ward and Fields, 2000). These systems can reduce costs, weight, complexity, labour and maintenance and also provide a number of enhanced features for vehicles. Four wheel steering and variable rate steering are two such features (Wright, 1990). Collision avoidance systems are another useful spin-off of drive-by-wire systems as well as automatically adjusting suspension to changing road conditions.

There are however, several downsides to using electronic controllers. Vehicle maintenance is more complex in engines utilising EMS's, because a specialist or the original equipment manufacturer is required to service the electronic components (Viljoen, 1996). Several engine manufacturers have reservations on the practicality of sophisticated electronic controllers in a country such as South Africa (Viljoen, 1996). Electronics are also quite costly and thus using EMS can increase vehicle prices and maintenance costs.

2.2 Powertrain Control Modules

When electronics were first introduced into vehicles, there were a number of individual controllers used to manage the various components of the engine. However, in time these minimal feedback applications were integrated into one computer controlled system, called an EMS, with the powertrain control module (PCM) (also called the electronic control module or electronic control unit) as the central controller of the system. The PCM synchronises the components and subsystems in the vehicle, including injection timing and quantity through the use of pre-programmed software and the on-board vehicle network, so all systems function in harmony with one another and as a single entity (Chowanietz, 1995).

The main function of the PCM is to accurately control the fuel injection process for the operating state of the engine at any given time by defining the exact start of injection and injected fuel quantity (Bauer, 2000, p551). The PCM analyses all relevant status parameters in the system and then determines the required action in real time. In addition to fuel injection, the PCM can control functions for improved driving smoothness, such as idle-speed governors and adaptive cylinder matching as well as turbocharger pressures (Bauer, 2000, p551). The general structure of a diesel EMS is given below in Figure 2.1. The systems shown in the figure will be discussed in this document.

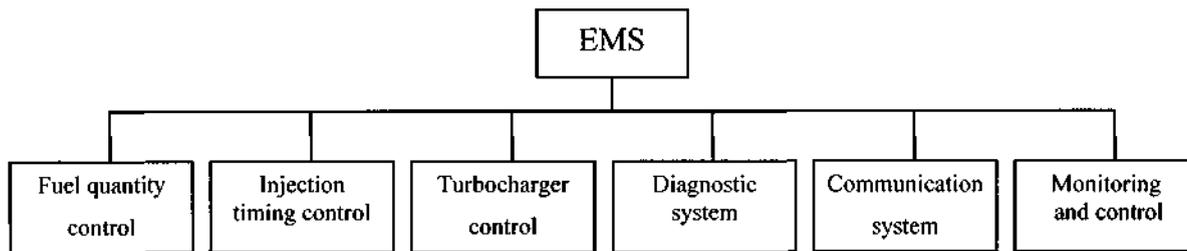


Figure 2.1 Typical structure of an EMS (after Yotong *et al.*, 1998)

Government authorities, particularly in America and Europe, have set strict standards in regard to fault diagnosis in EMS's, particularly in regard to the emissions control system (Bauer, 2000, p551 and Challen and Stobart, 1998). To this end, the PCM continually monitors the sensors for errors, and if a fault or deviation is detected in the system, e.g. coolant temperature or oil pressure out of the normal range, the PCM will attempt to correct or compensate for the fault (Bauer, 2000, p551). A warning light will be displayed on the dashboard to provide diagnostic information to the driver. If the problem remains unresolved for a set time period, the PCM will shutdown the system to prevent further damage (Viljoen, 1996). As another safety feature and to limit the maximum vehicle speed, the PCM can stop injector operation if the engine speed exceeds a pre-designated r/min (White, 1997 and Wright, 1990).

The EMS hardware requires software to operate and manage the various linked systems installed throughout the vehicle. The design methodology for EMS control software varies between system designers, but according to Yotong *et al.* (1998), the key features of a real-time EMS software design are:

- Real time multi-tasking control technology (as shown in Figure 2.2);
- Open software structure for additional system modules;
- Real time control software with a central interrupt program;
- Multi-module interface (running control function can be added);
- Controls fuel injection quantity and timing as well as turbocharger operation;
- Allows easy upgrading of software.

Originally designers employed map-based control systems to run the PCM, however, more recently real-time model-based control systems are being developed. These two approaches shall be investigated in further detail in the following two sections.

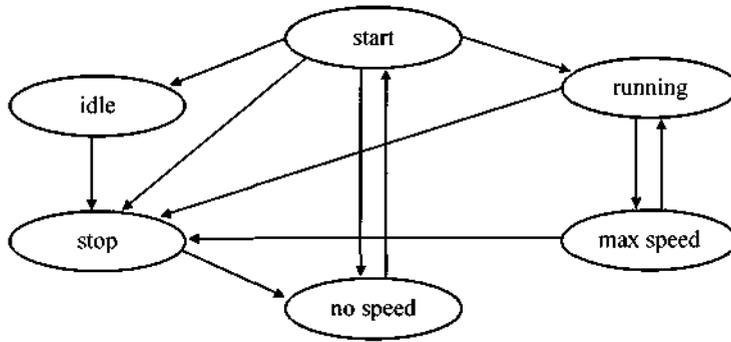


Figure 2.2 Real time running mode (after Yotong *et al.*, 1998)

2.2.1 Map-Based Control Systems

In map-based control systems, data on injection timing and duration are digitally mapped and stored in the read-only memory of the PCM in a series of two- or three- dimensional maps (Jordan, 1987a and White, 1997). These maps or lookup tables allow the PCM to choose the correct injection timing and duration according to the current engine load and speed. The maps are fairly definitive in terms of coverage of the various engine parameters. However, to accommodate the files in the read-only memory, the parameters are only mapped at specific intervals e.g. speed stored in 5 r/min steps, leaving the PCM to interpolate in between these values. The advantages of using a map-based approach are that control system designers are able to enjoy the freedom of a three-dimensionally mapped surface to predict fuel requirements and are restricted only by the amount of available read only memory and the resolution of the sensors (Jordan, 1987a).

The development of map-based control systems is an iterative empirical process requiring corrections for such factors as temperature and transient operation (Challen and Stobart, 1998). This development methodology relies on extensive testing of the system and there are restrictions on adjustment as the maps have an interdependent nature and a change in one map would affect other maps. In effect this results in increased development times due to the extended calibration periods. The adoption of real-time model-based controllers, on the other hand, can considerably ease these development challenges (Challen and Stobart, 1998).

2.2.2 Real-Time Model-Based Control Systems

With the development of electronic control systems, there has been a rapid expansion in the number of parameters involved such as injected fuel quantity and timing, exhaust gas recirculation (EGR) rate and fuel rail pressure (Challen and Stobart, 1998). This has resulted in systems that are increasingly complex and difficult to manage, even if the variables are independent (Challen and Stobart, 1998 and Yotong *et al.*, 1998). The parameters often have an inter-related nature and influence one another causing difficulty in achieving optimum performance.

In the model-based system the controller concurrently gathers information from the sensors and runs a mathematical model to deduce the system behaviour (Challen and Stobart, 1998). Although there is no absolute prediction involved, the controller pre-processes sensor information and uses this to anticipate the likely future, which achieves an improved control performance. This system comes at the cost of some extra computational load being placed on the PCM (Challen and Stobart, 1998). According to Challen and Stobart (1998) high-resolution engine models capable of tracking engine operation in real time are being developed and have already been demonstrated.

The Nitrogen Oxide (NO_x) regulation system for example could incorporate a high speed model control system. The controller uses the expected amounts of NO_x production and the calculated fuel burn rate to determine the injection rate in a feedback loop (Challen and Stobart, 1998). On-board-diagnostics, condition monitoring and knock detection model-based systems are expected further developments in this field (Challen and Stobart, 1998).

Today microprocessors are faster and more affordable than previous generations, but fundamental barriers still need to be addressed before model-based systems can be implemented. The timescales of different events occurring in an engine are wide, for instance comparing the timescales of cylinder events (in the millisecond range) and monitoring on-board diagnostics (in the seconds range). This has led to the derivation of “virtual sensors”, where information on the operation of a part or aspect of the engine or powertrain can be deduced from other available data (Challen and Stobart, 1998). Another approach includes the use of fuzzy logic models to assess engine demands and maintenance needs to assist in exhaust emission control (Challen and Stobart, 1998).

Challen and Stobart (1998) state that the EGR control system provides an example of the use of “virtual sensors”. Knowledge of engine intake and exhaust system dynamics has been used to develop an exhaust model (Challen and Stobart, 1998). Physical sensors for pressure and differential pressure in the intake and exhaust systems are negated as the system relies solely on a measurement of intake airflow to predict emissions (Challen and Stobart, 1998). On-board-diagnostics systems use similar approaches (Challen and Stobart, 1998).

2.3 Electronic Fuel Injection Systems

The air-fuel mixture supplied to the engine can significantly influence fuel consumption, emissions and combustion noise; so achieving the correct mixture is vital (Benninger and Plapp, 1991 and Riesenber *et al.*, 1999, p18). Maximising power from the engine results in high fuel consumption, while minimising fuel consumption means sacrificing power and driveability. Choosing to either maximise or minimise fuel consumption will result in increased emissions, as the minimum emissions production point lies in between these two extremes. The fuel delivery system must therefore be able to find the best compromise between fuel economy, vehicle performance and lower emissions and meet these conflicting demands in the most satisfactory way. According to Probst (1991), this means slight sacrifices of power and fuel economy in exchange for optimum emissions control.

To meet the compromise between low fuel consumption and emission limits compliance, the start of injection needs to be precisely timed with an accuracy of $\pm 1^\circ$ crankshaft rotation (Bauer, 2000, p534). The start of injection controls the start of combustion and is adjusted to compensate for the pressure-wave propagation times in the pressure lines (Bauer, 2000, p534). The PCM processes injection timing and duration jointly so that the optimum ignition timing is determined for every operating condition.

The two types of injection systems available in the diesel engine are prechamber combustion and direct-injection. The fuel consumption in direct-injection engines can be up to 20% less than prechamber engines and for this reason direct-injection engines are used in situations where economy, rather than driver comfort, is the deciding issue, such as in the transport and agricultural industry (Riesenber *et al.*, 1999, p6). Hence this review will only consider direct-injection diesel engines.

2.3.1 Operation

The FI system consists of a low-pressure stage and a high-pressure stage. A mechanical or electric pump, fuel filters, fuel tank and fuel lines form the low-pressure stage. A high-pressure pump generates the fuel pressure required for injection in the high-pressure stage. The pressure in the FI line is regulated by the PCM using a control valve, which is generally solenoid controlled (Bauer, 2000, p534). However, using piezo-valves is a possible future development (Bauer, 2000, p535).

The quantity of injected fuel is defined by the injector opening period and pressure in the fuel rail. Injector opening period is governed by a signal pulse width, while the frequency of the signal determines the timing (Anonymous, 1990). Input signals from the sensors are used to determine the signal properties, such as engine speed and crank angle, temperatures of the engine coolant, fuel and intake air, starter operation, boost and atmospheric pressure, turbocharger speed, engine knock and the speed control lever position (Hlousek, 1998). The PCM synchronises injection timing with the piston position using signals from the engine position sensor (Bauer, 2000, p551 and Riesenber *et al.*, 1999, p19). Fuel temperature is also measured so the PCM can compensate for fluctuations in the fuel density (Viljoen, 1996).

2.3.2 Fuel Pumps

In-line FI pumps can be electronically controlled through actuators adjusting the pump rack position to regulate fuel quantity (Riesenber *et al.*, 1999, p48 and Yotong *et al.*, 1998). In control-sleeve in-line injection pumps the start of injection is controlled electronically through an actuator mechanism turning the control shaft (Bauer, 2000, p542). The start of injection is monitored at the injection nozzle by a needle-motion sensor (Bauer, 2000, p542).

Distributor FI pumps have electronic governing with an integrated timing device (Riesenber *et al.*, 1999, p48). One type of a distributor pump has a solenoid rotary actuator to adjust the position of the control collar (Bauer, 2000, p545). The timing device consists of a needle-motion sensor and a clocked solenoid valve connected to the working chamber of the plunger in the timing device (Bauer, 2000, p545). Solenoid valve controlled distributor pumps meter fuel by a high-pressure solenoid valve, which directly closes off the pump's element chamber

(Bauer, 2000, p545). Other types of solenoid controlled distributor pumps include axial-piston and radial-piston distributor pumps.

2.3.3 Time-Controlled Single-Cylinder Pump Systems

One method of FI control is the use of time-controlled single-cylinder pump systems, which are modular systems utilising one high pressure pump per cylinder. This method can be further split into unit injector systems and unit pump systems.

2.3.3.1 Unit Injector System

The electronically controlled unit injector consists of a pump and nozzle integrated into a single unit with a solenoid controller valve (Bauer, 2000, p548 and Riesenberg *et al.*, 1999, p49). Each cylinder is given its own unit installed directly in the cylinder head and is operated by a rocker arm driven by an injection cam on the engine camshaft (Bauer, 2000, p548 and Riesenberg *et al.*, 1999, p49). The high speed acting solenoid valve controls the start and duration of injected fuel and allows a precisely defined rate-of-discharge curve (Hlousek, 1998).

This system operates at injection pressures up to 200 MPa for improved fuel atomisation and this, combined with electronic map-based control, allows reduced emission and fuel consumption levels (Hlousek, 1998). Driving smoothness is also improved in this system as any irregularities in the drivetrain may be accommodated for by adaptive cylinder matching (Bauer, 2000, p548). This system also permits the use of electrically controlled pilot injection (Hlousek, 1998).

Electrically controlled pilot injection operates by the PCM triggering the solenoid valve twice in rapid succession (Bauer, 2000, p546). Pilot injection spreads out the combustion event over time thus reducing the maximum combustion temperature (Johnson and Diamond, 2001). Pilot injection can thus reduce NO_x production and combustion noise while allowing improved cold-starting characteristics as well as a high degree of precision and dynamic response of injection (Bauer, 2000, p546 and Johnson and Diamond, 2001).

2.3.3.2 Unit Pump System

The principle of the unit pump system is the same as the unit injector system, as it is a modular, high pressure, time-controlled, single-cylinder pump system (Bauer, 2000, p549 and Riesenberg *et al.*, 1999, p49). However, the unit pump system features a short delivery line between the pump and the injector (Hlousek, 1998). This system enables the integration of the pump within the engine block to be driven by an injection cam on the engine's camshaft via a tappet roller (Hlousek, 1998). The unit pump system negates using the rocker arm as in the unit injection system to simplify the design. Both systems allow limited operation strategy as the solenoid valve is designed to fail in the shutoff position allowing individual cylinders to be isolated.

2.3.4 Common Rail System

In the common rail system the pressure generation and actual injection processes are decoupled (Bauer, 2000, p551, Ganser, 1998 and Riesenberg *et al.*, 1999, p49). The pressure in the fuel rail is generated independently of engine speed and injected fuel quantity and is stored in the rail as an accumulated volume (Ganser, 1998). A high-pressure in-line pump with a fuel quantity control system is used to generate the pressure. The pressure can be generated at low speeds and torques if required, which substantially reduces the power demands on the engine (Bauer, 2000, p551).

The injectors are electro-hydraulically controlled with the injector's needle valve opened by a solenoid (Ganser, 1998). The start and quantity of injected fuel is calculated by the PCM, which at the precise instant transmits an activation signal to the injector solenoid to initiate fuel delivery (Ganser, 1998). The fast acting solenoid can be triggered several times in succession for multiple injection (Bauer, 2000, p551). Hydraulic pressure is used to augment injector-needle closing and ensures rapid termination of the injection process (Bauer, 2000, p551). The PCM, sensors and most of the other system functions in the common rail system behave in the same way as the other time-controlled single-pump systems. To further enhance the system and reduce noise and emissions, pilot injection may be used (Ganser, 1998).

Ganser (1998), discusses a common rail system with an injection rate control consisting of a double-stop solenoid valve. When the solenoid valve moves only through the first stop, the needle valve can be opened slowly. By moving the solenoid valve through the first and second stops the needle valve can be opened faster. The two injection rates may be selected by the PCM switching between different pulse durations (Ganser, 1998). This offers a fully controllable initial shape of the injection rate curve (Ganser, 1998).

In common rail systems, the exact quantity of fuel injected must be measured, depending on the absolute rail pressure and pressure oscillations in the rail (Torkzadeh *et al.*, 2002). The delay between the trigger of the injection signal and the injector opening can result in non-smooth idling in common rail systems (Torkzadeh *et al.*, 2002). To solve these problems, the PCM needs to be supplied with improved information on the pressure fluctuations in the fuel rail and amount of fuel injected (Torkzadeh *et al.*, 2002). Torkzadeh *et al.* (2002) suggests that a magneto-elastic effect sensor installed ahead of the injector can be used to provide these increased information demands. This non-invasive pressure sensor can be used to measure the pressure progression inside an injection tube without any contact to the medium inside the tube (Torkzadeh *et al.*, 2002). This is useful in high pressure systems as the problem of measuring density is negated. This sensor has no mechanical elements and is relatively low cost (Torkzadeh *et al.*, 2002).

2.4 Features of Engine Management Systems

Three specific applications of EMS's in vehicles are presented in this section, i.e. engine management utilised in engine starting, the electronic governor and emission control systems. This demonstrates the purpose and benefit of electronics and EMS's to vehicle performance and the environment through reduced emissions and lower fuel consumption.

2.4.1 Engine Starting

There are three techniques vehicle manufacturers can use to assist starting: firstly, a fuel adaptation processes can be used where the fuel is heated or an additive used in the fuel; secondly, start-assist systems such as a glow plug can be used; and thirdly, an adaptation of the injection system can also be used. Glow plug operation is controlled by the PCM which uses information on the engine state, i.e. coolant temperature, engine speed and load (Bauer,

2000, p570). The control unit integrated into the glow plug can signal any faults it detects to the PCM, which then can store or display this information (Bauer, 2000, p570). Starting is also assisted by the fuel injection system providing excess fuel at start-up to compensate for condensation and leakage losses (Riesenberg *et al.*, 1999, p13). As an additional measure, injection timing may be advanced to compensate for ignition lag and ensure correct ignition timing (Riesenberg *et al.*, 1999, p13).

2.4.2 Electronic Governor

An electronic governor may be used in place of flyweights in all fuel-injection pump systems (Bauer, 2000, p541). The PCM determines the fuel quantity to inject by comparing the position of the speed control lever to the engine speed and then applying a number of correction factors from the program maps (Bauer, 2000, p541).

2.4.3 Emission Control

Reduced exhaust emissions are a high priority in automotive design and much of the rapid spread of electronics in EMS's are driven by the need to reduce harmful emissions to standardised levels (White, 1997). In order to achieve an ultra low emission engine, Bradley (1994), states that three possible approaches exist: a more efficient combustion system and improved combustion chamber design to reduce emissions at the source; increasing the efficiency of the oxidation type catalytic converter and particle filter, especially during cold starting; and improved engine and FI control to maintain the correct air/fuel ratio and injection rate. Another solution is the use of an EGR system (Bradley, 1994 and Riesenberg *et al.*, 1999, p26).

In EGR systems, exhaust gases are mixed with the intake air at part-load operation (Bauer, 2000, p561). The exhaust gas increases the specific heat of the charge and results in a lower combustion temperature and thus decreased NO_x formation (Bauer, 2000, p561). However, too much EGR can increase emissions of hydrocarbons and soot due to an air deficiency, so the quantity of EGR must be limited (Bauer, 2000, p561 and Riesenberg *et al.*, 1999, p23).

2.5 Sensors

The operating environment for automotive sensors is demanding in terms of ruggedness, durability and suitability; and specialised sensors with customised characteristics are required. Consequently many designers have difficulty in adapting novel systems to the severe demands (Challen and Stobart, 1998). The design and manufacture of sensors needs to meet the parallel demands of affordability, reliability, mechanical and environmental robustness combined with flexible packaging and variable demand (Challen and Stobart, 1998). While aiming for functionality, designers must also consider other aspects such as the choice of passive or powered sensors and harness optimisation (Challen and Stobart, 1998).

Diesel engines utilise numerous sensors for a range of sensing functions from speed and position, through to pressure and temperature sensing. Reliable data describing engine operation is vital to developing a successful, high performance control system. This demand has led the array of sensors currently available, as well as the continuing development of new technologies, such as fibre-optics and high-temperature electronics.

2.5.1 Smart and Integrated Sensors

It is increasingly common to have some local processing power integrated into sensors, both for performance and reliability reasons (Challen and Stobart, 1998). These types of sensors are termed “smart sensors” and can increase the performance of the system due to their ability to amplify and digitise signals. Using smart sensors can add an element of confidence to the output and provide a further safeguard against corruption in the measurement chain (Challen and Stobart, 1998). This is because smart sensors can calibrate themselves and are capable of error detection. Smart sensors can be fitted with data bus communication capability by providing a processor, protocol controller and transceiver chip to the sensor module (Sparks *et al.*, 1999). This allows the sensors to identify themselves to the PCM and simplifies the design and installation. A sensor that can locally compensate for temperature is an example of a smart sensor.

Considerable intelligence can be built into the sensor unit. It can often be cost-effective to include a processor core in the application specific integrated circuit on the sensor (Challen and Stobart, 1998). This allows various functions to be implemented in a flexible and reliable

manner (Challen and Stobart, 1998). These devices have low power consumption making them suitable for operation with vehicle batteries without significantly affecting the battery's storage time (Challen and Stobart, 1998). Providing an application specific integrated circuit for local data processing has an added component cost, but this will only constitute a small fraction of the total package if a reasonable production quantity is required (Challen and Stobart, 1998). Costs can be minimised by spreading generation costs across a segment of the production volume. Currently there is a rapid decrease in generation and production costs of application specific integrated circuits (Challen and Stobart, 1998).

Several sensors may be locally clustered (termed multi-sensor modules) to reduce weight and the number of connectors, chips, and cables (Sparks *et al.*, 1999). For example a cluster used in the climate control system of a vehicle may consist of a surface temperature sensor, a humidity sensor, and an air quality sensor (Sparks *et al.*, 1999). Using multi-sensor modules can save on costs and improve reliability while eliminating multiple housings (Sparks *et al.*, 1999). Future trends in the field of multi-sensor modules include integrating several sensors on one silicon chip (Sparks *et al.*, 1999).

2.5.2 Chemical Sensors

There has been significant progress in the last decade in the development of chemical sensors (Challen and Stobart, 1998). Challen and Stobart (1998) describe a chemical-gas sensor, still in development, which can chemically detect the concentration of a chosen gas species and produce an output signal by optically detecting a colour change in the chemical. This sensor can be produced as a smart sensor having the potential for accurate measurements at a reasonable cost (Challen and Stobart, 1998). This chemical-gas sensor can be used in the emission control system to detect the concentrations of NO_x.

2.5.3 Optical Sensors

Optical sensors have potential for automotive application and have been used in various research activities for extracting information from the optical output of the combustion process (Challen and Stobart, 1998). One application of optics to vehicles is optical connections to integrated sensors. Challen and Stobart (1998) conclude that only when the information from these sensors become more valuable (possibly for on-board-diagnostics

applications and long-term performance assurance), the disadvantages (cost and physical limitations) of using this type of sensor will be balanced out.

Other optical sensor applications developed for the diesel engine include measuring the cylinder pressure using the surface vibration as an indicator of the combustion process (Challen and Stobart, 1998). Challen and Stobart (1998) state that this technique may even be an improvement on the approximation of cylinder pressure using surface accelerometers. This technique has been successfully demonstrated, and could be used in production engines within the next decade, if the costs of processing continue to fall (Challen and Stobart, 1998).

With new legislation, exhaust particulate sensing is increasingly vital. An optical device has been developed which can sense light reflections from small particles in a gas stream, see Figure 2.3 (Challen and Stobart, 1998). Even low particulate concentrations can be detected with this method (Challen and Stobart, 1998).

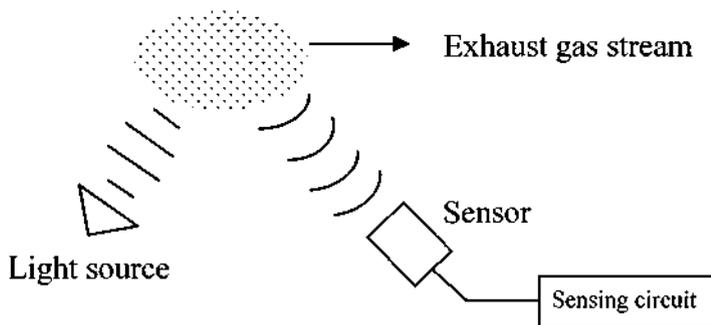


Figure 2.3 Schematic of an optical exhaust particulate sensor (after Challen and Stobart, 1998)

Challen and Stobart (1998) mention that significant information about the start of burn, burning temperatures and the end of the burning process can be obtained from detecting the broad range of wavelengths produced by combustion radiation. However, this technology is not presently commercially available, as handling the optic fibres, positioning and alignment of the sensors, critical to sensor performance, is proving difficult and costly (Challen and Stobart, 1998).

Pressure, temperature and strain sensing as well as communications between sensors and groups of sensors, are all possible areas for optical applications in vehicle systems (Challen

and Stobart, 1998). Challen and Stobart (1998) report that although optical sensors have been used for detection and counting tasks, their use is not widespread in diesel engines, despite their advantages.

2.5.4 Knock Sensor

Engine knock is a phenomenon that occurs primarily at medium to low load conditions and arises due to poor mixture formation (Bauer, 2000, p383). Knock is characterised by a pulsating metallic sound, usually producing a frequency in the 6 to 15 kHz band (Chowanietz, 1995). Knock can occur at different times in each individual cylinder, so the PCM has a knock control processor to identify the cylinder or cylinders that are knocking (Jordan, 1987a and White, 1997). When knocking is detected, the PCM retards the injection timing in the cylinder that is knocking by a defined step (White, 1997). After knocking ceases, timing is advanced until the knocking limit is reached (Krebs, 1993 and White, 1997).

A knock sensor consists of a piezo-ceramic measuring element that is mounted on the engine block and responds to engine noise (Jordan, 1987a and White, 1997). The signal generated in the sensor is proportional to the level of knock (White, 1997). The presence of engine knock is identified if the knocking intensity of any one combustion cycle exceeds a predefined threshold (Jordan, 1987a and White, 1997). The system uses frequency filters and measuring windows to detect the knocking event (Krebs, 1993).

2.5.5 Inductive Speed and Position Sensors

Inductive sensors, also called variable reluctance sensors, consist of a coil on a soft iron core attached to a permanent magnet (Jordan, 1987a, Pawlak, 1999 and White 1997). This sensor produces a signal every time a tooth on a gear or rotor passes the transducer and thus indicates rotational speed or position (Jordan, 1987a and White, 1997). Inductive sensors can be used as crank angle sensors to determine both speed and position of the crankshaft.

Pawlak (1999) discusses a distributed variable reluctance sensor, which consists of a radially orientated multi-pole magnet and a coil. A toothed exciter wheel runs on the inside of the sensor and is the reluctor. This design provides an advantage over conventional pick-up limb type inductive sensors as it utilises the variation in the magnetic field three-dimensionally to

provide improved reliability with less variation in the signal (Pawlak, 1999). The ring-shaped distributed variable reluctance sensor also allows for integration into a wheel bearing for speed sensing and has low sensitivity to radial tolerances (Pawlak, 1999).

Inductive sensors are used as rack-travel sensors to monitor the position of the control-rack and as cylinder identification sensors (White, 1997). The port-closing sensor is another use of the inductive sensor and it monitors the point at which pump delivery starts when the engine is running and can also be used as a check for the timing device (Bauer, 2000, p540). According to Jordan (1987a) and Pawlak (1999), inductive sensors are widely used because they are reliable, inexpensive and self-excited. The drawback of using inductive sensors is that they produce output voltages ranging from 500 mV to 80 V, depending on the rotor speed and require signal processing to discriminate the signal from the noise (Jordan, 1987a and White, 1997).

2.5.6 Temperature Sensors

The temperature sensors are used for air temperature and coolant temperature determination. Sensors can be mounted at various locations (for example on the inlet manifold and in the air filter) depending on the temperature to be sensed. Two air temperature sensors are sometimes used, one to measure ambient engine compartment air temperature, and one to measure intake air temperature. This may be important if a turbocharger is used (White, 1997). The coolant temperature sensor is immersed in engine coolant and is used by the PCM for a correction factor when calculating the injection timing and duration (White, 1997).

2.5.7 Speed Control Lever Position Sensors

Speed control lever position may be detected by a throttle switch and a throttle potentiometer sensor, in a combination or as separate components (White, 1997). The throttle potentiometer sensor informs the PCM of the position and rate movement of the speed control lever (White, 1997). The throttle switch has two contact points, an idle and a full throttle contact, and when the speed control lever is at an extremity, the corresponding contact is made. Hence the throttle switch senses three states – idle, throttle open, or full throttle (White, 1997). Hall Effect rotary-position sensors are now being used as speed control lever position sensors. The advantages to using Hall Effect sensors are that they are durable and as non-contacting

sensors have low wear; they also allow for the integration of electronics to become smart sensors. The speed control lever position is used by the PCM to determine the quantity and timing of the injected fuel. In vehicles with automatic transmissions, the speed control lever position is used to determine the gear shift timing.

2.5.8 Inertia Switch

Should a vehicle be involved in a collision, any fuel in the fuel lines could ignite. The inertia switch is thus provided as a safety device to electrically isolate the fuel pump in an event of a sharp deceleration – i.e. a collision (White, 1997). This then prevents the further flow of fuel into the fuel lines to minimise the chance of fuel ignition. Once triggered, the switch will remain active until reset (White, 1997).

2.6 Networking Vehicle Electronic Systems

Modern vehicles are equipped with a number of electronic controllers and sensors that need to exchange data in order to perform their various functions (Bauer, 2000, p884 and Uno *et al.*, 2001). According to Fenton (1996), a luxury car can contain over 2 km of cables and 1500 electrical connections. The use of a specialised vehicle data bus network, in a multiplexing process presents a solution and the Controller Area Network (CAN) protocol has emerged as an ISO standard (Fenton, 1996). The CAN communication protocol is a carrier sense multiple access with collision detection protocol, capable of transfer speeds up to 1 Mbit/s (Pazul, 1999). The robustness and reliability of the CAN system has resulted in a large following from the automotive and semiconductor industry.

There are four areas of application of the CAN bus in vehicles: real-time, multiplex, mobile-communications, and diagnostic applications. Systems used to control vehicle dynamics are real-time applications and require bus speeds of 125 kbit/s to 1 Mbit/s (Bauer, 2000, p884). Multiplex applications are associated with control and regulation of body-component and luxury / convenience systems and bus speeds of 10 kbit/s to 125 kbit/s are required (Bauer, 2000, p884). Mobile-communication applications connect communication devices such as the navigation and audio system with central controls at bus speeds below 125 kbit/s (Bauer, 2000, p884). The CAN bus can be used for diagnostic applications in which bus speeds of 500 kbit/s are used (Bauer, 2000, p884).

The CAN bus functions on a multi-master principle, in which a linear bus structure connects several control units that are each assigned equal priority ratings (Bauer, 2000, p884). This ensures a malfunction at one node does not affect the remaining devices. Data transmitted via the CAN bus contains no address information; instead the message is labelled by an identifier that is unique throughout the network and describes the content of the message. All nodes on the network receive the message and process only those messages whose identifiers are stored in their acceptance list (Bauer, 2000, p884). The identifier labels also contain priority assignments. When transmitting data, the most important message is sent first, as soon as the bus is unoccupied (Bauer, 2000, p884). To detect errors, the CAN protocol includes a cyclic redundancy check, bit stuffing check and frame checking (Bauer, 2000, p886). Monitoring features where the transmitter compares the transmitted and scanned bit are also used as an error check (Bauer, 2000, p886).

Fenton (1996) states that two bus systems should be used in a vehicle – one a high-speed ring for the powertrain, chassis, antilock braking system, suspension etc.; and the other a low-speed body control data bus for such systems as vehicle lights, seats, cabin environment etc. Most of the information transmitted on the low speed bus is event driven and can tolerate relatively long delays, while information on the high-speed bus is real-time control data that is either time or safety critical and usually of a periodic nature with higher bus loads (Fenton, 1996). Fenton (1996), also mentions that the master controller of the CAN bus system can be programmed in a high level language and interfaced to a high-speed data logger.

Uno *et al.* (2001) describes a technique recently developed to handle large amounts of data transmitted via the CAN bus. This technique aims at minimising central processing unit overhead to maximise performance of the central processing unit and CAN bus. This involves the use of a CAN controller module, called the Hitachi Intelligent Time Triggered CAN. This system was simulated in real time with an adaptive cruise control application. Compared with the Hitachi CAN system (an older version of the Hitachi Intelligent Time Triggered CAN), Hitachi Intelligent Time Triggered CAN is able to reduce central processing unit instructions by approximately 90% and allow the central processing unit to be 99.76% free to handle other tasks. (Uno *et al.*, 2001)

Zeltwanger (1999) discusses “CANopen” networks, which are “open” CAN networks in that they allow for easy addition of systems and are based on a higher layer protocol than CAN as

they were originally developed for industrial purposes. CANopen networks can be used for general-purpose control applications such as engine control for buses and passenger information systems in public transportation (Zeltwanger, 1999).

2.7 Discussion

The use of electronics in vehicles has revolutionised the automotive industry. Vehicles fitted with EMS's are able to outperform equivalent vehicles with purely mechanical systems under a range of conditions, insofar as improved drivability, higher performance and lower emissions. Stricter government emission legislation has left mechanical fuel systems unable to meet these new standards, as their control systems are not precise enough. Hence, a more permanent solution to emission regulation lies in the improved control of the engine (Jordan, 1987a). Mechanical control systems have some ingenious systems to deal with a variety of engine operating conditions, but electronic control systems are able to manage the engine more efficiently and precisely while also introducing some new features. Mechanical systems are also heavier, have higher wear rates and are more expensive to manufacture and install when compared to electronic systems.

The future of vehicle electronics looks promising with new systems such as mobile multimedia, "drive-by-wire" systems, advanced safety interiors, self-adjusting suspension and collision avoidance systems coupled with smart sensors (Ward and Fields, 2000). Most vehicle improvements desired by consumers are achieved by electronics, providing vehicles that are efficient, safe, non-polluting, comfortable, affordable and with integrated entertainment and communication features (Ward and Fields, 2000). Sensor technology is also evolving as consumer's performance expectations are rising, calibration times are reducing and systems are becoming more complex and dependent on a variety of sensors.

It seems that the main disadvantage of using electronics in vehicles is increased complexity. According to Ward and Fields (2000), future vehicle electronics will be more complex and pose significant challenges to designers. Nevertheless electronics will continue to be used in future, as the advantages outweigh the disadvantages – this can be verified if one considers the increasing use of electronics in vehicles.

Technicians require more specialised training in order to service vehicles fitted with EMS's. However, this can be seen as an advantage as the further education can have social and economic benefits for the technicians. Yet there are still companies and people who resist the change and complain about how easy it was to work on a "good old fashioned" car without complicated electronics. These businesses and individuals may find themselves in the same untenable position as the cynical buggy-whip makers, put out of business by the motor vehicle in previous years (Amey, 1995).

In conclusion, EMS's are beneficial to vehicle performance and are here to stay. EMS's can boast many advantages over mechanical systems with relatively few drawbacks and even then, the disadvantages are largely design issues that won't affect the end user. Electronics allow improved vehicle performance, superior safety and reliability, communications facilities and lower emissions. New telemetry and data logging systems provide opportunities for transport fleet managers to run their fleets more efficiently, economically and ensure destinations are reached in a timely fashion.

EMS's are constantly evolving and new fresh ideas are being developed to break old barriers and improve on existing technologies. Ward and Fields (2000) predict that in future, the power and speed of on-board vehicle computers will approach those of office applications; and software complexity will increase by orders of magnitude compared to what is used in vehicles today. New sensors and sensor-systems with increased capabilities will emerge to meet the advanced information demands of these ever complex systems (Challen and Stobart, 1998). The continuing environmental pressures from society and demands for increased vehicle performance will drive the spread of control technology apace into previously inconceivable areas.

3. VEHICLE PERFORMANCE MANAGEMENT SYSTEMS

Computer based transport decision support systems are available. However, many of these systems focus on driver training and vehicle dynamics. SimTrans and TransSolve can simulate the performance of a diesel haulage vehicle travelling over a route under normal operating conditions. These models were studied and the investigations are presented in this chapter. SimTrans produces a fully comprehensive output of the simulated vehicle performance results over the entire route and was chosen to use for further study.

3.1 TransSolve

TransSolve, Version 7 is a transport management software suite developed by Hellberg Transport Management (Hellberg Transport Management, 2006). The TransSolve suite consists of the following modules:

- **Load Distribution** – This module is aimed at examining the effects of differing loads in various positions on a vehicle. The module then calculates the resulting axle loadings. Other loads on the vehicle such as bodywork, trailers, fridge units, cranes and other equipment can be included in the calculation.
- **Vehicle Performance Simulation** – This module simulates a vehicle's performance over a route. The routes are chosen either by setting waypoints or from a supplied database. The program also allows for routes not on the database to be surveyed and then used in the simulation. Vehicles are chosen out of a database and users have the option to input information on trailers, edit tyre details, retarder description, coefficient of drag, frontal area, vehicle and trailer tare mass, payload, drive axle ratio and maximum engine speed. The maximum allowable vehicle speed can be set and there is an option of setting gear skip changing.

The factors simulated encompass the rolling resistance, wind resistance, maximum engine revolutions, drive-line efficiencies and engine performance. The simulated results include distance, average speed, fuel consumption, trip time, engine load factor,

propulsion and braking work (kW/h), number of gear changes, time in top gear and power to mass ratio. The simulations are useful for determining vehicle suitability and for realistic costing calculations.

- **Operating Cost Estimate** – The module provides the user with a detailed cost breakdown for a transport operation and is split between fixed and variable costs. The fixed costs are further broken down into the following options: the instalment, capital cost, crew, insurance, licence fees, other vehicle expenses, and overheads; while the variable costs include: the cost of fuel and lubricants, tyres, maintenance, toll fees and permits, and other vehicle expenses. Each cost element can be adjusted and there is an option of displaying the costs in terms of R/km, R/ton, R/annum, R/trip, R/hour, R/ton/km. Costs can also be displayed as a percentage of either fixed or variable costs and of the total cost.
- **Finance Calculation** – This module allows the user to compare the financial consequences and cash-flow implications of several financing options for a vehicle procurement. The options are outright purchase, financial lease (residual can also be included in the lease), or structured financing. The user selected option can then be sent directly to the Operating Costs Estimate module.
- **Maintenance Contract Rates** – This module predicts the maintenance and servicing monthly cost and cost per kilometre for a vehicle over a certain period.
- **Specification Comparison** – This module allows the user to compare vehicles in the same class, from different manufacturers through “side-by-side” comparisons. The database of vehicles includes all new vehicles on the South African market.
- **Strategic Information** – The module is a reference system for transport information and includes: altitude tables, approximate vehicle body masses, commodity masses and densities, distance tables, blank operating cost estimate input sheets, vehicle licence fees guide, current legislation summary, toll fees, vehicle performance calculation formulae and vehicle performance terminology.

TransSolve is a useful tool for people in the transport industry as it can be used for solving a range of questions related to haulage operations and can optimise haulage systems. However, TransSolve has limitations. For the simulations, TransSolve has a collection of vehicles currently available on the South African market that can be used. However, the Mercedes Benz 2637 used for the vehicle tests was not an available option in TransSolve. TransSolve is also known to produce errors in the simulations (Lyne, 2004). Errors in the simulated fuel consumption and vehicle speed were two such examples (Lyne, 2004). More importantly, TransSolve does not provide the user with a comprehensive output of simulation results and thus it is not possible to compare simulated performance with measured performance. Hence, there was no means of identifying where or why the program was producing errors as it was a “black box model” and one cannot verify the simulated data.

3.2 SimTrans

SimTrans is a vehicle performance simulation model for use on a personal computer, running under the Microsoft Windows operating system (Cole *et al.*, 1999, Lyne *et al.*, 1996 and Lyne *et al.*, 1998). SimTrans was originally developed in March 1994 by the School of Bioresources Engineering and Environmental Hydrology at the University of KwaZulu-Natal by Professor P W L Lyne, Professor A C Hansen and Mr E de la Harpe as a research project for the South African Sugar Association. Since 1994, there have been several additions and revisions to the SimTrans model. Mr D J Clark investigated forces resisting vehicle motion and improved some of the program algorithms (Clark, 1996). Mr P Govindasamy conducted an emissions prediction research project (Govindasamy, 2002). The version of SimTrans referred to in this dissertation as the original model was Version 3.

The SimTrans model was designed to simulate the performance of a diesel haulage vehicle travelling over a route under normal operating conditions. SimTrans enables the user to examine the performance of various configurations of transport vehicles operating on different road surfaces and profiles. Thus, the user can optimise a haulage operation by considering differing vehicle and route configurations, such as different engine sizes, gearbox ratios, routes, road surfaces and optimum driving strategy.

The SimTrans model was chosen as a suitable model for further investigation, rather than TransSolve for the following reasons. Firstly SimTrans produces a fully comprehensive

output of the simulated vehicle performance results over the entire route, which allows one to compare simulated results to measured performance. SimTrans also has an accessible coding structure (the program is written in Pascal language using Borland Delphi) and there is a host of published literature documenting the SimTrans model development.

3.2.1 User Interface

In SimTrans the user interface enables the user to select data input files and change program settings. The user interface has both a graphical and numeric display of the various simulated vehicle performance data during the simulation.

Input data for SimTrans is entered in the form of three ASCII text input files, each with a specific format: the route profile, speed limit and vehicle file. The route profile file contains the route name, the source of the route data, the starting altitude and a list of the details of each road segment. The road segment description includes the length of the road segment, the segment gradient in degrees, the rolling resistance of the surface expressed as a percentage of the vehicle mass, a data flag and a comment or name associated with the road segment (e.g. dirt road). Data flags are useful in order to instruct the program to stop the vehicle at a certain point, offload the vehicle, set the vehicle's starting speed etc.

The speed limit file contains a list of road section(s) and for each section details the road segment distance and the speed that the vehicle should not exceed until it covers this distance. The speed limit referred to in SimTrans is the lesser of the legal speed limit, the maximum safe speed at which a vehicle can negotiate a section of road or the maximum vehicle speed that can be attained in the current gear ratio (de la Harpe *et al.*, 1998). SimTrans is not able to model driver behaviour and hence the speed limit file is used to help the program choose the correct speed to drive the vehicle at in each situation.

The vehicle input file has the following structure:

- the name of the vehicle and its description;
- the engine name, engine data source and the engine capacity (l);
- the data set of engine speed (r/min) and torque pairs (Nm) used to describe the maximum torque curve of the engine;

- the six constants used to calculate the specific fuel consumption of the engine (kg/h);
- gear box name and data source;
- number of gears, percentage efficiency and change time required (s);
- the ratio of each gear;
- torque converter name and source of torque converter data;
- converter ratio, percentage efficiency and type;
- drive axle name and source of drive axle data;
- reduction ratio and percentage efficiency;
- drag coefficient and frontal area of the vehicle (m^2);
- vehicle type, vehicle mass, payload mass, vehicle geometry details;
- vehicle tyre information;
- if any trailers are being towed by the vehicle, then information on the trailer is required: trailer mass, payload mass and tyre information;
- the greenband limits (r/min).

Once the data files have been entered, the user has the following options in the program:

- Setting the gear selection mode (green band, maximum torque driving or to stay in any one gear);
- Setting the green band upper and lower limits;
- Running the vehicle on a return trip;
- Either running the vehicle loaded or unloaded (if using a return trip, then loaded in both directions or loaded in one direction only);
- The option of saving the results from the simulation either: at the end of each road segment, after each calculation, or no data saving (just a summary screen is displayed at the end of the simulation);
- Setting a percentage wheel slip.

The SimTrans user interface is shown in Figure 3.1 and consists of four separate display panels: the vehicle progress panel, the wheel thrust panel, the engine map panel and the gears panel. The vehicle progress display panel is split into four different sections showing the vehicle specifications; a selection of real-time simulated data; a pictorial representation of the vehicle on the current road segment; and the vehicle's progress on the route profile. At the bottom of the vehicle progress panel are the four simulation control buttons – restart, go, stop

and fast forward (to run the simulation at an increased speed). The wheel thrust display panel displays a graph of wheel thrust versus road speed in each gear. The engine map panel displays a graph of engine torque versus engine speed, or engine power versus engine speed (selected by the user). The engine map panel also has the option of displaying the fuel consumption or specific fuel consumption contours on the graph. Finally the gears panel shows the range of vehicle ground speeds between low and high idle available in each gear ratio. On each graph there is a “cross” indicator showing the current magnitude of vehicle’s thrust, power or torque and the gear selected on each respective graph.

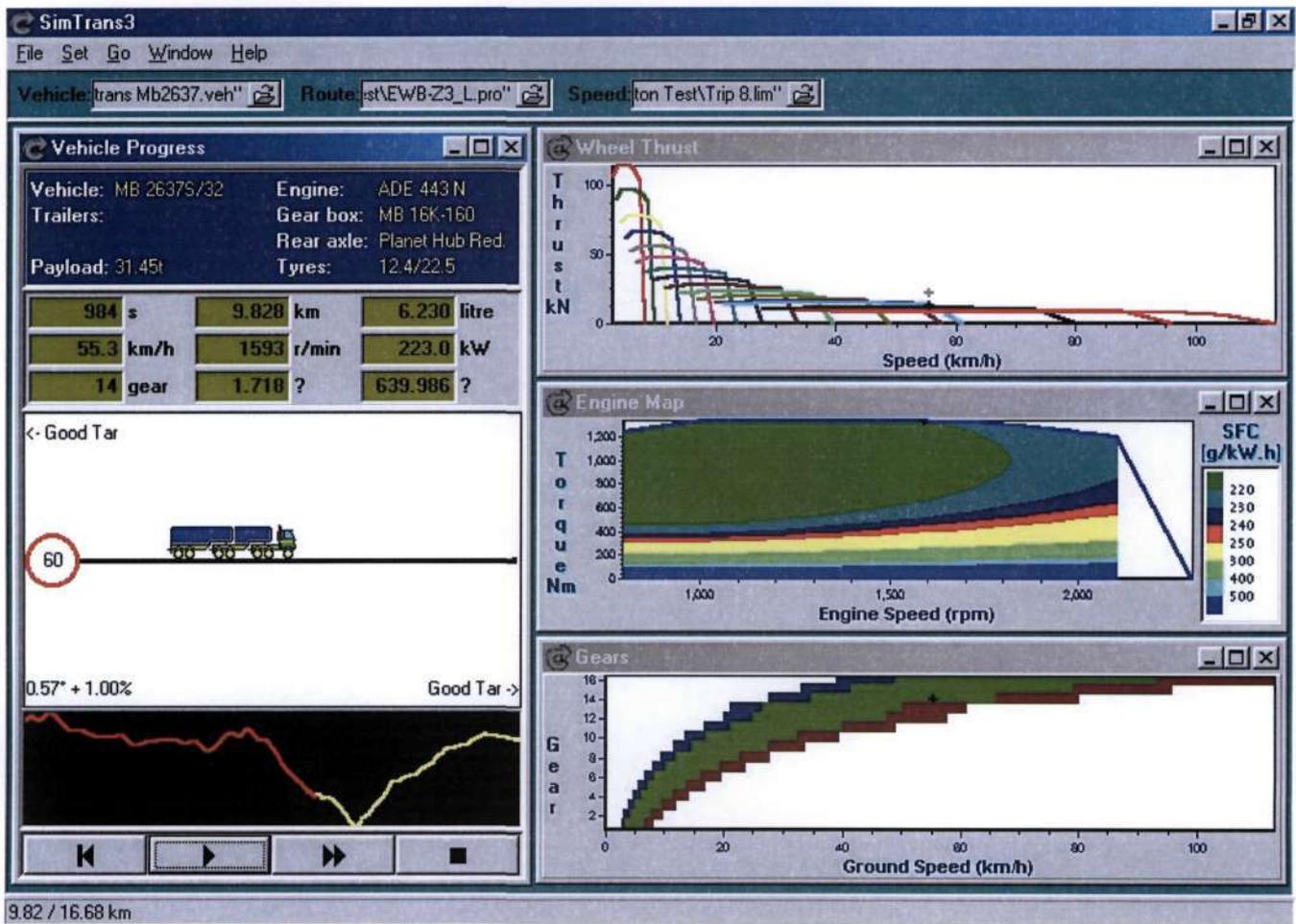


Figure 3.1 The SimTrans user interface

3.2.2 Model Structure

Using the vertical road profile, speed limit and vehicle input file, SimTrans models a vehicle’s performance over the route in half second time increments. The half second time increment is a compromise as using a smaller time increment would result in a more accurate simulation,

but at the expense of a longer simulation operating time and an increased output file size. The model can shorten the time increment to suit the current simulation calculation. Adjusting the time increment is important for determining exact times when the clutch is disengaged or engaged, and when the road segment and/or the speed limit changes. In each time increment the model determines the forces resisting the motion of the vehicle, the torque requirements of the vehicle, the power available from the engine, the vehicle and engine speeds and possible gear changes (if required).

A basic assumption in SimTrans is that while the vehicle speed is below the speed limit, the driver will operate the vehicle under maximum acceleration until the vehicle reaches a speed close to the speed limit. Acceleration is a function of the maximum available thrust in the engine, the resistances to motion and the vehicle mass. Now, just before the vehicle reaches the speed limit, it is assumed that the rate of acceleration is steadily decreased until the vehicle speed is equal to the speed limit. Once the vehicle is travelling at the speed limit and no further acceleration is required, the engine only needs to output sufficient torque to overcome the resistances to motion force. If the speed limit is reduced and the vehicle speed is then above the speed limit, the desired acceleration becomes negative and a braking torque is required to reduce speed. The braking torque is defined as the braking force required to slow a vehicle expressed as an equivalent torque at the flywheel. If the engine braking is insufficient to adequately slow the vehicle then the vehicle brakes are applied. For safety reasons and to prevent vehicle or engine damage, haulage vehicles operating in real life do not accelerate from rest at the maximum rate. Thus in SimTrans acceleration is limited when the vehicle is accelerating from rest.

The following sections will describe each of the procedures and functions which perform the simulation calculations. The flow charts illustrating each of these procedures and functions described below are shown in Appendix B together with a glossary of the variables used.

3.2.2.1 AirResistance Function

Illustrated in Figure B.1, page 95, is the AirResistance function which calculates the air resistance (in kN) on the vehicle. The function firstly makes provision for a headwind, but presently this is set to a constant value of zero. The function then assumes conditions of a standard pressure of 101.325 kPa and an air density of 1.225 kg/m^3 . Air temperature is set at

25°C. The air density is corrected for temperature and pressure variations from standard conditions by (SAE, 1990):

$$\rho = 1.225 * \left(\frac{288.16}{(273.16 + T)} \right) \left(\frac{p}{101.325} \right) \quad (3.1)$$

where ρ = air density (kg/m³)
 T = air temperature (°C)
 p = atmospheric pressure (kPa)

The air density is further adjusted for altitude by:

$$\rho = \rho * \left(1 - \text{Altitude} * \frac{1}{85.3} * \frac{1}{101.325} \right) \quad (3.2)$$

The air resistance force is then calculated by using the fluid drag equation:

$$F_a = \frac{1}{2} \rho \left(\frac{V + Hw}{3.6} \right)^2 A C_d \quad (3.3)$$

where F_a = air resistance force (N)
 V = vehicle speed (km/h)
 Hw = headwind speed (km/h)
 ρ = air density (kg/m³)
 A = projected vehicle frontal area (m²)
 C_d = coefficient of drag

The vehicle frontal area and the coefficient of drag are read from the vehicle input file.

3.2.2.2 GradeResistance Function

The GradeResistance function is illustrated in Figure B.2, page 95, and calculates the grade resistance by determining the magnitude of the vehicle weight acting parallel to the road surface:

$$R_g = GVM * g * \sin(\alpha) \quad (3.4)$$

where R_g = grade resistance (N)
 GVM = gross vehicle mass (kg)
 g = acceleration due to gravity = 9.81 (m/s²)

α = road grade (degrees)

The gross vehicle mass is read from the vehicle input file while the road grade is obtained from the route profile input file.

3.2.2.3 RollingResistance Function

The function RollingResistance is shown in Figure B.3, page 96. The rolling resistance force on the vehicle is determined from the rolling resistance force on the tyres acting parallel to the road surface and factored by a surface related rolling resistance coefficient:

$$RR = GVM * g * \cos(\alpha) * \left(\frac{SurfaceRR}{100} \right) \quad (3.5)$$

where RR = rolling resistance (N)

$SurfaceRR$ = rolling resistance coefficient

The rolling resistance coefficient is influenced by several variables including the deformation of the road surface, which is assumed to be zero on hard surfaces; tyre deformation which is influenced by load, temperature and inflation pressure; tyre design; materials and torque transmission (Clark *et al.*, 1996). Determining the value of the coefficient for each section of road surface is a somewhat subjective process, however, the values as listed by Webster (1995) in Table 3.1 should be used as a guide. The surface rolling resistance coefficients are read into the model from the route profile input file.

Table 3.1 Equivalent percentage rolling resistance coefficients for radial and bias ply tyres on various travel surfaces (after Webster, 1995)

| Type of Road | Road Condition | Bias Ply Tyres | Radial Ply Tyres |
|----------------------|----------------|----------------|------------------|
| Concrete and Asphalt | Good | 0.98 | 0.79 |
| Asphalt | Traffic worn | 1.22 | 0.98 |
| Gravel | Dry average | 2.00 | 1.60 |
| | Rough | 2.45 | 1.97 |
| | Loose | 4.50 – 5.00 | 4.50 – 5.00 |
| Sand | | 10.00 | 10.00 |

The RollingResistance function reduces the rolling resistance force by 20% for vehicles fitted with radial ply tyres.

3.2.2.4 GetMaxEngTorque Function

The GetMaxEngTorque function is illustrated in Figure B.4, page 97. This function determines the maximum torque available from the engine at a particular engine speed. When SimTrans loads the data from the vehicle input file, a lookup table is generated with corresponding engine speed and torque pairs depicting the maximum throttle torque curve for the engine.

The function firstly looks at whether the current engine speed is less than the lowest engine speed in the lookup table. Should this be the case, it is assumed that there is a linear drop off in maximum torque. The torque is then estimated by interpolating between the torque value corresponding to the lowest engine speed in the table and an engine speed of 0 r/min and 90% of this torque value. If however, the engine speed is greater than the rated speed, then the maximum torque is determined by interpolating along a straight line between the maximum torque at the rated speed and zero torque at 200 r/min above the rated speed. Finally, if the engine speed is greater than the lowest engine speed in the lookup table (this speed will, by deduction be less than the rated speed), then the current maximum torque is found by searching through the lookup table. Two engine speed pairs on either side of the current engine speed are found and by interpolating between these values the maximum torque is then determined.

3.2.2.5 GetTorqueLosses Function

The GetTorqueLosses function is illustrated in Figure B.5, page 98, and calculates the torque lost in a vehicle's transmission system due to friction and other inefficiencies. If the current engine torque is less than zero, then the transmission losses are assumed to be zero. For positive torque values, the transmission losses are constant for a specific vehicle and these losses are calculated using the gear box efficiency and the drive axle efficiency. These efficiency values are read from the vehicle input file. For example in the case of the Mercedes Benz 2637 (see specifications in Table H.1, page 124 in Appendix H), which has a gear box efficiency of 90 % and a drive axle efficiency of 90 %, the transmission losses are calculated by:

$$\text{Losses} = 100 * \left(1 - \left(\frac{90}{100} * \frac{90}{100} \right) \right) = 19\%$$

3.2.2.6 GetBrakeTorque Function

The GetBrakeTorque function is illustrated in Figure B.6, page 98. Initially the minimum torque value is determined based on the current engine speed and engine size. Minimum torque is defined as the friction torque at the flywheel that a particular engine can produce at a given engine speed. If the current engine torque value is greater than zero, then it is assumed that the driver can still rely on engine braking to slow the vehicle (if required). However, should the current torque value be less than zero, then a braking torque value is determined from the negative of the sum of the torque and the minimum torque value. If the braking torque is less than zero then it is assumed that the driver can still get enough braking torque from the engine and hence no braking force is required.

3.2.2.7 SelectGear Function

Figure B.7.1 and Figure B.7.2 on pages 99 and 100, show the flow chart of the SelectGear function. This function determines if a gear change is required and if so what gear to select, given the current vehicle ground speed and motion resistance force. Firstly the function calculates the deceleration and corresponding vehicle speed drop which will occur while the clutch is disengaged during the gear change. Depending on the value of the motion resistance, the speed drop may become negative and thus become a speed gain. The current engine speed in the currently selected gear is then determined by using the vehicle ground speed.

SimTrans allows two gear change methods, namely green band driving and maximum torque driving. For diagnostic purposes a third gear change method exists where the vehicle can be specified to drive in a single gear during the entire simulation (this option is meant primarily for agricultural tractors).

In green band driving the vehicle is driven with the engine operating between two specified engine speeds (i.e. the green band). The green band limits are either read from the vehicle input file or they can be specified in the user interface. If the current engine speed is greater than the upper green band limit and the top gear (i.e. the lowest ratio) is not selected, then a possible gear up shift will be investigated. A new engine speed in the next highest gear is then calculated based on the current vehicle speed less the speed drop due to the gear change.

This new calculated engine speed is then used to determine the maximum engine torque, the transmission losses and thus the new engine torque value. If this new maximum engine torque is greater than the resistance force converted into an equivalent torque at the engine, then a gear up shift is made.

The vehicle will conversely only shift down a gear if the engine speed becomes less than the lower green band limit and first gear is not selected. The new engine speed in the next lowest gear is then calculated based on the current vehicle speed less the speed drop due to the gear change. If the new engine speed is less than the rated speed then a gear downshift is made.

In maximum torque driving, the vehicle is driven with the maximum possible driving torque for the current vehicle ground speed, without going into the governed region unless in top gear. The maximum engine torque in the current gear ratio is determined, followed by a calculation of the engine speed using the current vehicle speed in the next highest gear ratio. The new maximum engine torque is then determined based on this new engine speed in the next highest gear. If the wheel torque in the next highest gear is greater than the wheel torque in the current gear then a gear up shift is investigated, otherwise a gear downshift is investigated. If a gear up shift is to be investigated the new engine speed is determined using the current vehicle speed less the speed drop in the next highest gear. The maximum torque for this new engine speed is then determined. The current engine speed (OldRPM) is determined but also accounting for the speed drop due to the gear shift. The current maximum engine torque is then determined based on the OldRPM, i.e. OldTorque. Should the wheel torque in the next highest gear be greater than the wheel torque in the current gear accounting for the speed drop, then a gear up shift is made. Vice versa, if a gear downshift is investigated, the new engine speed, maximum engine torque, current engine speed and current maximum engine torque are calculated all taking into account the speed drop. If the wheel torque in the next lowest gear is greater than the wheel torque in the current gear, accounting for the speed drop, then a gear downshift is made.

3.2.2.8 GetFuel Function

The GetFuel function is shown in Figure B.8, page 101. Firstly there is a check to test if the engine is over-revving. If the engine speed is greater than the rated speed plus a proportionally calculated engine speed using the torque, then the fuel flow is set at zero. In

all other circumstances, the fuel consumption is calculated using a polynomial regression equation based on engine speed and torque:

$$\text{Fuel Flow} = C_1 + C_2 \cdot \text{RPM}^2 + C_3 \cdot Tq^2 + C_4 \cdot \text{RPM}^2 \cdot Tq + C_5 \cdot \text{RPM} \cdot Tq^2 + C_6 \cdot \text{RPM} \cdot Tq \quad (3.6)$$

where *Fuel Flow* = fuel consumption (kg/h)

$C_1 \dots C_6$ = regression constants

Tq = engine torque (Nm)

RPM = engine speed (r/min)

The fuel flow regression constants are uniquely determined for each different engine by a multiple regression analysis of the engine specific fuel consumption contour map. These constants are stored in the vehicle data file.

If the fuel consumption value is less than zero, it is assumed that the governor has shut off fuel to the engine and thus fuel is set to zero. If the current engine torque is a braking torque and is thus less than the negative value of 40% of rated torque, then fuel flow is set equal to a value of zero.

3.2.2.9 Main Procedure

The Main procedure is illustrated in Figure B.9.1 through to Figure B.9.5 on pages 102 to 106 and is the “engine” of the program controlling the simulation of the vehicle in SimTrans. Initially the time step variable (dt) is set to a value of 0.5 s and the time to change gear value is set at 1.5 s (setting the time to change gear value at this point is an error and it is corrected in the modified Main procedure). If the total time elapsed is less than zero, i.e. the vehicle is at the beginning of the simulation, then the “previous” gear selected (the OldGear variable) is set to first gear and the time a gear change last occurred is set to an initial value of -5 s. The reason for the OldGear variable is so that the programme is able to determine, through comparison of the current gear verses OldGear, if a gear change has occurred.

Next, if the vehicle is stationary, it is assumed as an initial value that the engine speed is 1000 r/min. The vehicle starting speed is then determined for this engine speed. If the vehicle has not travelled any distance then a check is made as to whether the user has specified a return trip, and if so then the current altitude is set to the last altitude in the route file. Otherwise the

vehicle is at the start of the route and the current altitude is the first altitude specified on the route profile. It is then determined if the vehicle is at the start of the route and if the first value in the speed limit file is greater than 1000. If this condition is met then the speed limit value is a Flag and it is specifying the vehicle's starting speed. The procedure then determines the appropriate gear that the vehicle should be driving in with respect to the current vehicle speed by stepping through each gear until the engine speed is less than the green band upper limit.

The next check made is to determine if the vehicle is required to be offloaded and if so the gross vehicle mass is set equal to the vehicle mass (read from the vehicle file). There is then a check whether the Flag value (read from the route file) divided by 4 is an odd number, which controls the implement setting. There is a check as to whether the surface rolling resistance coefficient is greater than 100 (i.e. in soft sand). These two checks are intended for calculations involving tractors with implements. However, studying this particular feature of the programme is not an objective of this dissertation and thus sections relating specifically to agricultural tractors will not be discussed.

The motion resistance functions are called and a total resistance force on the vehicle (excluding the transmission losses) is calculated by summing the air resistance, grade resistance and rolling resistance forces. The thrust value is then initialised at a value of 1 N.

It is then checked whether the total elapsed time minus the last time a gear change occurred is less than the time to change a gear, and if so implies the clutch is disengaged. If the total elapsed time minus the last time a gear change occurred is greater than the time to change a gear, then the value of dt is adjusted accordingly (it is to be noted that this check is in error and it is corrected in the modified programme). The engine speed is set to the lowest value in the lookup table (idle speed) and the engine torque and thrust are set to zero.

If however, the clutch is engaged and thus no gear change is in progress, then a possible gear change is investigated. Firstly the current "gear mode" is checked to see if the vehicle must be forced to drive in a single specified gear (this feature is useful for agricultural tractors and for possible diagnostic purposes). The torque converter values are set to 1 to nullify their effect on the calculations should the vehicle not be fitted with a torque converter. The torque converter sections of SimTrans shall be ignored in this document as no torque converter was

fitted in the test vehicle and studying the effect of a torque converter is not an objective of this dissertation. The SelectGear function is called. If a gear change did occur then the current elapsed time is recorded, the engine speed is set to idle, torque and thrust are set to zero and OldGear is set to the gear the vehicle was driving in before the current gear change.

If a gear change is not in progress and thus thrust has a value other than zero, the engine speed is calculated along with the maximum engine torque (by calling the GetMaxEngTorque function) and the torque losses in the transmission (by calling the GetTorqueLosses function). The current thrust available at the wheels is then determined using these calculated values. The next step is to calculate the “desired acceleration”, which is the acceleration that would be required to accelerate (or decelerate) the vehicle to reach the speed limit in the next second. This desired acceleration is limited to either $+ 1.5 \text{ m/s}^2$ or $- 1.5 \text{ m/s}^2$ accordingly to prevent the vehicle from accelerating at an unrealistic pace when starting or stopping. If the surface is hard, the required wheel thrust is determined based on the desired acceleration. If the required thrust is greater than the available thrust at the wheels, the vehicle acceleration is then recalculated using the available thrust. However, should there be more than sufficient thrust available to accelerate the vehicle; the thrust is then calculated based on the desired acceleration.

The procedure then calls the GetBrakeTorque function to determine if the vehicle brakes are required or if sufficient braking force can be obtained by engine braking. The resultant torque is then calculated. The procedure then calls the GetMaxTorque function and compares whether the current engine torque is less than the maximum torque at the current engine speed (less a small tolerance). If the engine power is found to be limiting, this fact is recorded and this is used in the summary page at the end of the simulation. The current vehicle speed is then determined from the acceleration over the current time period. A check is made as to whether the vehicle speed is negative and if so the vehicle speed is set to zero (this is included as a program error check to ensure the speed remains positive or zero).

The fuel consumption is then determined by calling the GetFuel function. Next the procedure calculates the distance the vehicle will travel in the current time step using an equation of motion. Should this distance be larger than the remaining distance of the current road segment, then the time step is adjusted accordingly. Firstly the vehicle speed is recalculated without acceleration. If the acceleration is zero then the new time step is simply the

remaining distance in the current road segment divided by the vehicle speed. However, if the acceleration is greater than zero then the equation of motion is solved for time.

$$s = ut + \frac{1}{2}at^2 \quad (3.7)$$

where s = distance (m)

u = speed (m/s)

a = acceleration (m/s²)

t = time (s)

If it is determined that the time step is zero, then the vehicle is already at the end of the current road segment. If the time step has a value greater than zero, then the vehicle speed and distance travelled in the time step are recalculated using this new time step value. Finally the current route altitude and total distance travelled after the current time step is then determined.

After the Main procedure has executed, the calculated simulation data for the time step just completed are sent to a Progress Unit. This unit updates the user interface with the relevant simulated data and saves the results in a file (if specified). The Main procedure is then repeated until the vehicle has covered the specified route in the route profile file and if required, also completed a return trip. At the end of the simulation, a summary screen is displayed in the user interface showing various averaged or summed results from the simulation.

4. MEASUREMENT OF VEHICLE PERFORMANCE

To achieve the aims of this dissertation, an observed dataset was required to compare with simulated performance from SimTrans. This chapter focuses on the development and calibration of the Vehicle Performance Monitoring System (VPMS) which was used in gathering the observed data. The discussion moves on to the installation of the VPMS on a haulage vehicle and the development of the files required for the SimTrans simulation.

4.1 Development of a Vehicle Performance Data Acquisition System

In order to obtain an accurate observed data set from a haulage vehicle, the Vehicle Performance Monitoring System was developed at the School of Bioresources Engineering and Environmental Hydrology at the University of KwaZulu-Natal, by Mr R L Herbert and Mr R van Zyl. The VPMS consisted of a Campbell Scientific CR10X Data Logging System as the central control device and data storage facility, with several instruments connected to the data logger to measure and record data. Instruments used were:

- Trimble SVEEeight Plus GPS receiver – to measure the vehicle ground speed and geographic position;
- Two Flowmate (OVAL M-III) Model LSN45L fuel flow meters – to measure fuel consumption and fuel flow rate;
- The vehicle alternator tachograph signal – to measure the engine speed;
- The vehicle alternator tachograph signal – to measure vehicle speed (providing a backup to the GPS data);
- A linear potentiometer – to measure the throttle position.

The logging system was powered by a 12V battery. Software was developed for the CR10X by Mr R L Herbert, to control the sampling of sensors, process the data as required and then record the received data. An interactive keyboard with a liquid crystal display could be connected to the VPMS to enable the user to view data or to set program settings. This chapter will describe the VPMS components, development and implementation.

4.1.1 Fuel Flow Rate Measurement

Fuel consumption was measured with a Flowmate (OVAL M-III) Model LSN45L, positive displacement fuel flow meter. This flow meter was capable of measuring a minimum flow rate of 2.5 l/h and a maximum of 500 l/h. The fuel flow meter had a magnetic sensor and produced a digital pulse proportional to a set quantity of fuel passing through the meter. In order to measure the fuel flow rate in a diesel engine, the fuel flow on both the inlet to the injector pump and the flow in the return line into the fuel tank needs to be measured. Thus two flow meters were used (model serial numbers 83014 and 83015). The total fuel consumption of the engine can then be calculated by finding the difference between the readings of the two fuel flow meters.

4.1.1.1 Flow Meter Calibration

The two flow meters were individually calibrated by passing a volume of fuel through each flow meter, timing the flow via a stop watch and measuring the corresponding number of pulses produced in this period. The flow rate was then calculated by dividing the volume collected by the time taken. The calibrating apparatus consisted of a fuel reservoir mounted above the flow meter with a tube connecting the outlet of the reservoir to the inlet of the flow meter. A stop cock was connected inline with the flow meter on the inlet side to control the flow into the meter. The outlet of the flow meter drained the fuel into a measuring flask. The test was repeated with several different flow rates ranging from 9 l/h to 158 l/h. The total volume collected in each test was between 1.7 l and 2 l. The volume reading on the measuring flask was checked by weighing the collected fuel and measuring the density using a hydrometer. The volume was determined by the mass divided by density. See Figure C.1, page 107 and Figure C.3, page 108 in Appendix C for the graph of flow rate versus number of pulses for each flow meter.

A calibration constant for each flow meter was achieved by plotting the pairs of corresponding volume collected against number of pulses measured. A linear regression line was plotted through the points and the slope of the line is thus the calibration constant, which was determined as 2.50 ml/pulse for both meters (83014 and 83015). The graphs of this calibration are shown in Figure C.2, page 107 and Figure C.4, page 108 in Appendix C.

4.1.1.2 Flow Meter Digital Signal Conditioning Board

The CR10X was unable to directly measure the pulse rate produced by both flow meters, due to internal limitations of the CR10X. Thus a signal conditioning board, dubbed the Dual Fuel Period Calculator, was developed by Mr R van Zyl to enable the pulse outputs of the flow meters to be read by the CR10X through some real-time local processing on the board. The Dual Fuel Period Calculator operated by measuring the period between the last two pluses received from each flow meter and it then outputted this period (ms/pulse). The CR10X requests the period data by setting a control port high, after which the Dual Fuel Period Calculator then sent two values of four characters, each value corresponding to the period measured for the supply and return flow meters.

The fuel flow rate (ml/s) was calculated by inverting the period (ms/pulse) and multiplying by the calibration factor, i.e.:

$$F = \frac{1}{P/1000} * C \quad (4.1)$$

Where: F = fuel flow rate (ml/s)

P = period (ms/pulse)

C = calibration factor (ml/pulse) = 2.5 ml/pulse

However, it was found that calculating a fuel consumption total (I) using the fuel consumption rate (ml/s) calculated from the inverse period readings multiplied by time, did not provide an accurate result. The reason for this was that the period measurements were instantaneous samples. Thus the Dual Fuel Period Calculator was given another function to maintain a separate running total for each flow meter by counting every pulse produced by each meter. To convert these pulse counts to fuel consumption, the total was multiplied by the calibration factor.

4.1.1.3 Fuel Flow Rate Measurement System Testing

The flow meters, Dual Fuel Period Calculator and CR10X were installed as a stand-alone system on an engine in the dynamometer engine test bay at the Agricultural Engineering Workshops at the Ukulinga Research Farm of the UKZN. This was in order to test the accuracy of the fuel flow rate measurement equipment intended to be used for the VPMS.

The volume of fuel supplied to the engine in the test bay was measured via a graduated supply tank. The volume of fuel returned was measured in a separate flask. The engine was run until two litres of fuel had passed through the supply line into the engine and the time and the returned volume of fuel were measured. Using this information the consumed volume of fuel and the fuel flow rate were calculated. The CR10X recorded the Dual Fuel Period Calculator's measurements of fuel flow from the flow meters during this same time period. This procedure was repeated over twelve separate tests, except in tests 10, 11 and 12 where the fuel return line was connected to the supply tank to reduce error by eliminating the flask read error for the return volume. The data recorded by the CR10X was then compared to the actual observed data. See Table D.1, page 109 in Appendix D for the tabulated results.

The percentage error between the measured data to the observed data was determined for every test, for each of the time duration of the test, supply volume, return volume, used volume and fuel flow rate. These results were plotted and are illustrated in Figure D.1, page 109 in Appendix D. The measured data error lies between $\pm 1\%$, with one outlier of -1.17% which could possibly be due to some human read error of the measuring flask. The fuel flow meter's manufacturer quotes an accuracy of $\pm 1\%$ for the meters. Thus errors observed in these tests are within allowable tolerances according to the manufacturer.

4.1.2 Throttle Position Sensor

The Throttle Position Sensor used in the VPMS (shown in Figure 4.1) was developed by Mr R van Zyl and was designed to measure the position of the vehicle's throttle. The Throttle Position Sensor was a linear potentiometer connected by a cable to the throttle control lever. The sensor's housing was constructed from a cylindrical plastic tube with the potentiometer mounted on the inside of the housing. The cable between the potentiometer and the throttle control lever was connected to a retractable coil spring with a similar mechanical method of operation to that of a tape measure. As the throttle control lever was depressed, the cable was drawn into the housing due to the action of the coil spring. The movement of the cable generated a voltage output on the potentiometer which was then measured by the CR10X. In order to calibrate these readings, all throttle position sensor outputs measured by the CR10X over the entire vehicle performance tests were analysed to determine the maximum and minimum values. The difference between these upper and lower readings was then used to convert the outputs into a percentage.



Figure 4.1 The Throttle Position Sensor installed in the test vehicle

4.1.3 Global Positioning System Receiver

Global Positioning Systems (GPS) provide an accurate and convenient means of determining geographic position. A Trimble SVeeEight Plus GPS receiver (SV8) was used for the VPMS. A useful feature of the SV8 was the local processing power built into it, enabling the receiver to determine the horizontal ground velocity of the antennae by calculating the relative movement of the antennae over time. The SV8 has eight processing channels operating on the L1 frequency of 1575.42 MHz and has a positional accuracy of 25 m (Circular Error Probability) and a velocity error of 0.1 m/s.

Due to the limitations of the CR10X serial data bus acquisition system, the data string sent out by the SV8 receiver was unable to be read directly by the CR10X. Thus a GPS post-processing board, dubbed the GPS black box, was developed by Mr R van Zyl to accept the data string from the SV8 and convert and delimit the data string to make the data string useable by the CR10X. The post processed data string sent by the GPS black box comprised:

- A GPS signal quality indicator;
- The SV8 measured antennae horizontal and vertical speed (miles/h), accurate to two decimal places;
- The latitude and longitude (degrees) and accurate to seven decimal places;
- The altitude (ft above mean sea level) and accurate to two decimal places.

The GPS quality indicator or age of data indicator could have three values:

- 2 if the GPS data was less than 10 seconds old (accurate data);
- 1 if the GPS data was greater than 10 seconds old (data still useable);
- 0 if the GPS data was invalid.

GPS data was only stored if the age of data indicator was greater than 0.

4.1.4 Speed Measurements

The vehicle ground speed and the engine speed were measured by connecting the CR10X to the vehicle's electronic alternator tachograph system and these signals were read as low level AC pulse inputs. In order to prevent damage to both the tachograph and the CR10X, the pulse outputs from the tachograph were passed through an optical isolator before being connected to the CR10X.

In order to calibrate the engine speed readings, actual engine speed measured with a hand tachometer at the engine flywheel was compared to the tachograph outputs at several throttle position settings. The pairs of actual engine speed (r/min) and measured pulses from the tachograph were plotted on a graph and a linear regression line was plotted through the points. The slope of the line is the calibration constant for engine speed and was found to be 2.265 r/min/pulse. The dataset for this calibration is shown in Table E.1, page 110 and the graph in Figure E.1, page 110 in Appendix E.

Although the vehicle speed was obtained from the tachograph system, the vehicle speed measured by the SV8 was used for the observed data set. It was assumed that the vehicle speed measured by the SV8 was more accurate, as the manufacturer quotes an accuracy of 0.1 m/s for the SV8's speed measurements, while there was no information available as to how accurate the tachograph readings were. Nevertheless, the data logging software required the vehicle speed to be able to determine when the vehicle was in motion or at rest, which was then in turn used to instruct the CR10X when to collect vehicle performance data. The vehicle speed outputs from the tachograph were quicker to measure than to obtain the vehicle speed readings from the SV8 and thus the tachograph readings were used by the CR10X software for control purposes.

During the vehicle performance tests it was found that in the first test, for approximately the first half an hour, GPS data was unavailable. Thus the tachograph vehicle speed data was used to fill in the vehicle performance data set in this instance. Hence it was necessary to determine a calibration constant for the vehicle speed readings acquired from the tachograph, by comparing these outputs to the vehicle speed obtained from the SV8. The pairs of tachograph speed outputs were plotted against the vehicle speed from the SV8. A regression line was plotted through the points and the slope of the line is the calibration constant, which was found to be 0.36 km/h/pulse. The plot of the calibration is shown in Figure F.1, page 111 in Appendix F. In the plot there are outliers lying greater than 20 % off the regression line. It is believed that the outliers are the result of power spikes in the vehicle's electrical system causing abnormal tachograph readings. These outliers account for less than 1 % of the points plotted and were thus deemed not significant.

4.1.5 Monitoring System Software Development

The CR10X requires control software to manage data acquisition and data storage. Software to run the CR10X in the VPMS was developed by Mr R L Herbert and was designed around the following criteria:

- Stored data is only required when the vehicle is in motion;
- The program execution interval must be sufficiently fast such that the dataset produced will provide as accurate as possible log of the various instantaneous vehicle performance parameters over time;
- GPS position data is only required when the vehicle has moved a specified distance or the program has progressed through several loops;
- When the vehicle pulls off and when it comes to rest a "stop condition" must be recorded with totalised fuel consumption data;
- The voltage of the battery powering the CR10X is required to be checked every 30 minutes to ensure data is valid.

When determining the program execution interval for the CR10X, several factors were considered. Firstly every program instruction takes a certain time period to execute and the time required varies with different instructions. Sampling of sensors and waiting for data to be transmitted back to the CR10X requires more time than processing instructions. Time

must also be allowed for data storage. Thus it was determined through careful examination of the VPMS operation with an oscilloscope that the shortest possible program execution interval was 1.5 seconds. This interval would allow sufficient time for all sensors to be sampled, GPS data to be acquired and time to store the data. An execution interval of six seconds was chosen in between receiving GPS position data as this interval provided the best compromise between conserving the data logger's storage memory and still achieving an accurate plot of the vehicle's motion over the route.

The control program was divided into four procedures. The main procedure contains the conditional checking and determines when data should be collected and stored. There are three subroutines which are called by the main procedure. The algorithm for the main procedure of the program is illustrated in Figure G.1.1, page 112 and Figure G.1.2, page 113 in Appendix G and is described below. A printout of the entire control program used for the CR10X is shown in Figure G.5 on page 117.

The main procedure initially samples the engine speed and vehicle speed sensors. Next a check is made to a diagnostic flag (Flag 1), which when set high can be used to override the program's control of operations. However, under normal operating conditions Flag 1 is set low and the program then proceeds through the following checks:

- If the engine is off (i.e. the engine speed is less than zero plus some tolerance), then Flag 2 is set low;
- If the engine is running (i.e. the engine speed is greater than or equal to zero plus some tolerance), then Flag 2 is set high;
- If the vehicle is not in motion (i.e. the vehicle speed is less than zero plus some tolerance), then Flag 3 is set low;
- If the vehicle is in motion (i.e. the vehicle speed is greater than or equal to zero plus some tolerance), then Flag 3 is set high.

The program then checks if the engine is running (Flag 2 set high) and if so, the program then checks if the vehicle is in motion (Flag 3 set high). If this is true a further check is made to a counter variable. The counter variable is intended to allow the program to only save the "stop condition" (i.e. execute Subroutine 3) when the vehicle pulls off or when the vehicle comes to a rest. The "stop condition" creates a break point in the data set and saves the totalised fuel

consumption. Subroutine 1 and Subroutine 2 are then called and the counter variable is then cleared to prevent the “stop condition” from being saved again while the vehicle is in motion. Now, if the vehicle is stationary (Flag 3 is low) there is another check on the counter variable. If the counter is found to be zero, i.e. the “stop condition” has not been saved yet, Subroutine 3 is called. The counter variable is then set to a value of one such that the “stop condition” can be saved only once the vehicle pulls off again. A GPS position data saving interval timer is then set to zero to ensure that once the vehicle pulls off the timer is clear.

The program then checks if the vehicle’s engine is off (Flag 2 set low) and if so, the counter variable used to control execution of the “stop condition”, is set to zero. It was found through testing of the VPMS that when the SV8 is powered up, the SV8 outputs a Position Velocity string every second, which interferes with the VPMS operation. Thus a command is sent to the SV8 to stop the Position Velocity string being transmitted. This command is only required to be sent once, when the GPS has powered up. However, it was anticipated that there could be a power dip or shortage on the logging system at any stage in the program execution. Thus it was assumed that there was a need to send this command in the program – not just at the initialising stage. This command is executed in this loop of the program since at this point in the program there is some “spare” program time as the vehicle’s engine is off and the execution time for the program is thus not limiting.

Finally the program measures the voltage of the battery powering the logging system every 30 minutes. The battery voltage along with the real time in days, hours and minutes, is then stored to the CR10X’s memory. The CR10X then reaches the end of one cycle or execution interval and then re-starts the program from the beginning.

Subroutine 1 of the program is illustrated in Figure G.2, page 114 and is described below. Subroutine 1 commences by firstly sending a request to the Dual Fuel Period Calculator for the fuel flow rates in the supply and return lines. Next the Throttle Position Sensor is sampled. The subroutine then finally stores the real time (day, hours / minutes and seconds), the vehicle speed, engine speed, fuel flow in the supply and return lines. The vehicle speed and engine speed are obtained in the main procedure before this subroutine is called.

Subroutine 2 of the program is illustrated in Figure G.3, page 115 and is described below. Subroutine 2 collects and stores the data gathered from the SV8 receiver. Initially the

procedure sets the age of data indicator equal to zero so as to prevent previous erroneous data from influencing the new data. The procedure then turns on the power to the GPS black box. The power is suspended from the GPS black box at all other times of the program execution for two reasons: firstly to save power; and secondly to prevent the GPS black box from locking up if it receives a data string from the SV8 other than the requested string. The Position Velocity string sent by the SV8 (mentioned above) could cause a lock up in the GPS black box.

Next Subroutine 2 sends a command directly to the SV8 to request the TAIP Long Navigation String. The SV8 outputs the requested string, which is then post-processed by the GPS black box to convert the string into the required format. The converted string is then passed back to the CR10X. Data received is read into variables: age (age of data indicator), latitude, longitude, altitude, vertical speed, horizontal speed and heading. As soon as the CR10X has received the data string from the GPS black box, the subroutine turns off the power to the GPS black box.

Subroutine 2 then checks the age of data indicator as to whether the data is valid – a value of one or two for the age of data indicator signifies valid data. The subroutine then stores the real time (in day, hours / minutes and seconds), vertical speed, horizontal speed and the heading. Now, since the GPS position data is only required to be stored every six seconds, the program must cycle through four loops before the GPS position data is stored (execution interval is 1.5 s). In order to ensure that this cycle is maintained, a timer variable is used to enable the program to determine when to save the data. The timer variable is then checked as to whether it is greater than or equal to five. A value of five or more indicates that the program has just cycled through the loop when GPS position data was to be saved, thus the timer is set equal to zero. The subroutine then checks if the timer variable is equal to four. If true, the subroutine then stores the age indicator, latitude, longitude and altitude. The subroutine then increments the timer variable by one.

Subroutine 3, controls the “stop condition” of the program and is illustrated in Figure G.4, page 116 and is described below. Subroutine 3 first calls the Dual Fuel Period Calculator for the summed fuel consumption values from the fuel supply and return line. Subroutine 3 then stores the real time (day, hours / minutes and seconds) and the summed fuel consumption from the supply and return lines.

Subroutine 3 then checks if the vehicle is stationary (Flag 3 set low). If the vehicle's engine has just started and the vehicle has not moved, or the vehicle has just come to rest, a value of one is stored at the end of the fuel consumption data set. Saving this value aids post-processing of the data, as one can differentiate when the vehicle is moving off or stopping.

4.1.6 Monitoring System Testing and Installation

The final stage in the development of the VPMS was to install the system on the engine in the dynamometer engine test bay at the Agricultural Engineering Workshop at the Ukulinga Research Farm of the UKZN. This process was to test the reliability and accuracy of the data logging system. The dynamometer was run at differing speeds and loads to simulate a vehicle operating at differing conditions. As this was purely a testing phase no results are presented here from these tests. Figure 4.2 shows the case containing the CR10X, SV8 and some of the other instruments used in the VPMS.

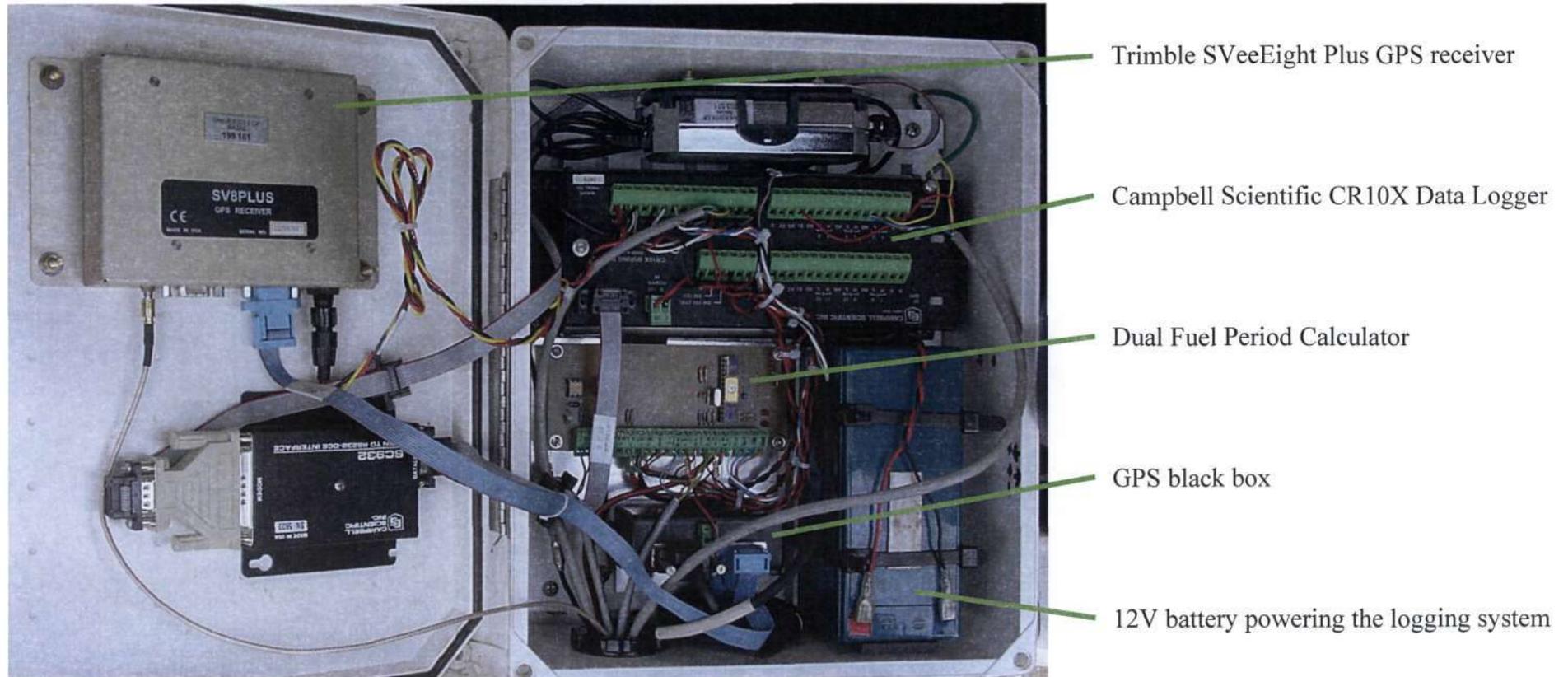


Figure 4.2 The case containing the core instruments of the VPMS used for the vehicle performance tests

4.2 Haulage Vehicle Performance Measurements

As outlined in the objectives of this project, in order to validate the SimTrans model, one would need to compare a measured dataset, captured from a haulage vehicle operating under normal conditions, to simulated data from SimTrans for the same vehicle. The haulage vehicle used for the performance tests was a Mercedes Benz 2637 V Series (shown in Figure 4.3) from the Unitrans Freight Sugar and Agriculture Company. The Mercedes Benz 2637 was fitted with an ADE 443N engine, which is a V10 naturally aspirated 260kW diesel engine. The vehicle was coupled to a conventional set of interlink trailers used to haul timber. The vehicle specifications are tabulated in Table H.1, page 124 in Appendix H. The graphs of torque versus engine speed and power versus engine speed for the ADE engine fitted to the truck are shown in Figure H.1 and Figure H.2 respectively, on page 125 in Appendix H. The vehicle driver was experienced and operated the engine revolutions strictly within the greenband range.



Figure 4.3 The Mercedes Benz 2637 haulage vehicle used in the vehicle performance tests

The VPMS once designed and tested, was installed on this vehicle and used to gather performance data. The haulage tests were performed on the 15th and 16th of January 2004.

The vehicle was driven on two alternative routes between the Eston Sugar Mill and a timber compartment near Mid-Illovo, designated as Zone Three. The first route, referred to as Route One was via the Umbumbulu (R78) road and was tarred and in good condition throughout the distance. The second route, referred to as Route Two, was partly tarred with a worn tar surface in some sections, while the rest of the road surface was gravel. Route Two is shorter than Route One by 4.8 km. The vehicle was driven in both directions, on both routes unloaded. The vehicle was then driven to another timber compartment called Zone Four, where a load of 31 450 kg of wattle timber was loaded onto the vehicle. The vehicle was then driven on both routes in both directions loaded. Consult Table 4.1 for a listing of the routes travelled in the tests. The VPMS was active during the testing and recorded vehicle performance data for all trips.

Table 4.1 Observed vehicle trip details

| Trip # | Detail | Route | Load | Start time | Trip distance (km) | Average speed (km/h) | Max speed (km/h) | End time | Trip time (min) |
|--------------------------------|------------------|-------|-------|------------|--------------------|----------------------|------------------|----------|-----------------|
| 1 | Mill to Zone 3 | Short | Empty | 08:27 | 11.8 | 36.5 | 66.6 | 08:46 | 19.6 |
| 2 | Zone 3 to Mill | Long | Empty | 08:52 | 16.6 | 42.4 | 72.4 | 09:15 | 23.5 |
| 3 | Mill to Zone 3 | Long | Empty | 09:23 | 16.6 | 46.0 | 76.6 | 09:45 | 21.7 |
| 4 | Zone 3 to Mill | Short | Empty | 09:47 | 11.8 | 36.1 | 73.3 | 10:07 | 19.9 |
| M-Z4 | Mill to Zone 4 | Short | Empty | 10:45 | 20.0 | 37.3 | 76.4 | 11:17 | 31.4 |
| Timber Loaded at Zone 4 | | | | | | | | | |
| Z4-Z3 | Zone 4 to Zone 3 | | Full | 13:19 | 8.3 | 23.4 | 77.8 | 13:40 | 21.6 |
| 5 | Zone 3 to Mill | Short | Full | 13:43 | 11.8 | 19.8 | 75.2 | 14:19 | 36.2 |
| 6 | Mill to Zone 3 | Short | Full | 14:27 | 11.8 | 25.1 | 59.1 | 14:56 | 28.6 |
| 7 | Zone 3 to Mill | Long | Full | 15:00 | 16.6 | 26.8 | 71.2 | 15:37 | 37.1 |
| 8 | Mill to Zone 3 | Long | Full | 15:42 | 16.6 | 30.1 | 79.4 | 16:15 | 33.0 |
| 9 | Zone 3 to Mill | Short | Full | 16:22 | 11.8 | 21.2 | 74.0 | 16:59 | 33.8 |

4.3 SimTrans Simulation

The SimTrans model requires the user to input three data files to describe the route and speed changes along the route and the vehicle details. The accuracy of the simulation results from SimTrans depends on the accuracy of the input data and thus careful consideration was given to the preparation of accurate data files (Clark, 1996).

The first input file required for SimTrans is the vehicle file, which is used to describe the engine, gear box, axles, the tyres and loading of the vehicle and the trailer, and other associated components of the vehicle. The vehicle specifications used for the simulation may be seen in Table H.1, page 124 in Appendix H.

The road profile was digitised by Green Belt Mapping using a Trimble Pro XRS mapping grade GPS with a sub-meter accuracy. Post processing was performed on the GPS data to enable the files to be compatible with SimTrans. The vertical route profiles are plotted in Figure 4.4 and Figure 4.5. The equivalent grade rolling resistance values, required for the road profile input file were estimated using Table 3.1, page 35.

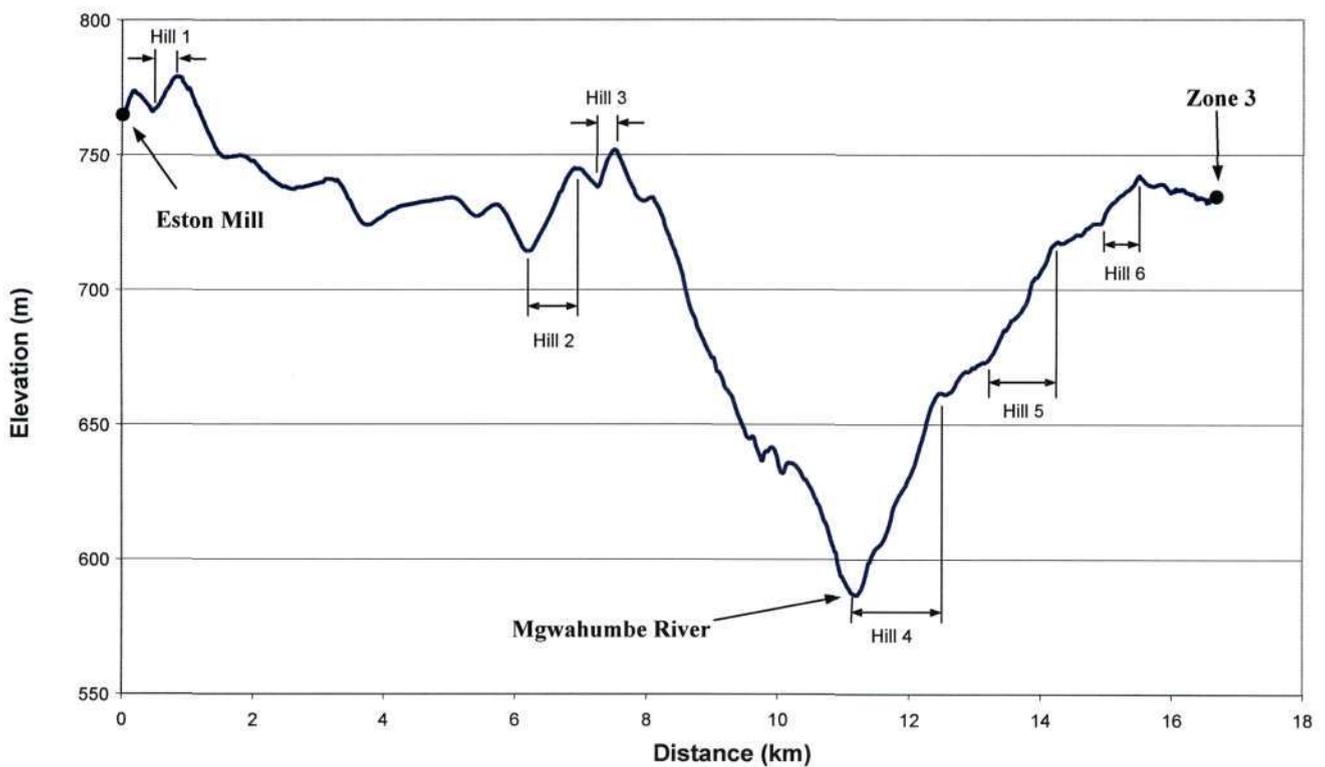


Figure 4.4 Vertical profile of Route One from the Eston Mill to Zone Three

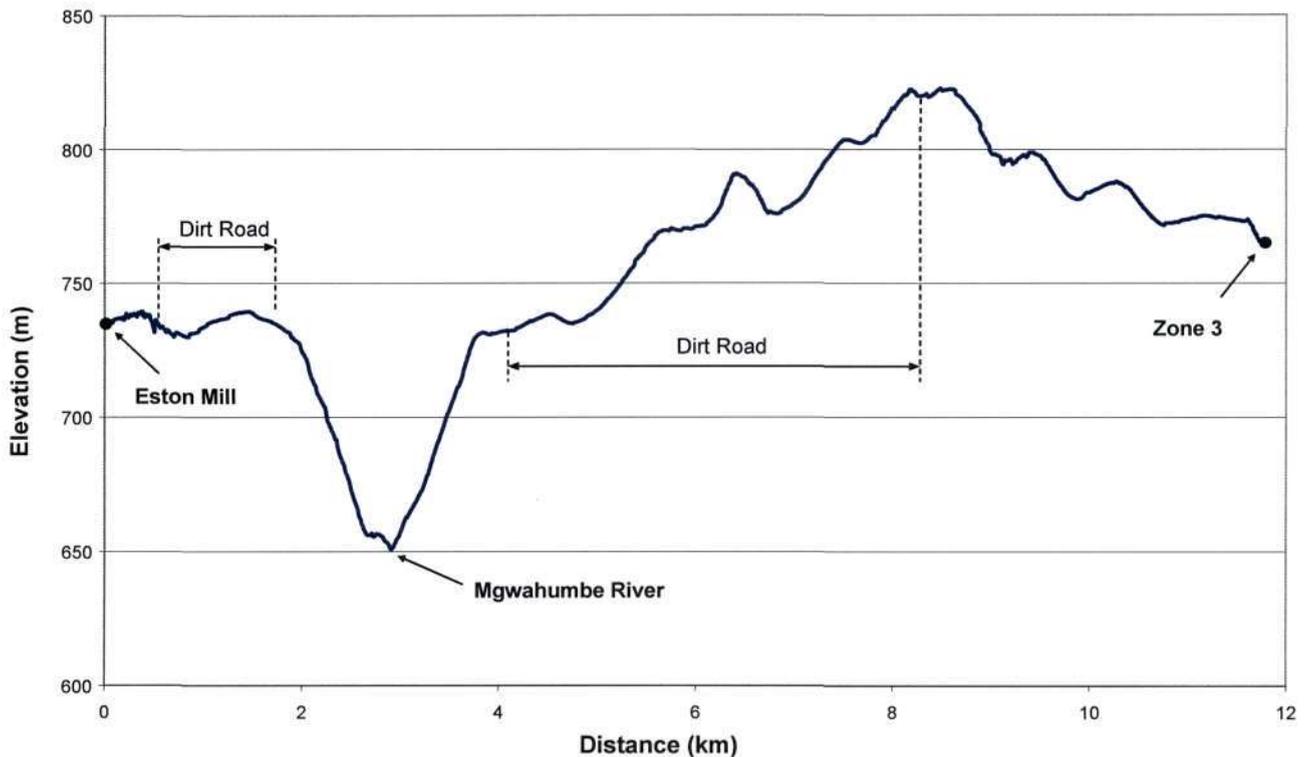


Figure 4.5 Vertical profile of Route Two from the Eston Mill to Zone Three

The speed limit file for the simulation was constructed by setting the observed vehicle’s maximum speed over road sections varying from 10 m to 50 m, as the speed limit for the simulated vehicle in these road sections. The road segments were determined using the GPS position data from the observed vehicle dataset and since the GPS position data was collected on a timed interval, the distance between samples varied. The road was split into these sections to limit the number of speed variations on the route and make data processing manageable.

Through studying the observed performance data, it was decided that the most realistic method of simulating the driver’s choice of gear changes was to run the SimTrans model in the “greenband” gear changing mode. The entire dataset of observed engine speed versus data point number of the observed engine speed values, as they appear in the observed dataset, was plotted in Figure H.3 on page 126 in Appendix H. From this plot it was determined the engine speed at which the gear up shifts occurred was approximately 1900 r/min, while down shifts occurred at approximately 1000 r/min. Thus these engine speeds were taken as the greenband upper and lower limit respectively for the SimTrans simulation. A stationary start was applied in all simulations.

5. RESULTS AND DISCUSSION

Clark (1996) recommended that further investigation on the SimTrans model was required, as there were variances between the simulated results produced by the model and measured performance data set. The gear selection routines were not well simulated in SimTrans and as such required further research (Lyne, 2002). For this research project, the approach adopted to evaluate and validate the SimTrans model was a fivefold method. Firstly the existing model was studied and is documented in Chapter Three. A data logging device was designed and installed on an actual haulage vehicle to capture an observed set of data while the vehicle was operating under normal working conditions. This haulage vehicle could then be simulated in SimTrans using the same route, speed limits and loading conditions as experienced in the observed test. The observed data set was then compared to the simulated results. By comparing these two data sets, the specific instances where the model was not producing accurate simulated results were identified. Finally, through careful examination of each of these instances where there was a deviation of the simulated results compared to the observed results, suitable modifications to the model were then devised and implemented. In order to demonstrate that the modifications would enable the model to produce more accurate simulated results, the modified model was run with the same input parameters used in the original model. The simulated data was once again compared to the measured data set.

5.1 Comparison of Simulated and Observed Data

The data sets for both the observed vehicle performance data and the simulated results from SimTrans consisted of a significant number of data points. Thus, displaying the results for each test on a single graph would be impractical. Hence it was decided to divide up the routes into several hill sections. Now, since the objectives of this dissertation aimed at identifying problems in SimTrans and then solving these problems, rather than a comprehensive verification of the model, it was decided to investigate only certain sections of the observed dataset. The down slope sections of the route were not investigated in this dissertation since the vehicle would be running under compression with torque not being a limiting factor on these sections of the road. Through some careful examination of the data, the hill sections Hill Two, Hill Three and Hill Five, as designated in Figure 4.4 on page 57, were chosen. In

these hill sections the errors in SimTrans were identified and the improvements in performance of the modified model were demonstrated.

Looking at the data, from Figure 5.1 on Hill Two, the altitude rises from 718 m to 745 m and is a 27 m climb over 580 m (4.7 % slope). In Figure 5.1 the observed vehicle's speed drops as the hill rises as the engine is unable to deliver sufficient torque to maintain the vehicle speed. Once the crest of the hill is mounted at 6.90 km however, the vehicle speed increases again as the torque demands are reduced. Looking at the simulated vehicle, initially the speed drops off at a similar rate to the observed vehicle. Between 6.67 km and 6.70 km the hill's grade flattens off slightly and the simulated vehicle's speed becomes fairly constant. In this section of the road the simulated vehicle's engine seems able to meet the torque demand. However, at 6.71 km the simulated vehicle's speed drops off rapidly and the vehicle stalls at 6.75 km. The vehicle then accelerates from 0 km/h to try and reach the speed limit.

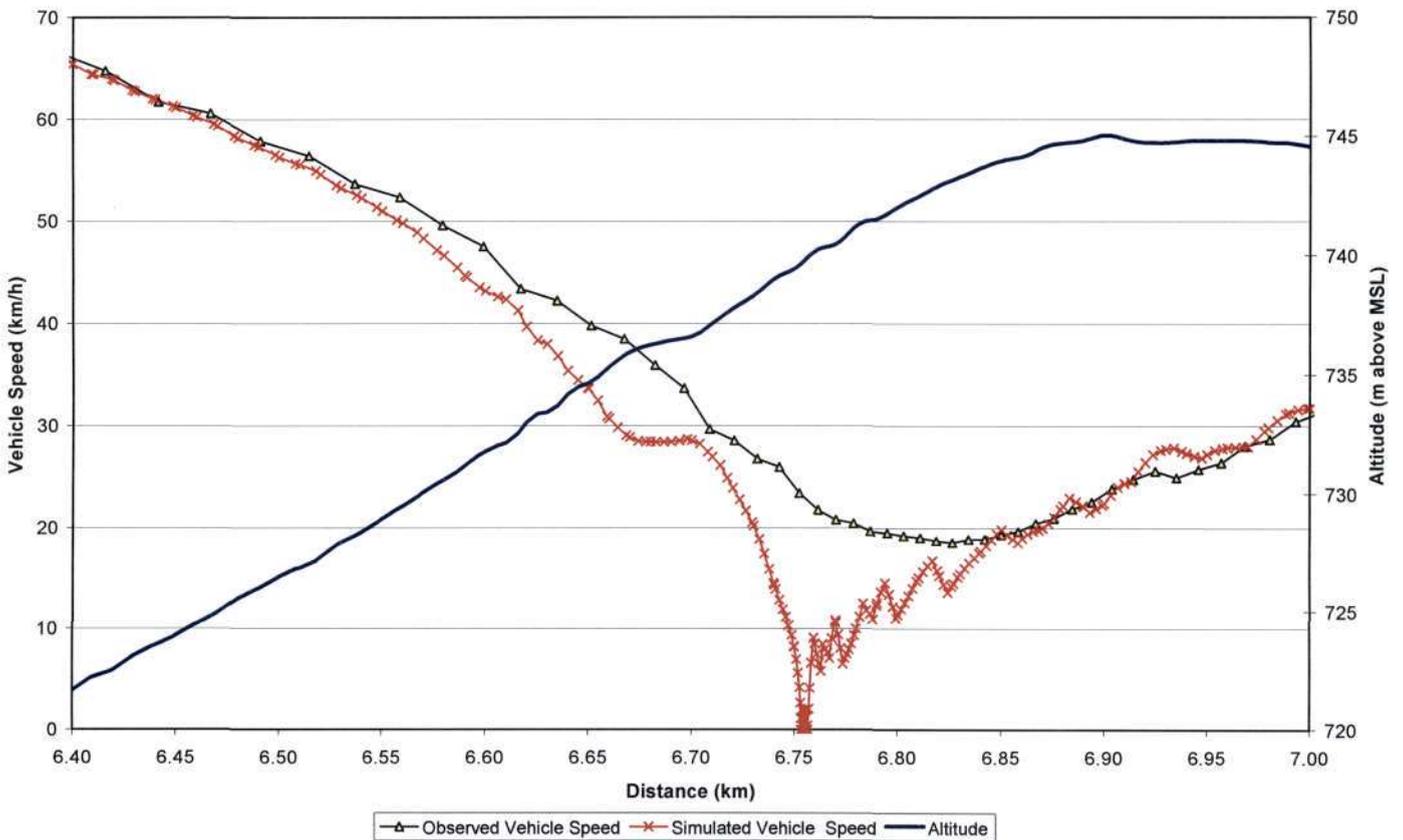


Figure 5.1 Comparison of observed and simulated vehicle speed and the altitude on Hill Two via Route One

In Figure 5.2, one can compare the sequence of gear changes of the observed and simulated vehicles. Shown in Figure 5.2 is the throttle percentage of the observed vehicle. The driver in the observed vehicle drove with full throttle for virtually the entire trip, reducing the throttle requirement only during gear changes. Now, as the vehicle speed was decreasing even with full throttle, thus the observed vehicle was operating at near peak torque. Looking at the observed vehicle's gear changes, one will observe that the driver skips down a gear from fourteenth to twelfth gear at 6.60 km and then again between twelfth and tenth gear at 6.67 km. The driver skips up a gear between ninth and eleventh gear at 6.93 km. When the vehicle is driving in ninth gear, the engine speed dips but the driver remains in the gear, maximising the torque in the gear.

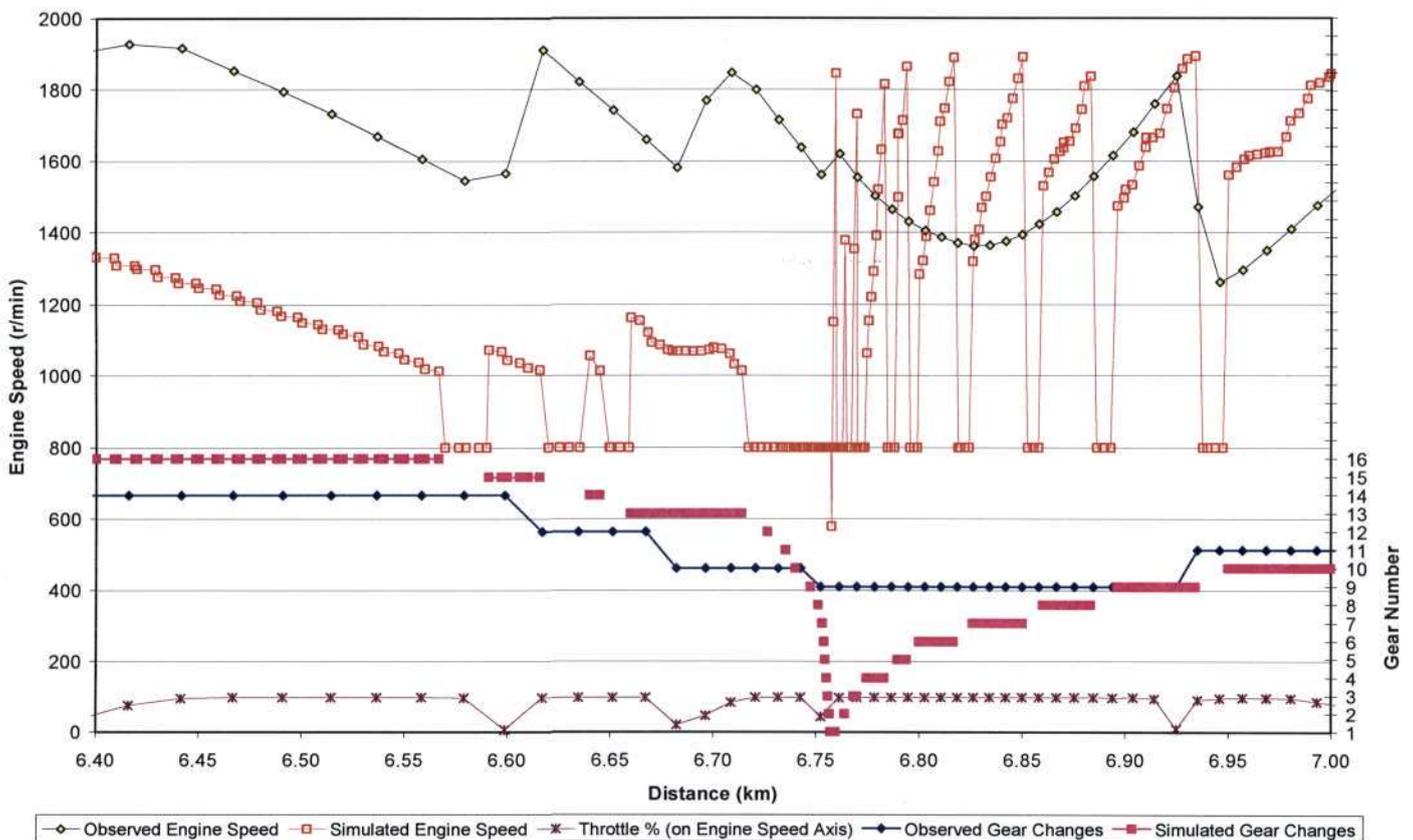


Figure 5.2 Comparison of observed and simulated engine speed and gear changes on Hill Two via Route One

Comparing the simulated vehicle's gear changes to that of the observed vehicle in Figure 5.2, it was noted that the simulated vehicle does not skip gears as this is not a feature of the SelectGear function in SimTrans. Another point to note on the simulated vehicle data is that

the time required for a gear change varies significantly in each gear change. Looking at the change down from sixteenth to fifteenth gear from 6.57 km to 6.59 km, the vehicle takes 1.8 seconds to change gear over a distance of 20 m. From fifteenth down to fourteenth gear from 6.62 km to 6.64 km, the vehicle takes 1.9 seconds to change gear over 20 m. From fourteenth down to thirteenth gear from 6.645 km to 6.66 km, there is a 1.2 second change time over 10m. However, from Table H.1, page 124 in Appendix H, a gear change time of 0.75 s is specified for the vehicle. In Figure B.9.1 on page 102, the Main procedure of SimTrans incorrectly applies a gear change time of 1.5 seconds irrespective of the value read from the vehicle file. The discussion above shows that not only is the clutch not being re-engaged within the required time, but there is a varying time to change, which is indicative of some incorrect assumption and/or program error for gear change time in SimTrans.

Considering the engine speeds of the simulated vehicle in Figure 5.2 between 6.4 km and 6.7 km, the engine speed as the clutch is engaged is never above 1200 r/min, which is near the bottom of the greenband window of 1000 r/min. Thus the simulated driver is not making efficient use of the gears in this section as the driver changes out of each gear after only driving a short distance in the gear. Frequent gear changing is inefficient as it results in increased transmission wear and higher fuel consumption. However, the observed vehicle maintains higher engine speeds and the driver skips down gears to ensure that when the clutch is engaged, the engine speed is close to the upper greenband limit. Thus the observed vehicle is able to remain in each gear longer than the simulated vehicle between 6.60 km and 6.95 km.

From at 6.71 km onwards the simulated vehicle speed drops and there is a rapid sequential progression of gear down shifts from thirteenth gear down to first gear, thereafter the vehicle stalls. The vehicle then has to shift up through each gear as it accelerates from 0 km/h to reach the speed limit. The time spent in each gear as the vehicle is accelerating is brief at approximately three to four seconds in each gear until seventh gear is reached, after which the driver is able remain in each gear for longer and thus make more efficient use of each gear. This sequentially shifting through the gear box is an inefficient process as time and vehicle speed are lost through each gear change and is also not normal practice for an experienced driver. The SimTrans routines therefore need to be refined so that they more closely represent the actual situation.

The comparison of observed and simulated vehicle speed on Hill Three on Route One is shown in Figure 5.3. The observed vehicle initially maintains a constant speed until 7.38 km, when there is a speed drop. Looking at Figure 5.4, at the point when the speed drop occurs there is a gear change from fifth up to sixth gear. The observed vehicle is operating at full throttle and starts to accelerate up the hill. There is a further gear up shift at 7.45 km from sixth to seventh gear and an accompanying speed drop. At 7.5 km there is another gear shift from seventh gear up to eighth gear. The hill has an average slope of 5.7 %.

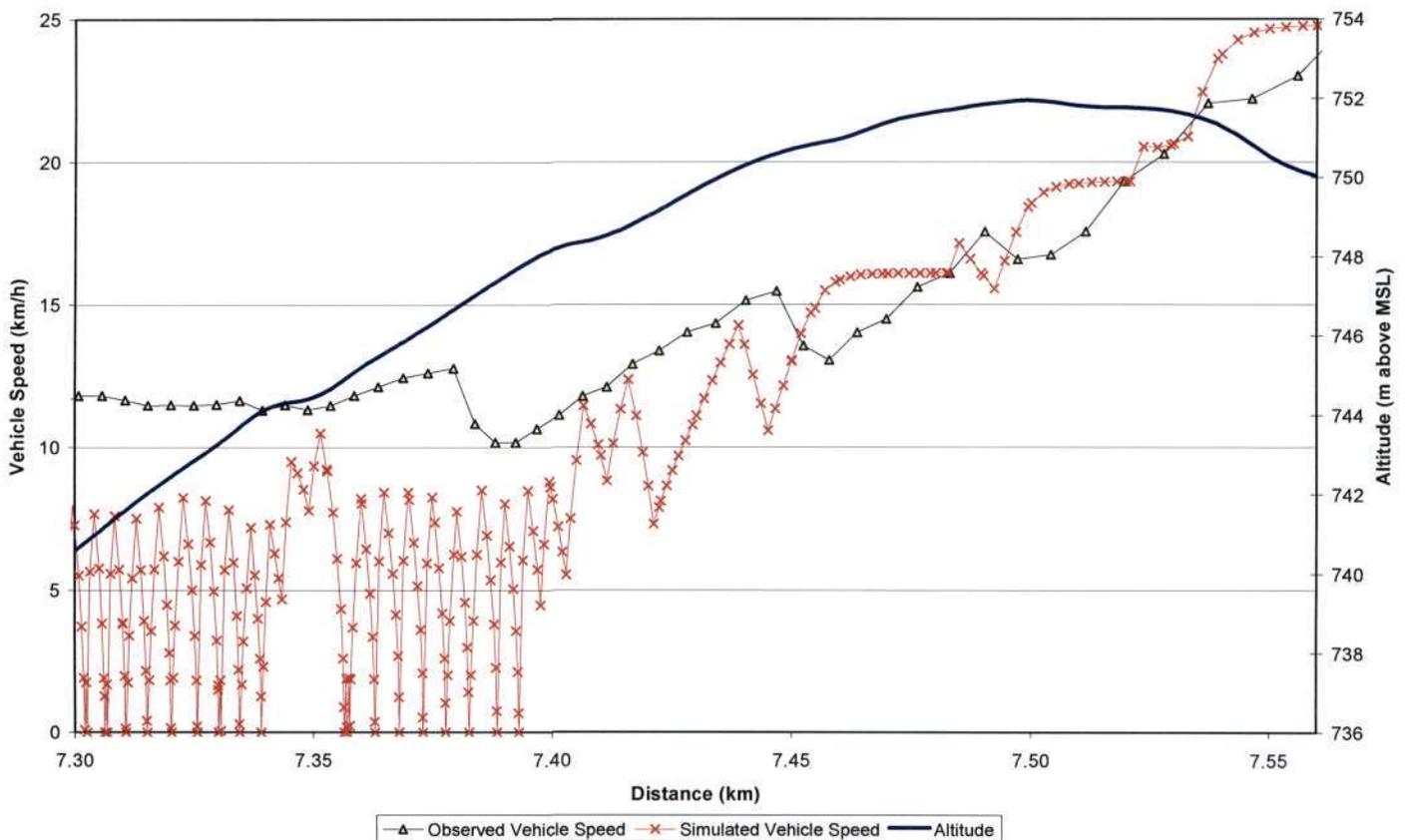


Figure 5.3 Comparison of observed and simulated vehicle speed and the altitude on Hill Three via Route One

The simulated vehicle's speed on the other hand fluctuates rapidly (every one and a half seconds there is a change in speed) while travelling over the distance 7.3 km to 7.4 km. The reason for these speed variations is explained by looking at the gear changes in Figure 5.4. The simulated vehicle shifts between first and second gear eleven times while travelling over the distance 7.30 km to 7.34 km and a further nine times while travelling over the distance 7.36 km to 7.4 km. Now, looking at one of the points when the simulated vehicle changes out

of first gear and into second gear in the above mentioned distance, by the time the clutch is re-engaged for second gear, the speed drop is such that the vehicle has lost momentum and thus the driver needs to go back to first gear instead of remaining in second gear. Should the vehicle have remained longer in first gear and then changed up a gear, all the repeated gear changing could possibly have been avoided. As the model is using 1.5 seconds a gear change time, this is the reason for the speed variations every 1.5 seconds. This abnormal simulated driver behaviour is indicative of a further programming or logic error in SimTrans. The observed vehicle on the other hand, remains in fifth gear all the while the simulated vehicle is shifting between first and second gear over the distance 7.3 km to 7.38 km.

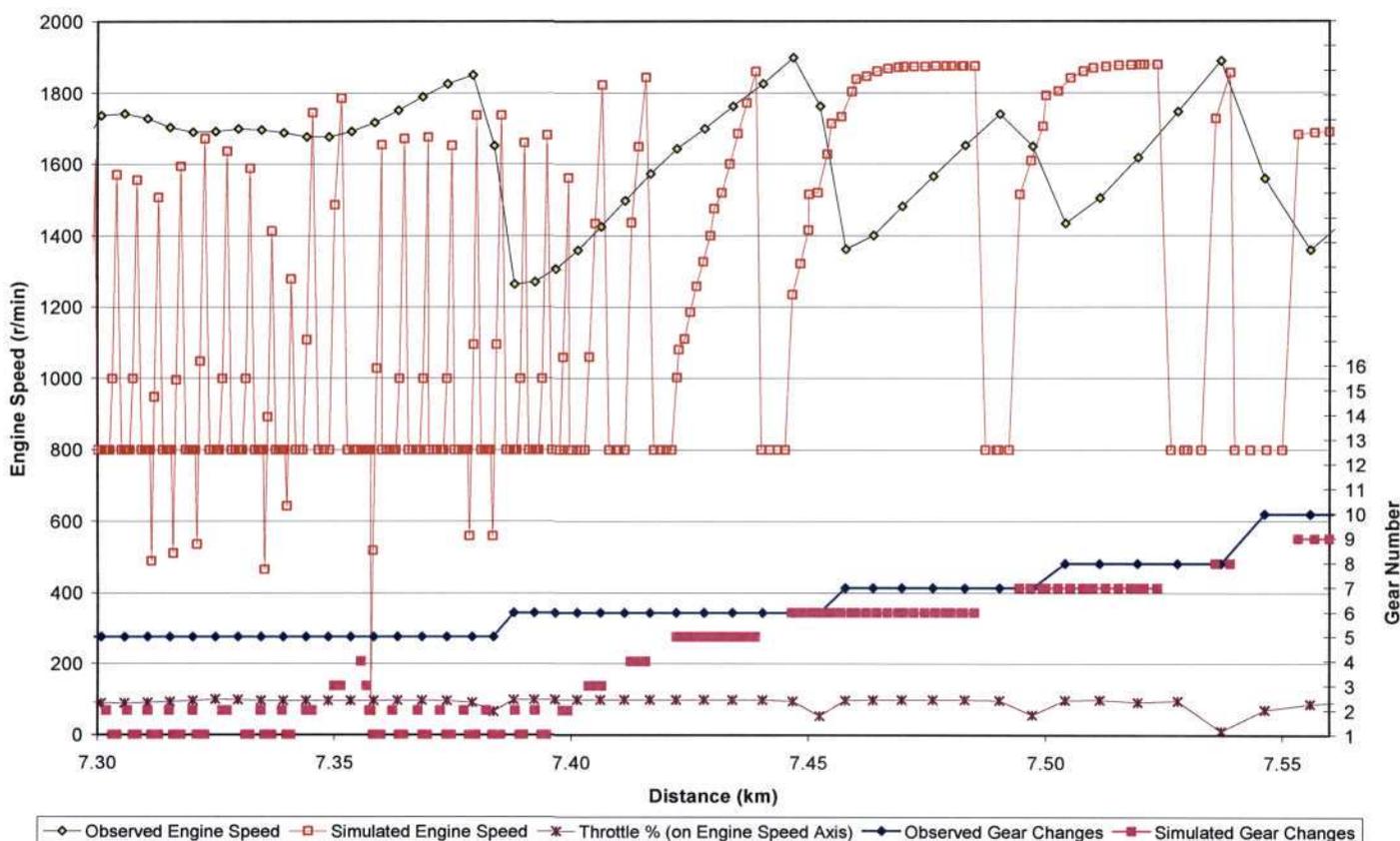


Figure 5.4 Comparison of observed and simulated engine speed and gear changes on Hill Three via Route One

As the observed vehicle reaches the crest of Hill Three, more torque is available so the driver can start moving up though the gears. It is to be noted that when the simulated vehicle passes 7.4 km, it is able to match the observed vehicle’s performance more closely as the vehicle is not shifting in and out of gears with quick succession. It is possible that over the distance

preceding 7.4 km the simulated vehicle was operating at a point where it was close to a gear change interval, but the model was unable to correctly determine the right gear for this section of the hill.

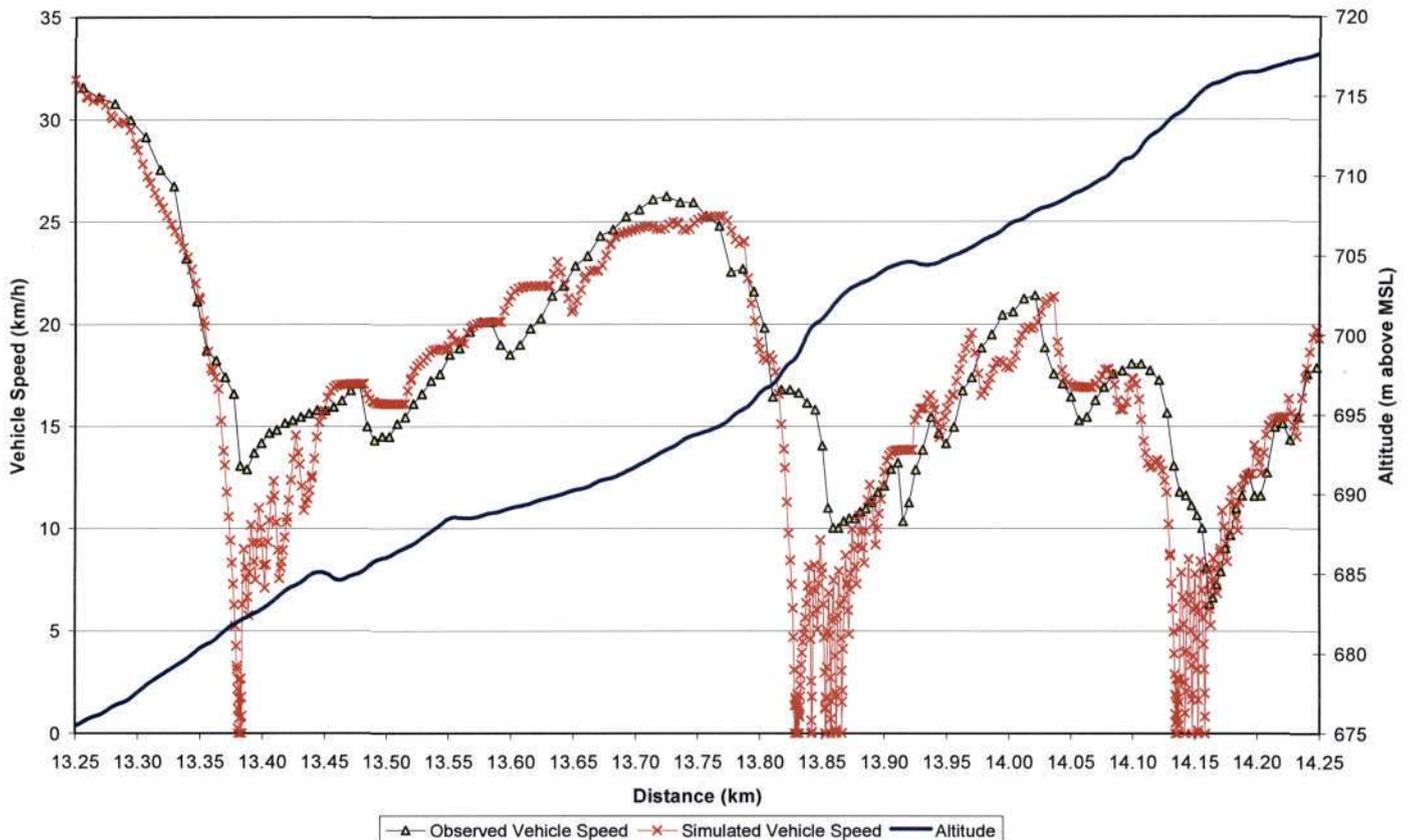


Figure 5.5 Comparison of observed and simulated vehicle speed and the altitude on Hill Five via Route One

Figure 5.5 and Figure 5.6 show observed and simulated vehicle performance data on Route One, Hill Five. The hill has an average slope of 4.2 % and the observed driver drove with full throttle over the entire distance, reducing the throttle only during gear changes. As the observed vehicle climbs the slope the vehicle speed drops from 32 km/h at 13.38 km to 13 km/h over 130 m with one gear change from eleventh to ninth gear. The driver then makes a further gear change down to seventh gear skipping down two gears at once. Over the section 13.4 km to 13.75 km the observed vehicle gains speed as the vehicle accelerates up the slope and a gear change is made from seventh up to eighth gear and from eighth up to tenth gear. The vehicle speed reaches 26 km/h at 13.72 km and then begins to drops off rapidly to 10 km/h, i.e. a drop of 16 km/h over a distance of 130 m. The slope of the hill over the distance

13.40 km to 13.75 km is an average of 9 %. At 13.77 km the observed vehicle gears down from tenth gear to eighth gear and then to seventh gear in preparation for the slope. In seventh gear the vehicle is able to maintain its speed for a distance of 34 m, but at 13.84 km, a further gear shift down is made to fifth gear as the vehicle speed is dropping at this point.

Remaining in fifth gear from 13.84 km, the observed vehicle begins to gain speed and as the vehicle reaches the crest of the steep section of Hill Five at 13.91 km, a gear up shift is made to sixth gear. The vehicle then shifts up to eighth gear and then into ninth gear at 14.04 km. There is a speed drop of 2 km/h at 14.06 km, which can be explained by considering that there is a brief reduction of the throttle to 40% at 14.05 km. The reason for this throttle reduction is unclear from the collected data and may have been a result of traffic or some road condition. At 14.12 km there is another steep section of the hill until 14.19 km, with an average slope of 5.5 %, causing the vehicle's speed to drop to 6 km/h at 14.16 km. The driver changes down from ninth gear, to seventh gear and finally to fifth gear. In fifth gear the vehicle's speed initially drops to 6 km/h at 14.16 km, but then the speed rises as the vehicle mounts the steep section of the hill at 14.19 km and the driver changes up to sixth gear and then into seventh gear.

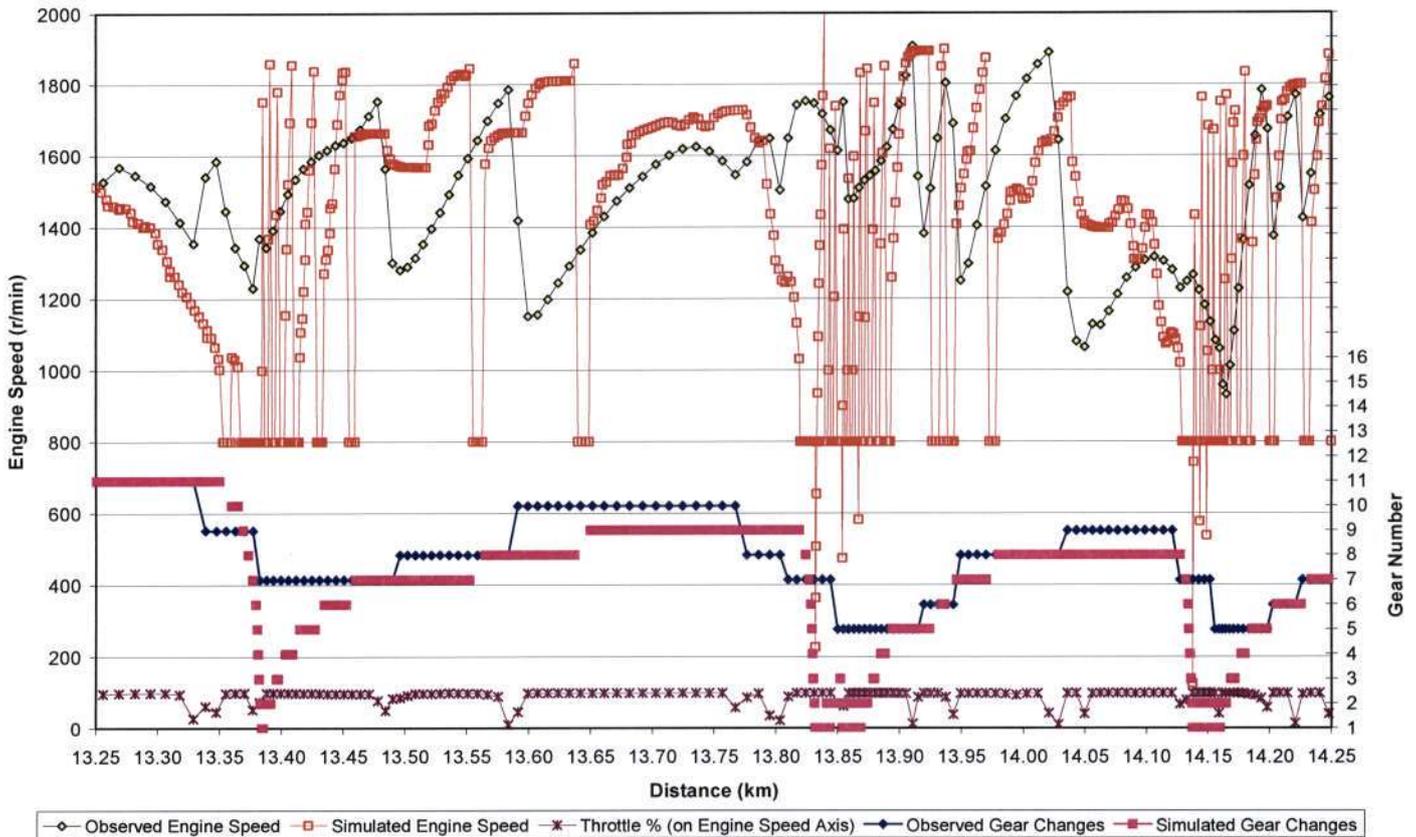


Figure 5.6 Comparison of observed and simulated engine speed and gear changes on Hill Five via Route One

The pattern of the simulated vehicle's speed is similar to that of the observed vehicle's speed on Hill Five via Route One. However, whenever the observed vehicle's speed drops sharply, i.e. at 13.35 km, at 13.81 km and at 14.13 km, the simulated vehicle's speed then drops to zero and the vehicle stalls. At 13.25 km the simulated vehicle is driving in eleventh gear, however, as the simulated vehicle climbs the slope the speed drops and a gear down shift to tenth gear is required at 13.35 km. Looking at the engine speed when tenth gear is engaged, one may observe that it is 1030 r/min, which is 30 r/min above the lower greenband limit. Consequently the driver remains in tenth gear for just over one second before another gear change is required. The simulated driver then sequentially shifts down through each gear to reach the correct required gear, but is unable to select the required gear in time and the vehicle stalls at 13.38 km. As each gear change is made in this section the driver still has a problem of being in too a high gear. This problem is further compounded since through each gear change there is a speed drop and the time lost in engaging and then disengaging the clutch means the driver has even less time available to reach the required gear. When first gear is

engaged, the vehicle accelerates to reach the speed limit, shifting through each gear. However, once the simulated vehicle reaches fifth gear at 13.42 km, it does not exhibit any irregular gear changing patterns until reaching 13.77 km.

One should note that at 13.77 km the driver of the observed vehicle seems to be anticipating the steep section of the hill and changing down gears ahead of time. However, as SimTrans does not have any predictive capability, the simulated vehicle remains in ninth gear for possibly too long as the simulated vehicle again stalls at 13.83 km. The reason for the vehicle stalling at this point is explained by the same argument as when the vehicle stalled at 13.38 km on this hill, i.e. the driver is unable to select the correct gear in time. After stalling, the vehicle attempts to accelerate up to the speed limit but appears to be unable to select the correct gear at the correct time. It changes between first and second gear seven times over the distance 13.83 km to 13.87 km, reaching third gear once at 13.85 km but then having to return to second gear. The vehicle is trying to accelerate off from rest on the steepest section of the hill which puts a high torque demand on the engine and thus could be contributing to the gear selection indecision. Again, as with the previous point when the simulated vehicle stalled, once the vehicle reaches fifth gear at 13.89 km, it is able to drive “normally” with no irregular gear changing patterns until 14.12 km. Thereafter, a similar pattern of the vehicle stalling and then unable to choose the correct gear is observed.

5.2 SimTrans Modifications

5.2.1 Main Procedure

One of the problems identified in the preceding section that negatively affected the simulated vehicles performance was the inability of the simulated vehicle (at times) to engage the clutch at the correct moment. The original Main procedure engaged the clutch based on a calculation of time elapsed since the clutch was disengaged. By studying the original Main procedure, it was observed that the time step value was being set incorrectly at several points in the procedure and consequently the procedure often missed the set moment to engage the clutch. This resulted in extended periods where the clutch was disengaged as the procedure was unable to determine the correct time to engage the clutch. The vehicle speed then dropped lower than the program anticipated and thus once the clutch was engaged, the simulated vehicle was in the incorrect gear and another gear change was required.

Thus the solution to this timing problem was to identify when the procedure was setting the time step value and ensure that these steps did not produce an error. Points in the procedure where it was found that the time step variable dt was being set incorrectly were mentioned in the following text. The time step variable was changed from being a local variable to being a global variable. Being a local variable essentially means that the value of dt was cleared each time the procedure ended and thus had to be re-assigned a value when the procedure started again. However, changing dt to a global variable means its value can now be set and would retain this value when the procedure began its next loop. The effect of this change in the variable type was that it eliminated the programming errors introduced by re-assigning a new value to the variable each time the procedure was executed.

In order to prevent the procedure from missing the time to engage the clutch, the method used by the procedure to control the clutch was altered. A new Boolean variable, *Clutch* was introduced which had the Boolean value true when the clutch was engaged and false when the clutch was disengaged. Hence by using this *Clutch* variable, the possibility of time advancing too far and gear change intervals being missed due to rounding of the time step variable or other program errors was eliminated.

The flowchart for the modified Main procedure is illustrated in Figure I.1.1, page 127 to Figure I.1.6, page 132 in Appendix I. The following changes were made to the original Main procedure. The initial step in the original procedure to set the time step variable to a value of 0.5 s and the time to change gear value to a value of 1.5 s was removed. The value for the time to change gear is now correctly read from the vehicle input file. Following the flowchart, as with the original Main procedure, there is then a check as to whether the total time elapsed is less than zero, i.e. the vehicle is at the beginning of the simulation. If true, as before the previous gear selected (which is undefined at this point in the procedure) is set to first gear and the time a gear change last occurred is set to an initial value of -5 s. However, a new line was added to set the time step variable to a value of 0.5 s, thus the value of dt is initialised at the beginning of a simulation and its value is not erroneously set back to 0.5 s each procedure execution loop as in the original procedure.

The next changes to the Main procedure were made in Section 3 of the flowchart (Figure I.1.3, page 129). The first decision statement in this section of the flowchart was altered

compared to the original procedure and is now a check on whether the clutch is disengaged (i.e. Clutch variable is false). If the clutch is disengaged then a check is made as to whether the current time step variable is greater than the time still remaining until the clutch is engaged. If true, then the time step is reduced to the time to change gear less the total elapsed time, less the total time elapsed at the moment when the clutch was disengaged. This was to ensure that the time to engage the clutch would not be missed in the next program cycle through the total elapsed time advancing beyond the time to engage the clutch.

If however, the clutch was found to be engaged, then the new Main procedure remains unchanged up until the check as to whether a gear change was made. An extra condition was added to this check to determine if the clutch is engaged (up until this point in the flowchart the clutch has not been set to be disengaged, but this forms an extra check). If a gear change did occur then a further line was added to set the Clutch to be disengaged. A further check was added in to determine if the time step is greater than the time to change gear and if so the time step is set equal to the time to change gear. This was to prevent the procedure from advancing the total time elapsed in the next procedure cycle to being greater than the sum of the current time elapsed and the time to change gear.

The procedure checks whether the currently chosen gear is the same as the previously selected gear and if the clutch was still engaged. If true, then no gear change was made and the time step is set to be equal to 0.5 s. The value of dt is often (in the course of running the simulation) adjusted to ensure that the total elapsed time correctly coincides with the gear change interval. Thus to prevent the rest of the simulation using a reduced dt value, which generates unnecessarily large data files and results in increased program execution times, dt should be set back to a value of 0.5 s. However, the point in the flowchart where dt is set back to 0.5 s is important. When the clutch is engaged and there is no associated danger of missing a gear change interval, the value of dt can then be changed (if it is not set to 0.5 s already). Next, there is a check to see if the clutch is currently disengaged and if so it is determined if the moment for the clutch to be re-engaged has arrived. If so the clutch is then engaged, the time step set to 0.5 s (see reasons in the paragraph above) and the thrust is given a nominal value of 1 N.

A further modification to the original Main procedure was added in Section 4 of the flowchart (Figure I.1.4, page 130). In the first check in this section, as to whether the thrust is not equal

to zero, an extra check whether the clutch is engaged was added (to prevent thrust values from being calculated if the clutch is still disengaged). The next change to the Main procedure was made in Section 6 of the flowchart (Figure I.1.6, page 132). After the GetFuel function is called, the procedure calculates the distance the vehicle will travel in the current time step using an equation of motion. The original Main procedure had an error in this equation of motion as there is a negative sign instead of an addition sign (see Equation 3.1).

Further changes were made to the original procedure after dt is adjusted for acceleration. If the adjusted value of dt is equal to zero, then the vehicle is already at the end of the current road segment, dt is set back to a value of 0.5 s and the new vehicle speed is determined. If however, dt is calculated greater than 0.5 s then dt is set to a value of 0.5 s, as using a larger value of dt would result in a smaller data file but the results would be less accurate. On the other hand, if dt is less than 0.5 s, then there is check if the clutch is disengaged and whether dt is greater than the time to change gear less the total time elapsed and less the time the last gear change took place. If true then dt is set equal to the time to change gear less the total time elapsed and less the time the last gear change took place. The remaining segment distance is then calculated using this new value of dt. However, if the clutch is engaged, the vehicle speed is recalculated using the acceleration and the value of the remaining segment distance is determined using an equation of motion. This equation of motion was also incorrect in the original Main procedure and is thus corrected in the modified procedure. Finally the altitude and total distance travelled are calculated.

5.2.2 AccelerationCalc Procedure

The AccelerationCalc Procedure is illustrated in Figure I.2, page 133 in Appendix I and was a new procedure in the modified SimTrans model. This procedure is adapted from the acceleration calculation algorithm in the Main procedure. This procedure was a necessary addition to SimTrans as it makes the acceleration calculation algorithm accessible to the SelectGear function and some further code was added to re-calculate the thrust. This procedure calculates the “desired acceleration”, which was the acceleration that would be required to accelerate (or decelerate) the vehicle to the reach the speed limit in the next second. An indicative acceleration (not actual) is determined by subtracting the current vehicle speed from the speed limit and this value is then converted to m/s^2 . This “desired acceleration” is then limited to either $+1.5 \text{ m/s}^2$ or -1.5 m/s^2 accordingly via the two checks,

to prevent the vehicle from accelerating at an unrealistically fast pace. The current maximum engine torque is calculated by calling the GetMaxEngTorque function. The torque losses in the transmission are then calculated by calling the GetTorqueLosses function. The current thrust available at the wheels is then determined using these calculated values.

If the surface is hard, the required thrust is determined based on the desired acceleration. If the required thrust is greater than the available thrust at the wheels, the vehicle acceleration is then recalculated using the available thrust. However, should there be more than sufficient thrust available to accelerate the vehicle; the acceleration is set equal to the desired acceleration.

5.2.3 SelectGear Function

The modified SelectGear function is shown in Figure I.3.1 to Figure I.3.3, pages 134 to 136 in Appendix I and had the same purpose as the original SelectGear function, i.e. to determine if a gear change was required and if so, what gear to select given the current vehicle ground speed and motion resistance force. However, in the modified SelectGear function, the greenband gear selection mode was altered and a gear skipping algorithm was added for both gear up shifts and gear down shifts.

The first modification to the function was to add an absolute operator to the speed drop equation. This is to ensure that when the new vehicle speed after a gear change is calculated, the speed drop value is not a speed gain, by the speed drop having a negative value. It was found that at times the program calculated that a speed gain would occur through the gear change, but before the clutch was engaged the grade changed and the estimated speed gain became a speed drop. The result was that the driver changed up or down a gear at the incorrect moment. Thus as SimTrans has no predictive capability, this was a more conservative approach to gear changing.

The next modification to the function was to add a gear skipping routine to the function. This new algorithm was devised through comparing the simulated and observed results as detailed in Chapter 5.1 and then drawing some conclusions from these results.

It was reasoned that the driver's desired acceleration would affect his choice of gear. By watching the driver of the test vehicle in action, several observations were made. If the vehicle's speed was approaching the speed limit, for instance and thus a low acceleration would be required, the driver would possibly skip up a gear instead of just selecting the next highest gear in the box. However, should the driver be accelerating rapidly, then maximum acceleration would be required from the engine and thus he may be inclined not to skip a gear.

Large haulage vehicles such as the Mercedes Benz 2637 used for the vehicle tests, were often fitted with gearboxes containing several gears and contain gear splitters so that each gear has a high range and a low range. The Mercedes Benz 2637 was fitted with a 16K-160 transmission box, which contains eight gears and eight split gears, resulting in 16 gear ratios. Thus if the driver accelerated off in first gear and then skipped into fifth, he would in fact be shifting from first gear high range to third gear high range. In order to then reach third gear low range, the driver would have to activate a switch on the gear selection stalk and depress the clutch pedal. It was observed that it was often more convenient and more efficient for the driver to change gears without switching to the gear split range as that would require more effort and result in a longer gear change time and thus reduced fuel efficiency.

In order to try and model the complex decision process of selecting the appropriate gear which occurs in the human brain, certain compromises were made. Firstly the effect of traffic and other external influences besides the route itself were ignored. SimTrans was aimed at being a transport manager's decision support system used in determining the optimum vehicle to choose based on a particular route. SimTrans was not a driver training aid and thus other external influences could be ignored for the gear selection algorithm. Another factor which was important in choosing the correct gear was evaluating the road grade in the upcoming road section. This aspect was not considered in this research and a more detailed explanation for this was given in Chapter 5.4.

Following the flowchart, the modified SelectGear function remains unchanged through the greenband driving decision, up until after the decision to shift up a gear. On the flowchart, once the function has changed up a gear for the first time during the flowchart cycle, a new algorithm was added in to make skipping up several gears possible. At this point, the new engine speed in the next highest gear is calculated based on the current vehicle speed, less the speed drop due to a gear change. The desired acceleration (which may be a deceleration) is

then determined using this new calculated engine speed in the next highest gear and the current vehicle speed, by calling the AccelerationCalc procedure. The new vehicle speed in the next highest gear than currently selected is determined using the new calculated engine speed plus the desired acceleration multiplied by 1.5 seconds.

Through analysis of the simulated data, it was found that the simulated driver would change up a gear, only to find that in the next program cycle, the gear just selected was too low and a higher gear was required. This was also true for down shifts, when just as the driver dropped down one gear, the gear ratio was too low and another downshift was required. This process was upon occasion, repeated several times. Through this process the simulated driver had to spend the time to change gear (in this case 0.75 s), one program cycle in the gear (i.e. 0.5 s) and then another 0.75 s to change up another gear. Hence 1.25 s were essentially “wasted” if the driver could have skipped out one gear change. The consequence of this sequential gear shifting was an increase in fuel consumption and increased wear and tear on the clutch and gear box, which was inefficient and costly as opposed to skipping directly to a gear a few ratios higher. Another result of this repeated sequential gear shifting was when a rapid speed drop was required, the driver could end up “falling behind” with gear changes; i.e. the driver would be continually shifting down a gear, each time finding that the incorrect gear was selected and a lower gear was required (as in discussion regarding Figures 5.2, 5.4 and 5.6). Thus if there could have a more accurate prediction of the future vehicle speed, the correct gear could be selected in the first place without having to repeatedly shift up or down gears. Hence multiplying the desired acceleration by 1.5 s provides an approximate prediction of the vehicle speed in 1.5 s time.

The decision to use 1.5 seconds as a suitable time was determined by studying the simulated data. It was decided that it would be useful to predict the vehicle speed in a few program cycles ahead. However, if one uses too large a time, the resultant predicted vehicle speed may not be descriptive of the actual conditions as there could be a sudden change of grade in the meantime. Through a trial and error process of testing the program, it was found that using a value of 1.5 s provided the best compromise of accurately predicting the vehicle speed as a result of acceleration.

Next the function calculates the maximum engine torque, the transmission losses and thus the new expected engine torque value. There is then a check as to whether the newly calculated

vehicle speed is greater than the maximum vehicle speed in the current selected gear (calculated using the upper greenband limit). There is also the condition that top gear is not selected and that the new maximum engine torque is greater than the resistance force converted into a torque (using the next highest gear ratio and the lever arm value). If this condition is true then another gear up shift is made and the previous gear is thus “skipped”. The algorithm then loops back to the top of the gear skip routine to determine whether it is possible to jump up to another higher gear. This process is repeated until the highest suitable gear is selected, or top gear is reached.

For shifting down gears a similar approach was adopted. The modifications were made to the SelectGear function in the flowchart Section 3 (Figure I.3.3, page 136), once the function has dropped down a gear for the first time during the flowchart cycle. A new algorithm was added in at this point to make skipping down several gears possible. The new engine speed in the next lowest gear is calculated based on the current vehicle speed, less the speed drop due to a gear change. The desired acceleration (which may be a deceleration) is then determined using this new calculated engine speed in the next lowest gear, by calling the AccelerationCalc procedure. The new vehicle speed in the next lowest gear than currently selected is determined using the new calculated engine speed plus the desired acceleration factored by 1.5 s. Multiplying the desired acceleration by 1.5 s, as described above for skipping up gears, provides an approximate prediction of the vehicle speed in 1.5 s time.

There is then a check as to whether the newly calculated vehicle speed is less than the minimum vehicle speed in the current selected gear (calculated using the lower greenband limit). There is also the condition that the newly calculated engine speed is less than the rated speed and that first gear is not selected. If these conditions are true then another gear down shift is made and the previous gear is thus “skipped” over. The algorithm then loops back to the top of the gear down skip routine and determines whether it is possible to drop down another gear. This process is repeated until the lowest suitable gear is selected or first gear is reached.

5.3 Validation of Modified Model Accuracy

In order to validate the modifications to the SimTrans model detailed in Chapter 5.2, the model was re-run with the same input parameters (vehicle, route and speed limit files) as

those used for the simulation in Chapter 5.1 and a new simulated data set was generated. The observed data and new simulated results were compared and the discussion of findings is presented in this section.

Comparing the simulated vehicle speed results in Figure 5.7 to the results in Figure 5.1, page 60, one will observe that in Figure 5.7 the simulated vehicle does not stall and is able to match the speed of observed vehicle more closely. Now in Figure 5.7 at 6.90 km the simulated vehicle begins to accelerate faster than the observed vehicle and its speed increases higher than that of the observed vehicle. However, due to the imposed speed limit of the observed vehicle's speed, at 6.93 km the simulated vehicle reduces its acceleration rate and achieves a similar speed to that of the observed vehicle. Another point to note is that in Figure 5.7 at 6.76 km and 6.78 km there are speed dips occurring at approximately at the same section of the road where the simulated vehicle stalled in Figure 5.1.

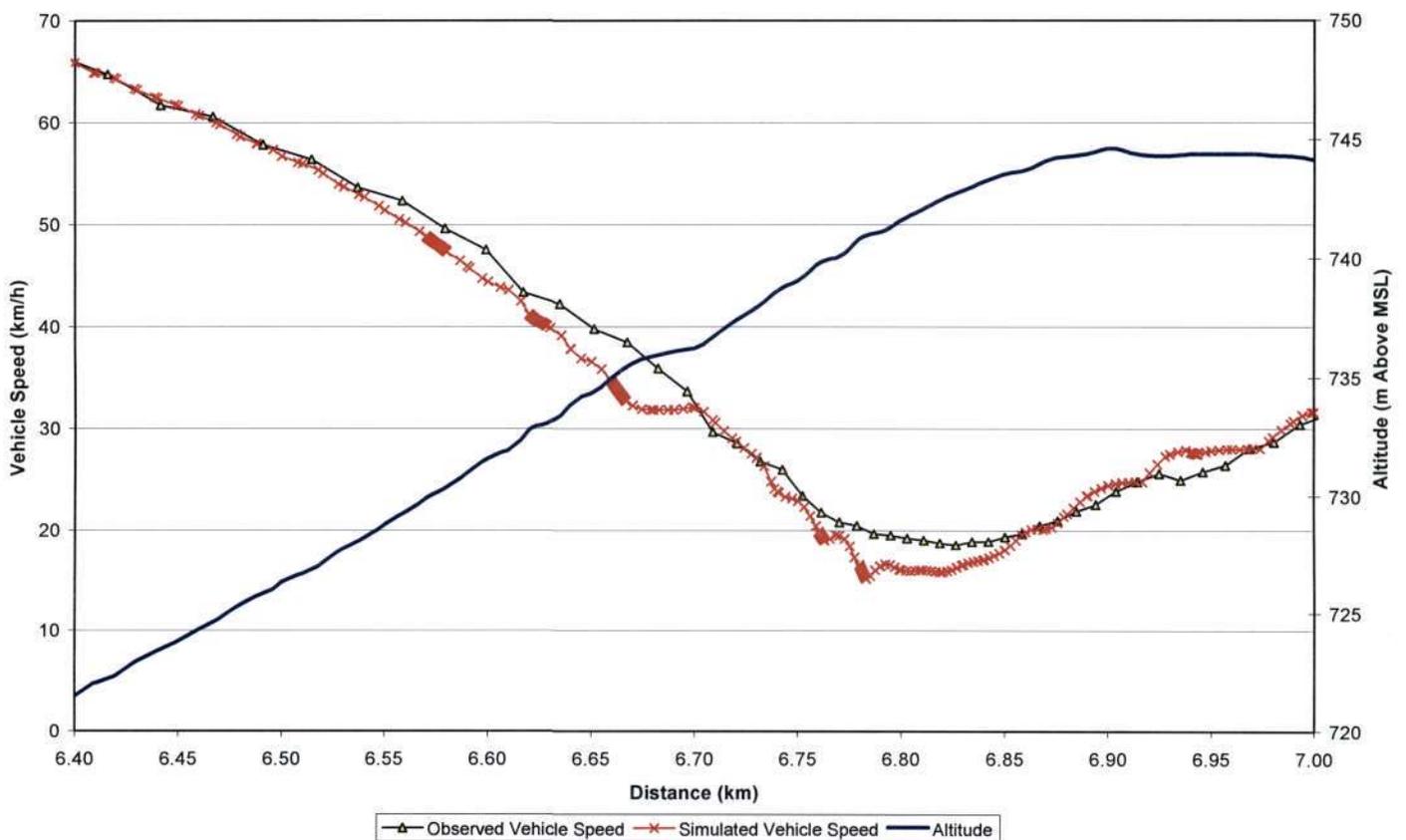


Figure 5.7 Comparison of observed and simulated vehicle speed and the altitude on Hill Two via Route One using the modified model

Shown in Figure 5.8 are the observed and simulated engine speed and gear changes on Hill Two via Route One. Initially the simulated vehicle starts out in sixteenth gear, one gear higher than the observed vehicle at the base of the hill. At 6.57 km the simulated vehicle changes down sequentially from fifteenth gear, to fourteen and then thirteenth gear, remaining in fifteenth and fourteenth gear for a distance of approximately 30 m each. The driver changes into thirteenth gear at 6.67 km as the vehicle reaches a flat section of the hill and one can see that the engine speed remains constant over this flat section of the hill, up till 6.7 km. As the gradient begins to increase, the engine speed drops and a further gear change is made, this time the driver skipping down an extra gear to eleventh gear. Two more gear changes are made to tenth and finally down to ninth gear just before the crest of the hill is reached. The speed dips at 6.76 km and 6.78 km, mentioned above, occur at the as these two gear changes are being made and thus can be explained as speed drops through the gear changes. When the simulated vehicle is travelling in ninth gear, it is in the same gear as the observed vehicle. On the top of the hill at 6.95 km, the simulated vehicle changes up to tenth gear, but the observed vehicle skips a gear from ninth gear up to eleventh gear with a consequentially larger engine speed drop. There is no associated speed drop for the simulated vehicle through this gear change and although the observed vehicle's speed does drop, it by less than 1 km/h.

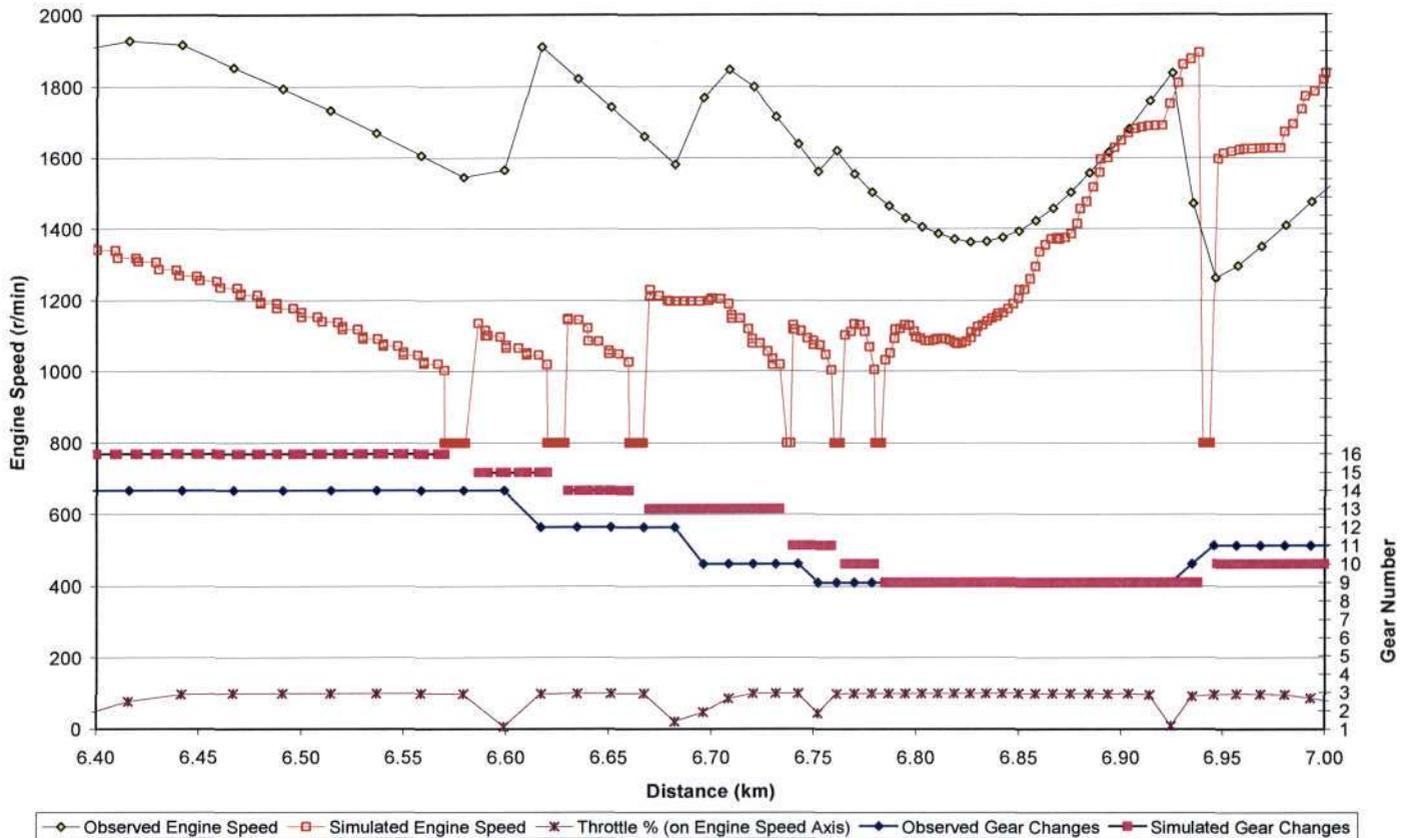


Figure 5.8 Comparison of observed and simulated engine speed and gear changes on Hill Two via Route One using the modified model

The simulated vehicle in Figure 5.2, page 61, made 24 gear changes on hill two, whereas the simulated vehicle in Figure 5.8 made seven gear changes and the observed vehicle made only four gear changes. The observed driver skipped three gears over the distance, whereas the simulated vehicle in Figure 5.8 only skipped one gear. The reason why the simulated vehicle only skipped over one gear could possibly be due to the observed driver being able to see the entire hill and predict well in advance the need to skip down an extra gear in each change while driving up the hill. The observed driver is thus changing down gears before the engine speed reaches the lower greenband limit. SimTrans on the other hand is only able to anticipate vehicle speed 1.5 s ahead when calculating a suitable gear selection, without taking in to account grade variations. However, the modified model shows significant improvement in gear changing ability over the previous version of SimTrans.

Figure 5.9 shows a comparison of the observed and simulated vehicle speed on Hill Three via Route One. For the first half of the hill, i.e. up to 7.43 km, the simulated vehicle has a higher

speed than the observed vehicle, as a result of using the maximum observed vehicle's speed as a speed limit over these distances. The simulated vehicle then has a brief speed drop of 4.8 km/h due to a gear change and thereafter accelerates, but with another speed drop at 7.49 km. The simulated vehicle seems to have bursts of acceleration and then maintains a constant speed. For instance at 7.35 km to 7.36 km the simulated vehicle speed increases 0.9 km/h and then up to 7.4 km the speed only increases 0.2 km/h to 12.8 km/h, after which the process is repeated. The speed limit for the simulated vehicle was taken as the observed vehicle's maximum speed over road sections of 10 m. SimTrans attempts to constantly maintain, but not exceed the speed limit and thus the simulated vehicle will accelerate (if possible) once it reaches a road section with an increased speed limit. Hence when the simulated vehicle seems to have bursts of acceleration, for instance at 7.35 km, 7.4 km and 7.49 km, when it is entering a new road section with an increased speed limit and thus accelerates up to the speed limit.

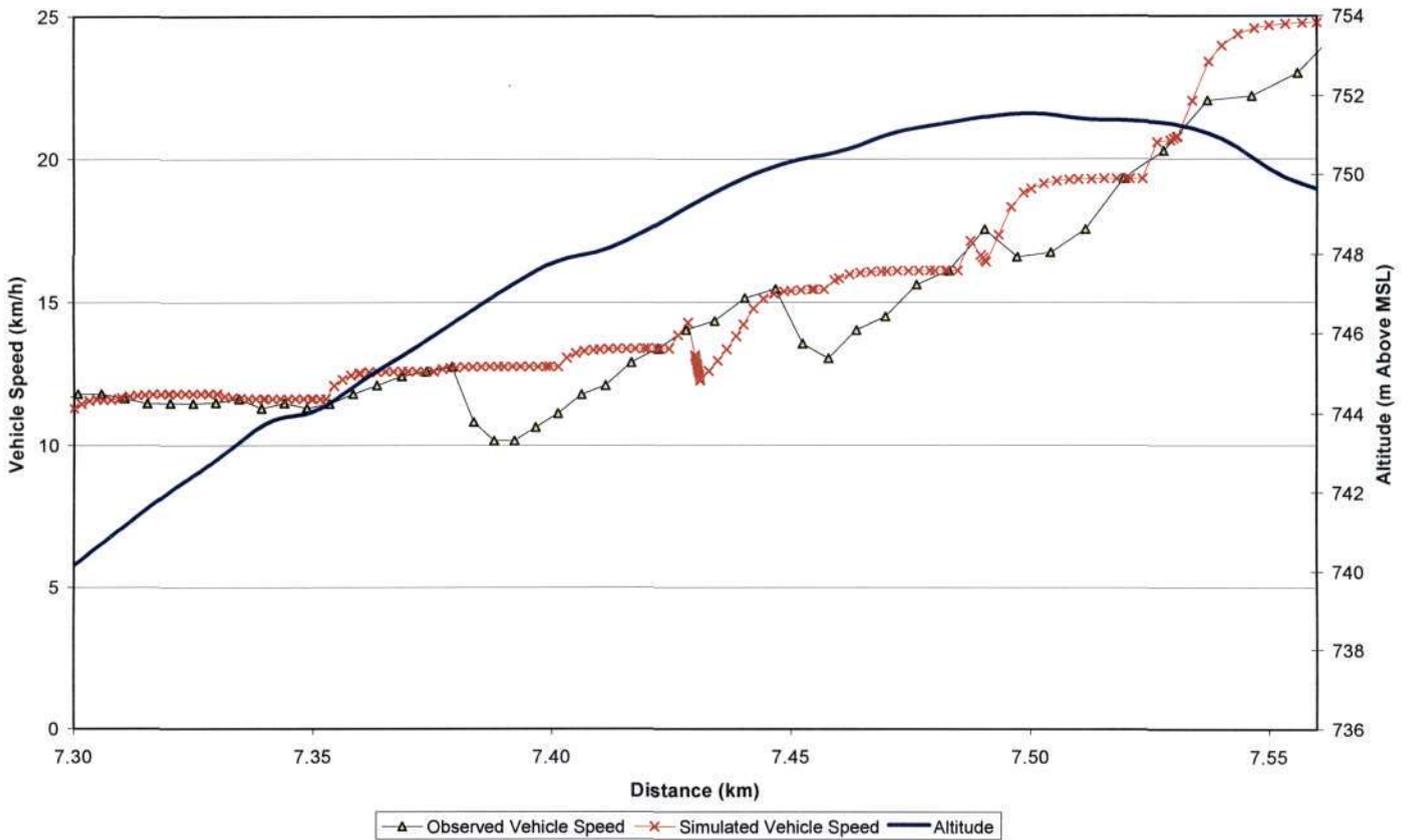


Figure 5.9 Comparison of observed and simulated vehicle speed and altitude on Hill Three via Route One using the modified model

Shown in Figure 5.10 is the comparison of observed and simulated engine speed and gear changes on Hill Three via Route One. Initially the simulated vehicle is driving in fifth gear and changes into sixth gear at 7.43 km. The observed vehicle changes out of fifth into sixth at 7.38 km, 50 m before the simulated vehicle. The simulated vehicle follows the observed vehicle's next change from sixth gear into seventh, but drove in sixth and seventh gear a distance of 30 m longer than the observed vehicle travelled in each of these respective gears. From seventh gear the simulated vehicle skips a gear to ninth gear, however, the observed vehicle shifts into eighth gear and then skips a gear into tenth gear. Looking at the engine speed, the observed vehicle reaches the upper greenband limit sooner than the simulated vehicle. This could be because the observed driver is anticipating the hill and thus using a higher acceleration than the simulated vehicle.

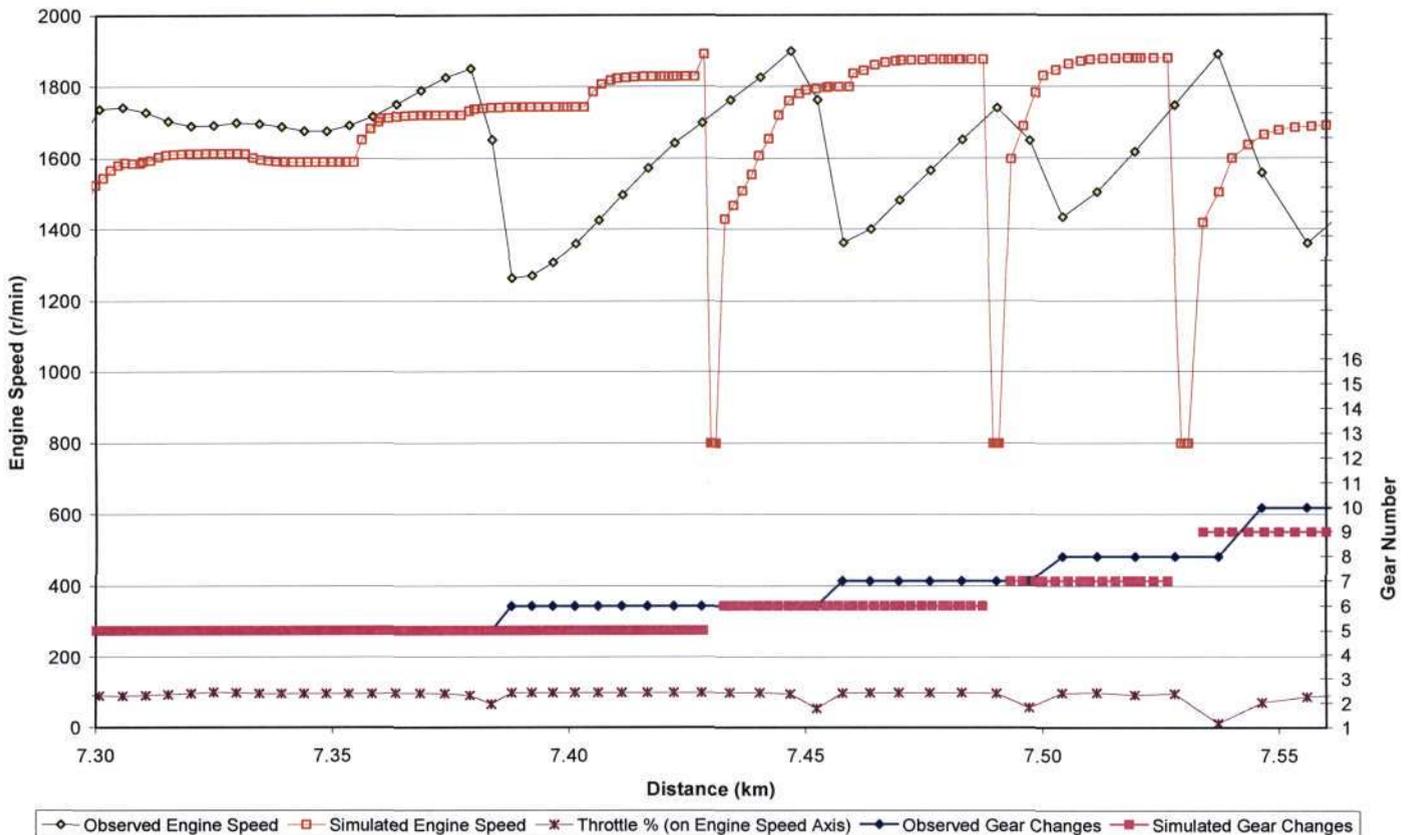


Figure 5.10 Comparison of observed and simulated engine speed and gear changes on Hill Three via Route One using the modified model

In Figure 5.10 on Hill Three, the observed vehicle makes four gear changes while the simulated vehicle makes only three gear changes. However, the simulated vehicle in Figure 5.4, page 64, changes gear 54 times on the same hill section. Comparing Figure 5.3, page 63 and Figure 5.9, one will observe that the problem of the simulated vehicle's speed variations occurring up to 7.45 km, as a result of the rapid gear changes in Figure 5.3, is solved by the modified model and it is able to simulate the observed vehicle's speed more accurately over this hill section. Thus this comparison shows that the modifications on the model represent a significant improvement over the previous version of SimTrans, insofar as gear changing routines.

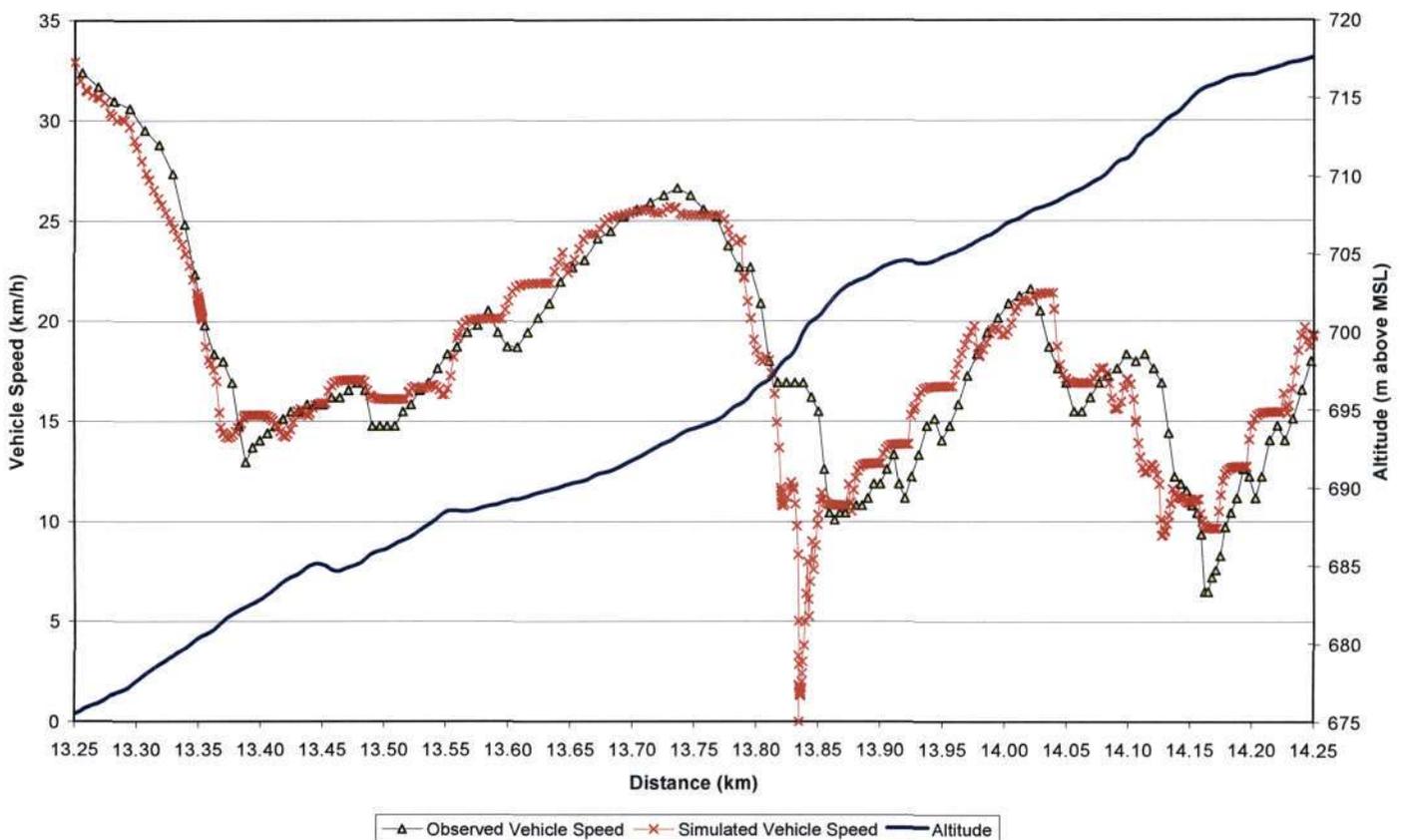


Figure 5.11 Comparison of observed and simulated vehicle speed and altitude on Hill Five via Route One using the modified model

Figure 5.11 presents the observed and simulated vehicle speed and altitude change on Hill Five via Route One and Figure 5.12 presents the comparison of observed and simulated engine speed and gear changes also on Hill Five. Figure 5.11 and Figure 5.12 will be studied in conjunction for the discussion of the simulated vehicle's performance on Hill Five. The

simulated driver is initially driving in eleventh gear and then shifts into tenth gear at 13.35 km as the vehicle speed is dropping rapidly due to the steep grade. Comparing Figure 5.12 to Figure 5.6, page 67, one will note that the simulated vehicle in both figures changes out of eleventh gear at the same point. However, in Figure 5.12 the simulated vehicle engages tenth gear a second before the vehicle in Figure 5.6, thus proving that the modified model has an improved clutch control. After changing out of tenth gear at 13.36 km, the simulated vehicle in Figure 5.12 skips down to eighth gear and remains in this gear until 13.64 km. In Figure 5.6 however, the simulated vehicle was unable to reach an appropriate gear in time and the vehicle consequentially stalled at 13.38 km after shifting down through each gear in the box. The observed driver on the other hand, was able to see the steep grade section in advance and shifted gears before the engine speed reached the lower greenband limit, thus changing from eleventh gear down to ninth gear 20 m before the simulated vehicle, in both Figure 5.6 and Figure 5.12. In Figure 5.11 looking at the road section up to 13.81 km, one will note that as the simulated vehicle ascends the hill, its speed follows a similar profile to the speed of the observed vehicle and it does not stall. Thus the modified model was able to simulate the observed vehicle more precisely for this section of the hill than the previous version of SimTrans.

Now, in Figure 5.11 at 13.81 km the simulated vehicle's speed drops off rapidly from 18 km/h to 0 km/h in 25 m. The vehicle is stalled at 13.83 km and in order to understand why the vehicle stalled, one needs to look at the engine speed and gear changes in the section 13.80 km to 13.83 km. At 13.81 km the engine speed has dropped to the lower greenband limit of 1000 r/min, so at 13.82 km the simulated driver changes down from ninth to sixth gear. Although the engine then begins to gain speed, the grade of the hill at this point is steep with a 10% slope and the engine speed drops off once more. At 13.83 km the driver has to change down gears again and skips right the way down to second gear. However, the vehicle remains in second gear for only half a second as the engine speed is at the lower greenband limit when the clutch is engaged. Consequently first gear is then selected and the vehicle's speed drops to 0 km/h through the gear change.

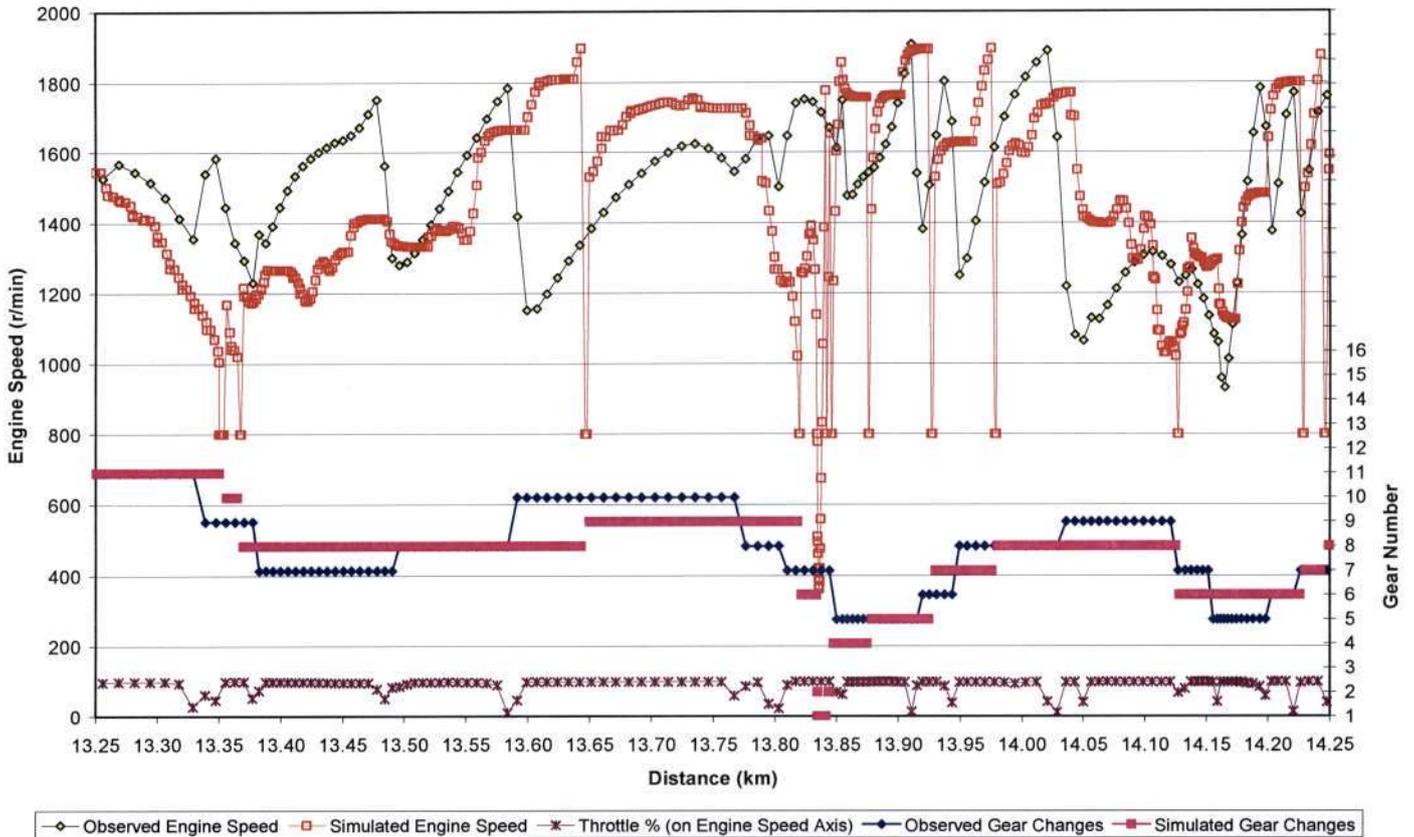


Figure 5.12 Comparison of observed and simulated engine speed and gear changes on Hill Five via Route One using the modified model

Once again the observed driver saw in advance the approaching steep section of the hill and changed out of tenth gear into eighth gear at 13.77 km, as the vehicle speed was beginning to drop due to the magnitude of the grade. The observed driver then shifted down through seventh and fifth gear, each change occurring before the engine speed had dropped to the lower greenband limit. Thus there was a smooth transition of power on the section of the hill 13.77 km to 13.90 km for the observed vehicle. More importantly, the observed vehicle ascended the hill without stalling as the driver saw in advance the steep grade and made the necessary gear changes ahead of time. The hill grade in the section after 13.81 km changes and becomes steeper rapidly. The simulated vehicle, despite being able to skip down gears, stalled because it was unable to maintain a high enough engine speed as the gear changes were made on the steep section of the hill. Instead, the gear changes should have occurred before the steep grade, where the engine could have gained speed.

In contrast, when the simulated vehicle stalls in Figure 5.5, page 65, it first dropped down through each gear, causing the vehicle's speed to drop rapidly. Once stalled, the vehicle shifted between first and second gear eleven times and the vehicle's speed varied rapidly between 8 km/h and 0 km/h for a distance of 30 m.

In Figure 5.12, after the simulated vehicle stalled at 13.83 km, it accelerates to reach the observed vehicle's speed. The driver moves up through the gears skipping over several, until eighth gear is reached at 13.98 km. For the next 0.27 km the simulated vehicle again maintains a similar speed pattern to that of the observed vehicle. At 14.1 km, the simulated vehicle's speed begins to drop 20 m before the observed vehicle's speed drops. The simulated vehicle remains in eighth gear to 14.13 km, where it then drops down to sixth gear and accelerates to match the observed vehicle's speed. From 14.15 km it maintains a higher speed than the observed vehicle for the remainder of the hill. However, at 14.13 km the simulated vehicle in Figure 5.6 shifts down through each gear in the box, stalls at 14.14 km and then proceeds to progress through a series of shifting in and out of first and second gear. Thus by the simulated vehicle skipping down a gear from eighth to sixth gear in Figure 5.12, the modified model prevents the vehicle from stalling.

If one compares the simulated vehicle performance in Figure 5.6 to Figure 5.12. In Figure 5.6 the simulated vehicle makes 66 gear changes, while in Figure 5.12 the simulated vehicle makes fourteen gear changes. The observed vehicle also makes fourteen gear changes. Although the simulated vehicle stalled once in Figure 5.12, the vehicle was able to accelerate back up to speed without rapid speed variations and poor gear selection behaviour as observed for the simulated vehicle in Figure 5.6. Thus the modifications to the model were able to successfully address the problem of the simulated vehicle stalling and prevented the rapid speed variations when the vehicle stalled.

In summary, studying the figures presented in this chapter one can conclude that the modifications to the model resulted in a significant improvement in the simulation over the previous version of SimTrans. The gear skipping routine enabled the simulated vehicle to model the observed vehicle's gear changing behaviour more closely and the simulated vehicle performed more realistically as a result. The vehicle speed of the simulated vehicle followed the observed vehicle's speed with greater accuracy in the modified model. Although the vehicle stalled once in the simulation using the modified model, it was able to accelerate up to

speed exhibiting no irregular gear selection patterns. On the other hand the simulated vehicle using the original version of SimTrans stalled and experienced gear selection problems on several occasions. Enhancements of the clutch engagement method by the addition of the Clutch variable, and other programme corrections in the modified model improved the simulation accuracy of the model, as witnessed by the comparison of figures generated from the previous and the modified version of SimTrans. It was evident however, that the modified version of SimTrans requires further research in terms of anticipating upcoming road conditions such as steep slopes and taking the necessary action ahead of time. The simulated driver would then be able to change gears before the greenband limit is reached.

Table 5.1 shows a comparison between the performance of the observed vehicle, the simulated vehicle with the original model and the modified model on Route One. The trip time and average speed of the simulated vehicle with the modified model is similar to that of the observed vehicle. However, the simulated vehicle with the original model took 15 % longer than the observed vehicle to complete the route and it has a lower average speed than the observed vehicle. Thus one can conclude that this summary data further demonstrates that the modified model was able to simulate the observed vehicle more accurately than the original model.

Table 5.1 Summary data for the simulated vehicle

| Time (s) | Distance (km) | Average Speed (km/h) | Max Speed (km/h) | kW Limit (% of time) |
|-------------------------------|--------------------------|---------------------------------|-----------------------------|---------------------------------|
| Trip Eight – Observed vehicle | | | | |
| 1980 | 16.68 | 30.1 | 79.4 | - |
| Trip Eight – Original model | | | | |
| 2344 | 16.68 | 25.6 | 79.4 | 32 |
| Trip Eight – Modified model | | | | |
| 1938 | 16.68 | 31 | 79.4 | 35 |

5.4 Further Proposed Solutions

Although the modifications to the SimTrans model did improve the overall model performance, several areas were identified where the modified SimTrans model could be further improved. The most notable addition to improve the accuracy of simulations would

be to add a predictive module to the model. This module could look at the grade and speed limit for the oncoming road section and use this data to help determine in advance the appropriate gear to select. If for instance the vehicle was approaching a steep upward slope, the simulated driver would drop down a gear or two before the vehicle reached the steeper grade section, to maintain engine speed. Alternatively if the vehicle was descending a steep slope with the engine running under compression and a steep upward slope was approaching on the other side of the valley, the driver would anticipate this and remain in the currently selected gear in preparation for the upcoming steep slope.

For example in Figure 5.11, page 81, at 13.77 km the observed vehicle started changing down gears in preparation for a steep section of the hill. Although the observed vehicle's speed dropped, the engine speed was maintained and the vehicle did not stall. However, the simulated vehicle only changed gears 50 m after the observed vehicle, when the engine speed was already at the lower greenband limit. The simulated vehicle then stalled. Currently SimTrans will only change down a gear once the engine speed drops below the lower greenband limit. On this slope, allowing the engine speed to drop to the lower greenband limit meant the gear changes occurred too late and resulted in the vehicle stalling. Had the simulated driver been able to anticipate the road grade approximately 40 m to 60 m ahead, the driver would probably have changed down gears as the observed driver shifted down gears and thus the vehicle would not have stalled.

However, successfully implementing a predictive module in the model would require further research and verification. One would need to determine a realistic and appropriate distance to look ahead and work would also be required to correctly integrate this module in the model.

Another important area for future research would be a comprehensive verification of the model through evaluating different vehicles travelling over other routes, to ensure that the model can simulate a range of different vehicle and route conditions. It would also need to be determined through this study whether SimTrans is robust enough to manage different vehicle configurations. Other useful items to add more functionality to the model would be to include modules such as an operating cost estimate, maintenance scheduling and side-by-side vehicle comparisons, similar to TransSolve.

6. CONCLUSIONS AND RECOMMENDATIONS

As transport forms a significant portion of the production costs in an agricultural system, vehicle and route selection is a critical aspect and must be optimised with due care. In order to achieve an optimal solution, a computer simulation such as SimTrans can be used to assist transport managers and advisors. However, before a model can be employed for its designed intention, the model must be proven to be sound and produce accurate results. The SimTrans model was used in this research project as it can provide the user with a comprehensive output of simulated results. This project aimed at validating the SimTrans model.

An initial investigation focused on diesel engine management systems for heavy-duty commercial vehicles and how these systems affect a vehicle's performance. The purpose of this part of the study was to gain an understanding of how performance data could be gathered directly from the onboard vehicle electronics. Specifics of engine management systems including types of sensors and actuators were investigated in detail. The Mercedes Benz 2637 obtained for the performance tests was however, not equipped with a suitable electronic management system to allow vehicle performance data to be captured from the management system. Thus a data logging system was developed to record vehicle performance data. The logging system took advantage of some of the onboard vehicle electronics in the form of the tachograph, but sensors were developed to collect other data. The logging system met its design objectives as it successfully recorded vehicle performance data and provided a useful observed vehicle data set.

The Mercedes Benz 2637 was simulated in SimTrans using the identical observed operating conditions. By comparing the observed and simulated datasets, errors in the SimTrans model were identified. With this knowledge, solutions were devised and SimTrans was effectively modified. Several errors in the model were corrected and a new gear skip module was also added to the program to enable the model to skip up or down several gears within one gear change operation. The modified model was run using the same input data as before and through comparing the observed data to the new simulated data, the modifications to the model were validated.

The updated model was able to successfully address the problems of the simulated vehicle stalling and prevented the rapid up and down gear changes observed in the previous version of SimTrans when the simulated vehicle stalled or drove at low speed. The gear skipping routine enables the simulated vehicle to perform more realistically compared with the observed vehicle. The vehicle speed of the simulated vehicle followed the observed vehicle's speed with greater accuracy in the modified model.

An area identified for further research is a predictive module which could anticipate approaching steep gradients and make the necessary gear changes ahead of time. A further research project could also aim at achieving comprehensive verification the model through evaluating different vehicles travelling over other routes, to ensure that the model can simulate a range of different vehicle and route conditions. It would also need to be determined through this study whether SimTrans is robust enough to manage different vehicle configurations.

In conclusion the modifications on the SimTrans model were successful in achieving a significant improvement in simulated results over the previous version of SimTrans and thus the primary objective of this research was achieved. SimTrans 2000 (the current modified version) is considered accurate enough to be valuable to transport managers as a decision support system for transport operations. Although further work may be required on the model to refine it, make it more user friendly and to include some extra modules for added functionality such as operating cost estimates, maintenance scheduling and side-by-side vehicle comparisons; the present model is a useful tool for the transport industry.

7. REFERENCES

- Amey R G, 1995. *Automotive Component Innovation: Development and Diffusion of Engine Management Technologies*. Technovation, May, 15, 4, p211-223.
- Anonymous, 1990. *EFI and Engine Management: A Comprehensive Guide to Diagnosis and Repair*. Gregory's Scientific Publications, Waterloo, Australia, Chap 1.
- Bauer H (Ed), 2000. *Automotive Handbook*. 5th ed. Robert Bosch GmbH, Stuttgart, Germany.
- Benninger N F and Plapp G, 1991. *Requirements and Performance of Engine Management Systems Under Transient Conditions*. SAE 910083.
- Bradley K, 1994. *Towards the Ultra Low Emission Engine*. Mechanical Technology, April, p23-24.
- Challen B J and Stobart R K, 1998. *Some More Diesel Engine Sensors*. International Congress and Exposition, Detroit, Michigan, USA, February. SAE 980167.
- Chowanietz E G, 1995. *Automobile Electronics in the 1990s. Part 1: Powertrain Electronics*. Electronics and Communication Engineering Journal, February, Vol 7, Iss 1, p23-36.
- Clark D J, 1996. *Model For Simulating On-Road Vehicle Performance*. Unpublished MScEng Dissertation, School of Bioresources Engineering and Environmental Hydrology, University of KwaZulu-Natal, Pietermaritzburg, RSA.
- Clark D J, Lyne P W L, de la Harpe E R and Hansen A C, 1996. *Rolling Resistance Of Pneumatic Tyres*. J. Agricultural Engineering in SA, SA Institute of Agricultural Engineering, Vol 28, 16 - 24.

- Cole A K, van Heerden C and Hundley K M, 1999. *Evaluation of the SimTrans Model and its Use in a Working Environment*. Procedures of SA Sugar Technology Association 73: p44-45.
- de la Harpe E R, Lyne P W L and Hansen A C, 1998. *SimTrans: A Computer Model For Predicting Tractor Performance On Hard Surfaces*. ASAE Annual International Meeting, Orlando, Florida, USA. ASAE 981082.
- Fenton J, 1996. *Focus on Networking of On-Board Vehicle Electronic Systems*. Automotive Engineer, Vol 21, Iss 3, June-July, p50-64.
- Ganser M A, 1998. *Common Rail with Injection Rate Control*. Future Transportation Technology Conference and Exposition, Costa Mesa, CA, August. SAE 981927.
- Govindasamy P, 2002. *Predicting emissions using an on-road vehicle performance simulator*. Unpublished MScEng Dissertation, School of Bioresources Engineering and Environmental Hydrology, University of KwaZulu-Natal, Pietermaritzburg, RSA.
- Hellberg Transport Management, 2006. HTM TransSolve Version 7 [Internet] Hellberg Transport Management, Durban, RSA. Available from: <http://www.htm.co.za>. [Accessed: 12 October 2006]
- Hlousek J, 1998. *Electronically Controlled Fuel Injection Systems for Medium Speed Diesel Engines*. Future Transportation Technology Conference and Exposition, Costa Mesa, CA, August. SAE 981928.
- Ishiguro S and Masui T, 1999. *Engine Control Optimization for Heavy-Duty Cargo Trucks*. International Truck and Bus Meeting and Exposition, Detroit, Michigan, USA, November. SAE 1999-01-3723.
- Johnson D R and Diamond S, 2001. *Heavy Vehicle Propulsion Materials: Recent Progress and Future Plans*. Government/Industry Meeting Washington D.C., USA, May. SAE 2001-01-2061

- Jordan P D, 1987a. *Microprocessor-Controlled Engine Management*. Electronics and Wireless World, September, Vol 93, Iss 1619, p876-880.
- Jordan P D, 1987b. *Advanced Engine Management Systems*. Electronics and Wireless World, December, Vol 93, Iss 1622, p1225-1228.
- Krebs S, 1993. *Advanced Engine Management Systems: The Key to Reduced Emissions and Improved Performance*. Siemens Review – R and D Special – Fall, p14-17.
- Lyne P W L, 2002. *Personal Communication*. School of Bioresources Engineering and Environmental Hydrology, University of KwaZulu-Natal, Pietermaritzburg, RSA, October.
- Lyne P W L, 2004. *Personal Communication*. South African Sugarcane Research Institute (SASRI), Durban, RSA, July.
- Lyne P W L, de la Harpe E, Hansen A C and Clark D J, 1996. *The Use of a Simulator to Optimise the Selection of a Haulage Vehicle / Route*. Procedures of SA Sugar Technology Association 70: p67.
- Lyne P W L, Clark D J, de la Harpe E and Hansen A C, 1998. *Development of a Vehicle Performance Simulator: Sensitivity Analysis*. Procedures of SA Sugar Technology Association 72: p24-27.
- Lyne P W L, Meyer E and Herbert R, 2005. *Evaluation of Two Possible Haulage Routes for the Transport of Sugarcane*. 2005 Procedures of SA Sugar Technology Association 79: p386-393.
- Pawlak A M, 1999. *Distributed Variable Reluctance Sensors*. International Congress and Exposition, Detroit, Michigan, USA, March. SAE 1999-01-1039.
- Pazul K, 1999. *Controller Area Network (CAN) Basics*. Microchip Technology Inc. AN713.

- Probst C O, 1991. *Bosch Fuel Injection & Engine Management*. Robert Bentley Inc., Cambridge, Massachusetts, USA, Chap 1-2.
- Riesenberg K O, Faupel W, Blaich B, Stumpp G, Ungerer E and Pfeifer U, 1999. *Diesel Fuel-Injection: An Overview*. 3rd ed. Robert Bosch GmbH, Stuttgart, Germany.
- SAE, 1990. *Vehicle Dynamics Terminology*. SAE J1594 Jun 87. SAE Handbook, 4: 28.07.
- Sparks D, Noll T, Agrotis D, Betzner T and Gschwend K, 1999. *Multi-Sensor Modules with Data Bus Communication Capability*. International Congress and Exposition, Detroit, Michigan, USA, March. SAE 1999-01-1277.
- Torkzadeh D D, Kiencke U and Keppler M, 2002. *Introduction of a New Non-Invasive Pressure Sensor for Common-Rail Systems*. SAE 2002 World Congress, Detroit, Michigan, USA, March. SAE 2002-01-0842.
- Uno S, Hirayama N, Watanabe A and Tsuchiya F, 2001. *Sophisticated CAN on Embedded Microcontrollers for Smart In-Vehicle Real-Time Control Systems*. Automotive and Transportation Technology Congress and Exhibition, Barcelona, Spain, October. SAE 2001-01-1218.
- Viljoen R, 1996. *After Fly-by-Wire We Now (Almost) Have Drive-by-Wire*. South African Transport, August, p61-69.
- Ward D K and Fields H L, 2000. *A Vision of the Future of Automotive Electronics*. SAE 2000 World Congress, Detroit, Michigan, USA, March. SAE 2000-01-1358.
- Webster J, 1995. *Road Traffic Legislation In Relation To The Conveyance Of Timber In The Forest Industry*. In Focus On Forest Engineering. Forest Engineering South Africa (FESA), Forest Engineering Technology, Faculty of Forestry, University of Stellenbosch, Stellenbosch, Western Cape, RSA.
- White C, 1997. *Automotive Engine Management Systems and Fuel Injection Techbook*. Haynes Publishing, Sparkford, England, Chap 1.

Wright D S, 1990. *Transportation – “Current On-Board Vehicle Systems”*. Transportation, Vol 17, Iss 3, p239-250.

Yotong Z, Changqi C and Xinghua L, 1998. *Study of Diesel Engine Management System*. Journal of Beijing Institute of Technology (English Edition), Procedures of SA Sugar Technology Association Vol 7, No. 3, p280-285.

Zeltwanger H, 1999. *CANopen Networks in Vehicles*. International Congress and Exposition, Detroit, Michigan, USA, March. SAE 1999-01-1275.

8. APPENDICES

APPENDIX A – DEFINITIONS

Drivability – The ability of the vehicle to provide smooth, trouble free performance under virtually any condition while delivering power, fuel economy, and controlled emissions (Probst, 1991).

APPENDIX B – FLOWCHARTS FOR THE ORIGINAL SIMTRANS MODEL

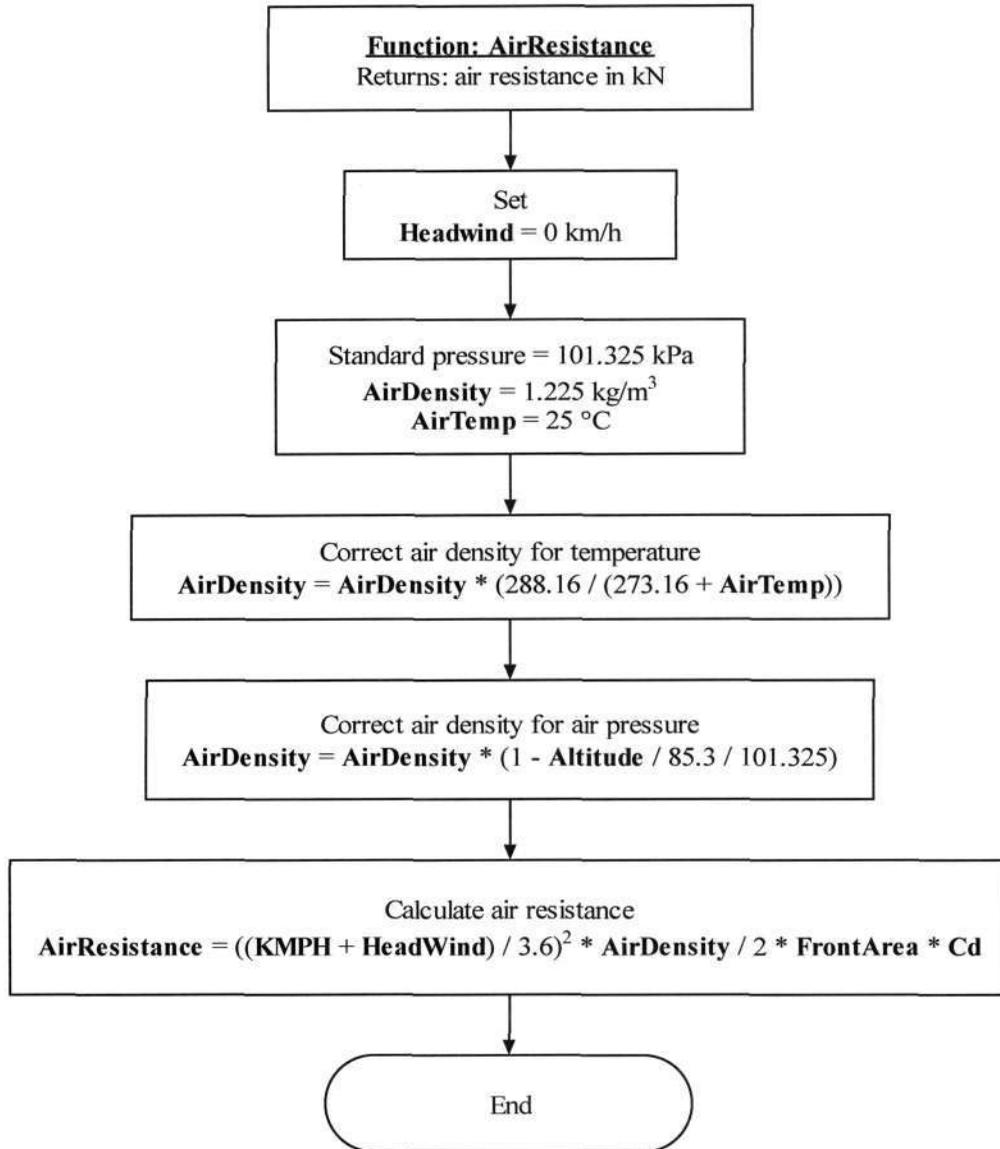


Figure B.1 Flowchart illustrating the algorithm for the AirResistance function

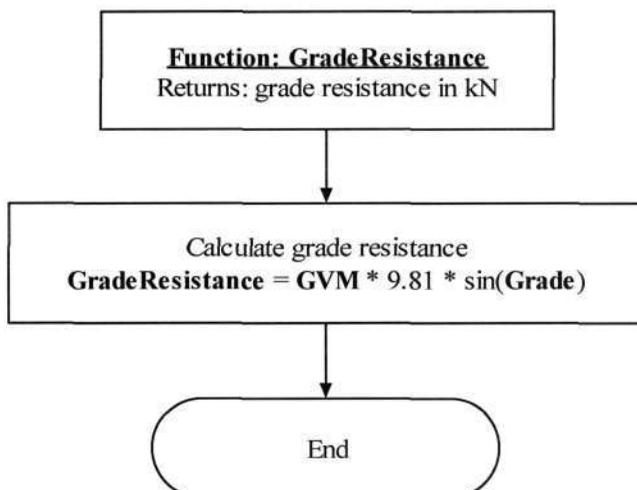


Figure B.2 Flowchart illustrating the algorithm for the GradeResistance function

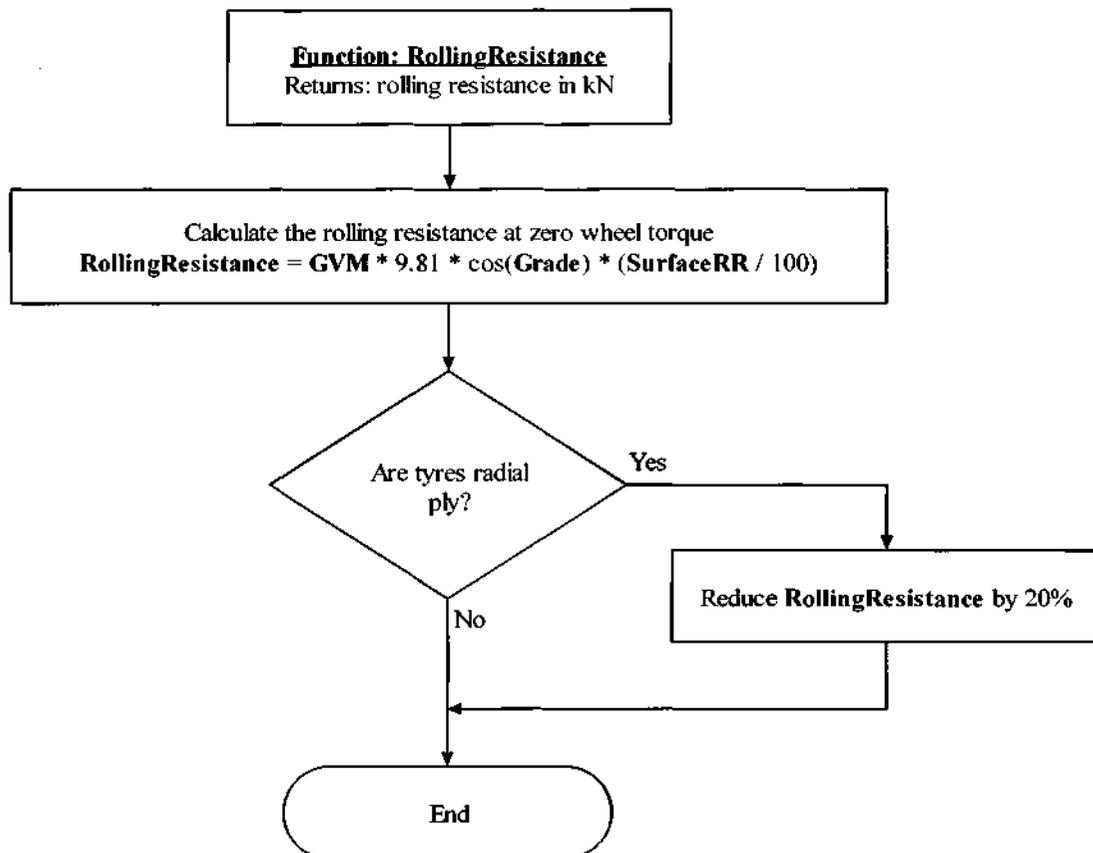


Figure B.3 Flowchart illustrating the algorithm for the RollingResistance function

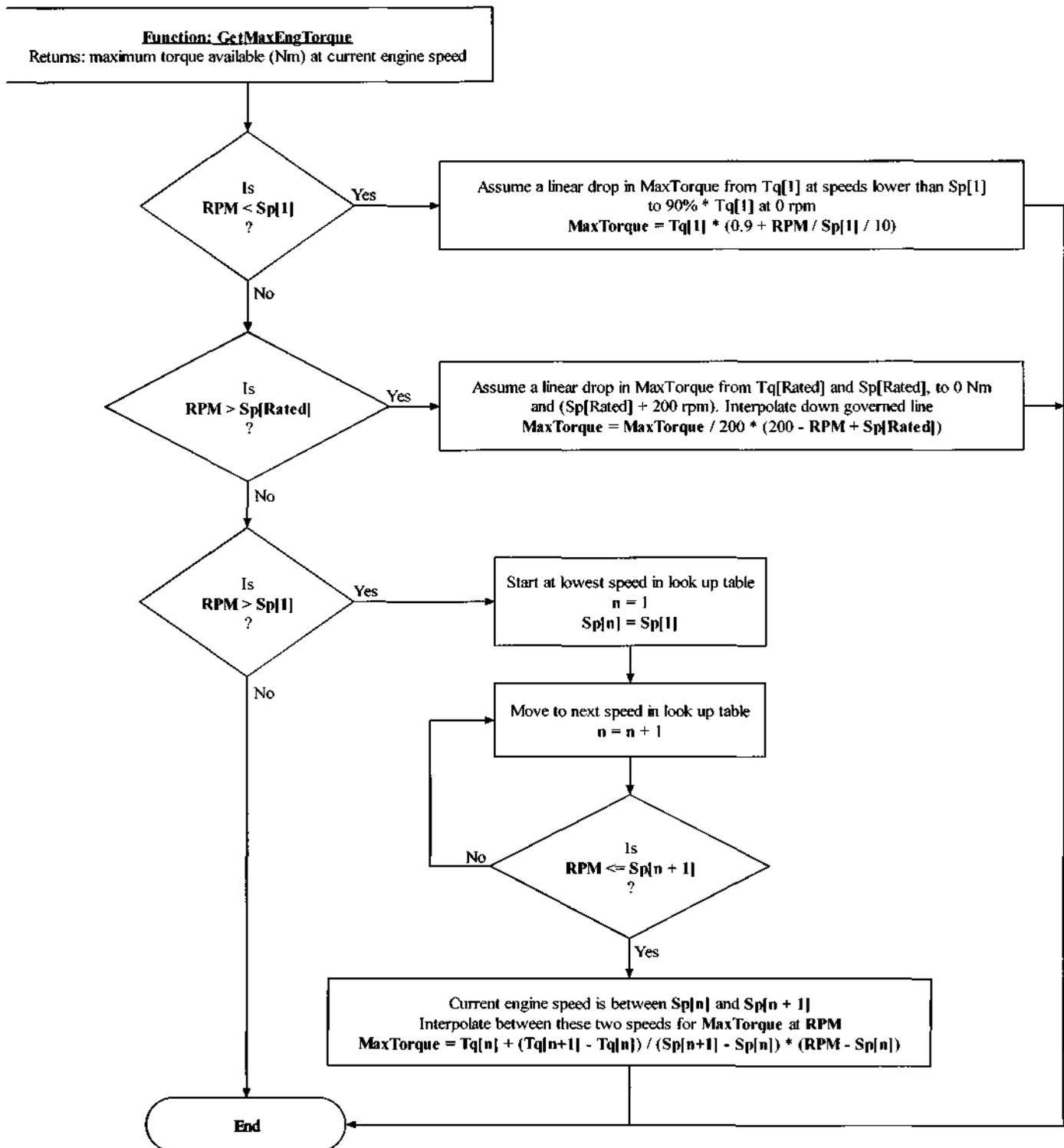


Figure B.4 Flowchart illustrating the algorithm for the GetMaxEngTorque function

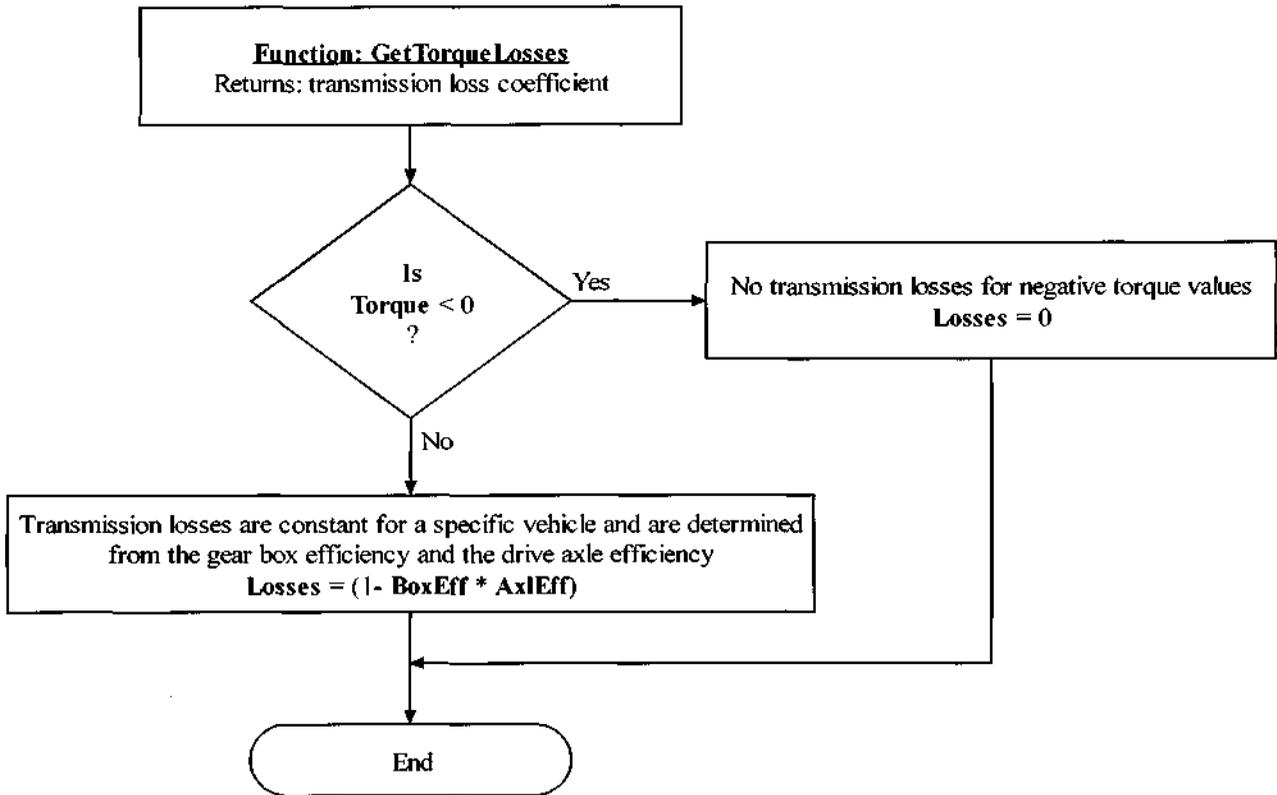


Figure B.5 Flowchart illustrating the algorithm for the GetTorqueLosses function

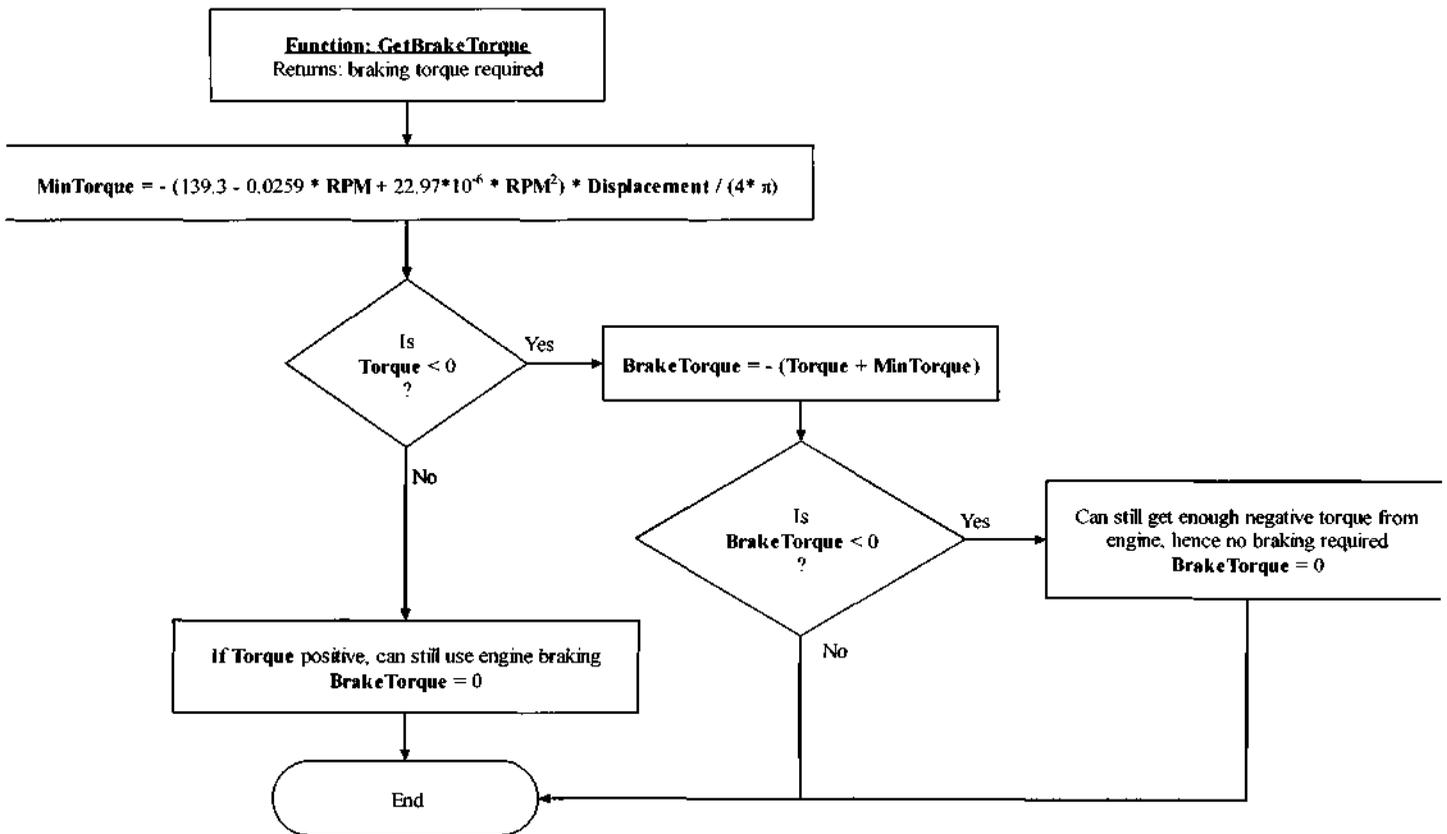


Figure B.6 Flowchart illustrating the algorithm for the GetBrakeTorque function

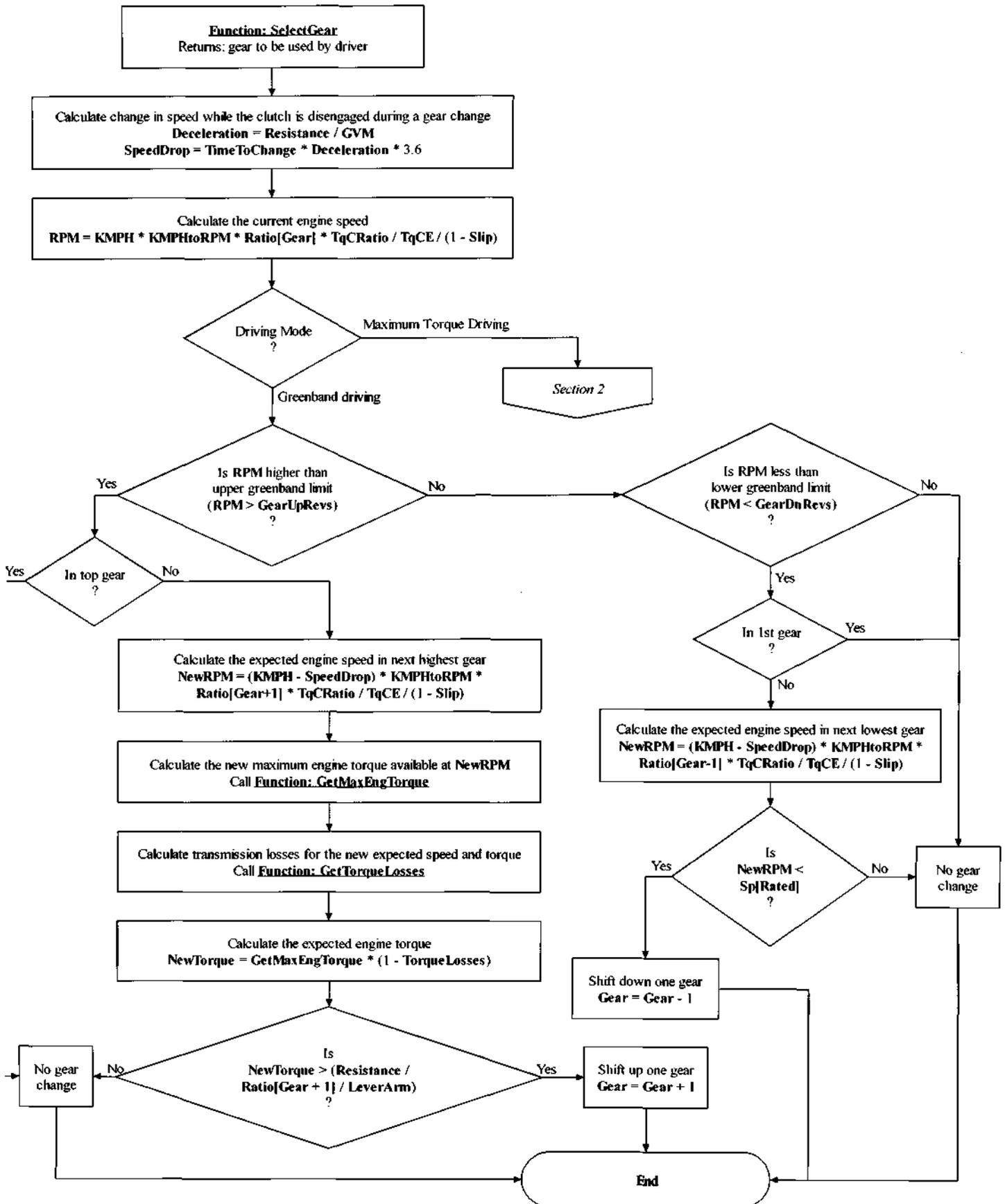


Figure B.7.1 Flowchart illustrating the algorithm for the SelectGear function

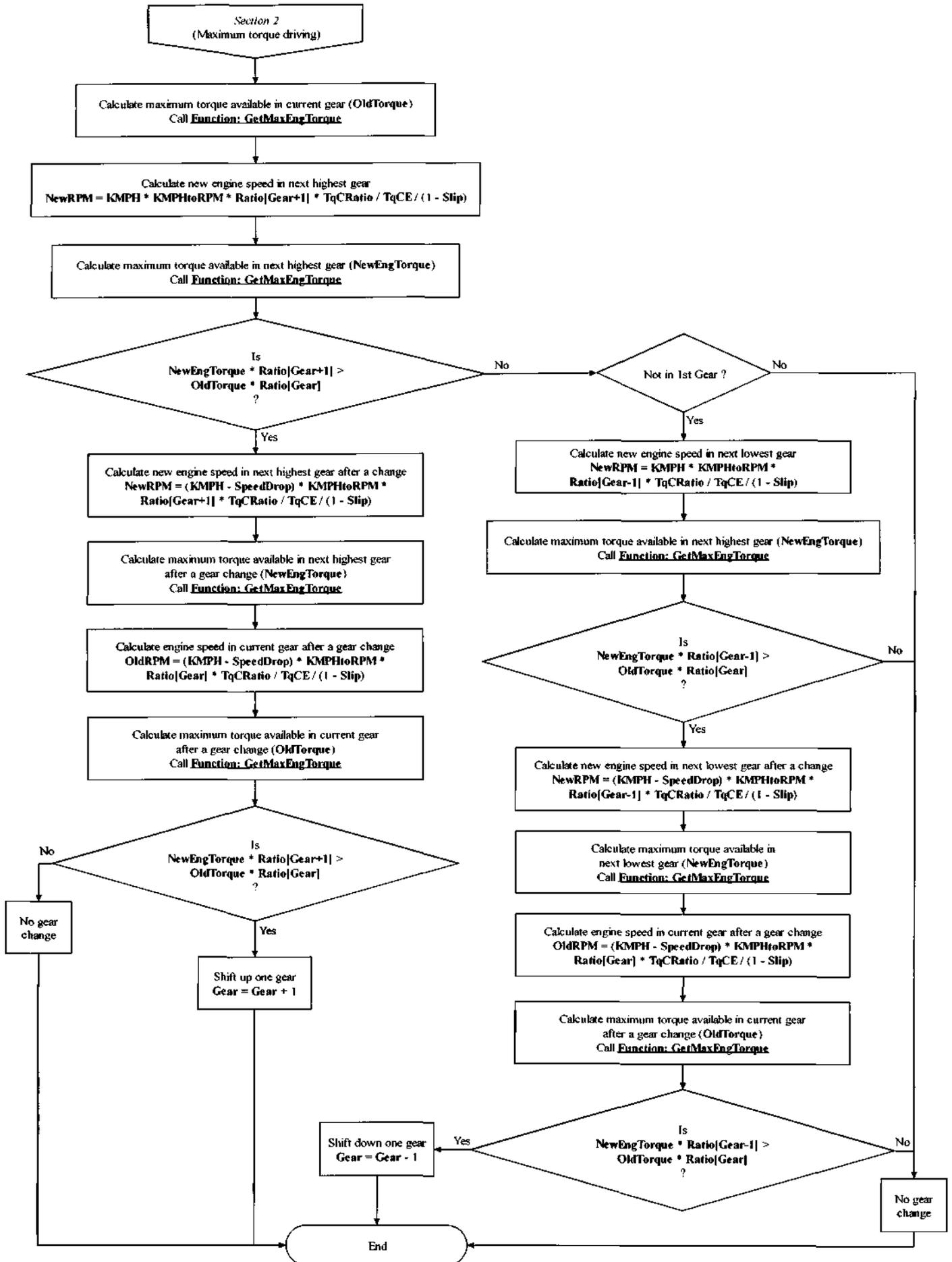


Figure B.7.2 Flowchart illustrating the algorithm for the SelectGear function

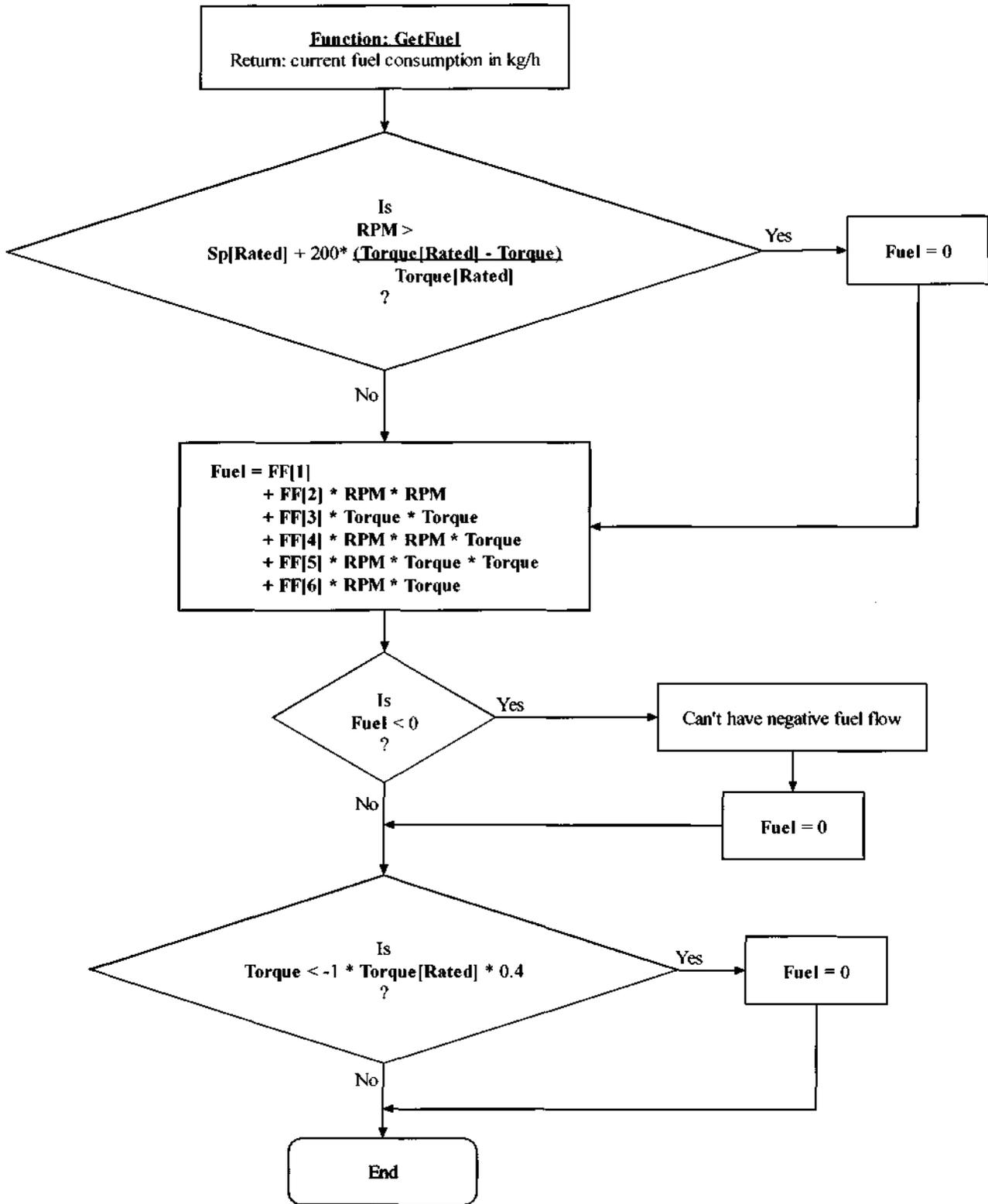


Figure B.8 Flowchart illustrating the algorithm for the GetFuel function

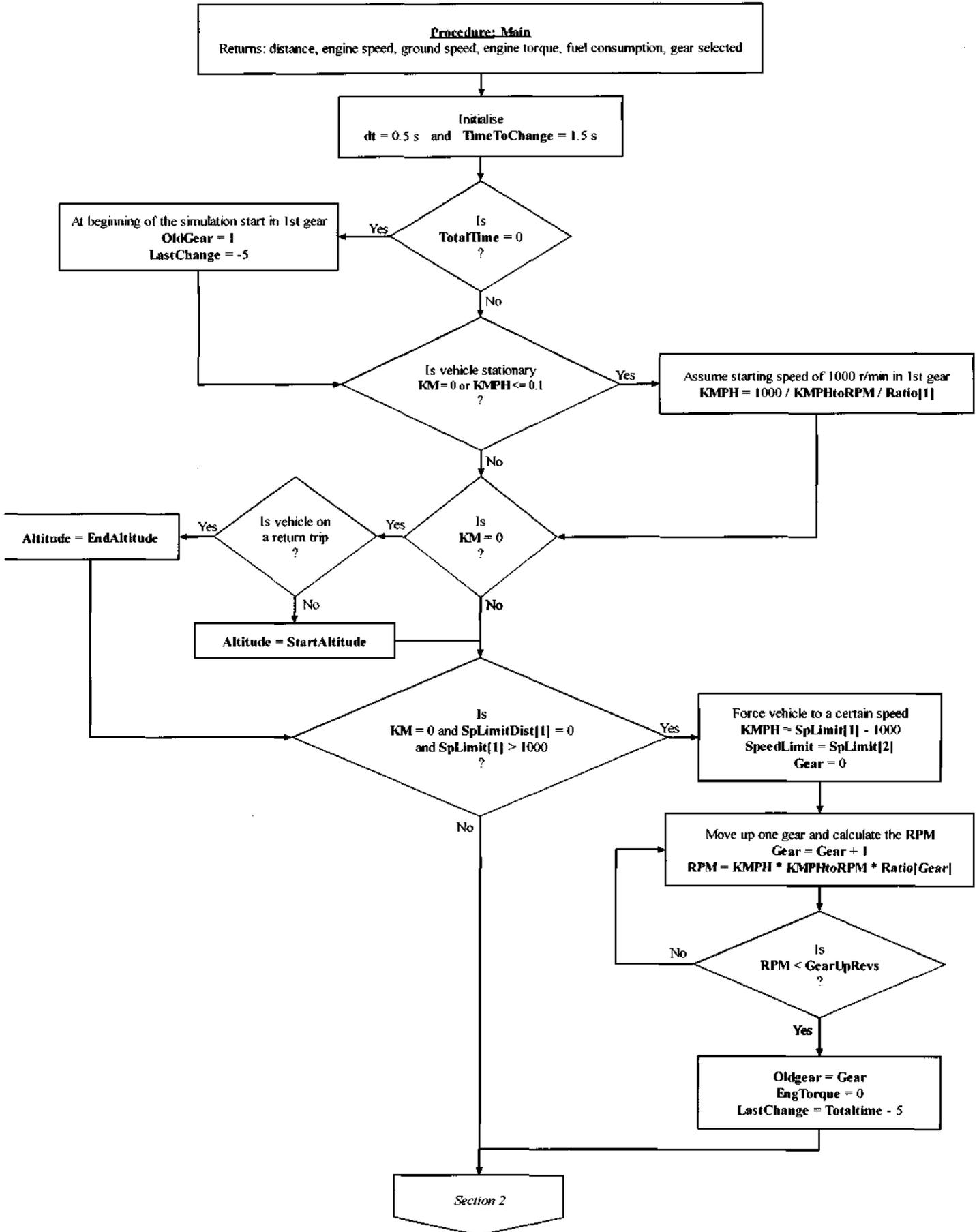


Figure B.9.1 Flowchart illustrating the algorithm for the Main procedure

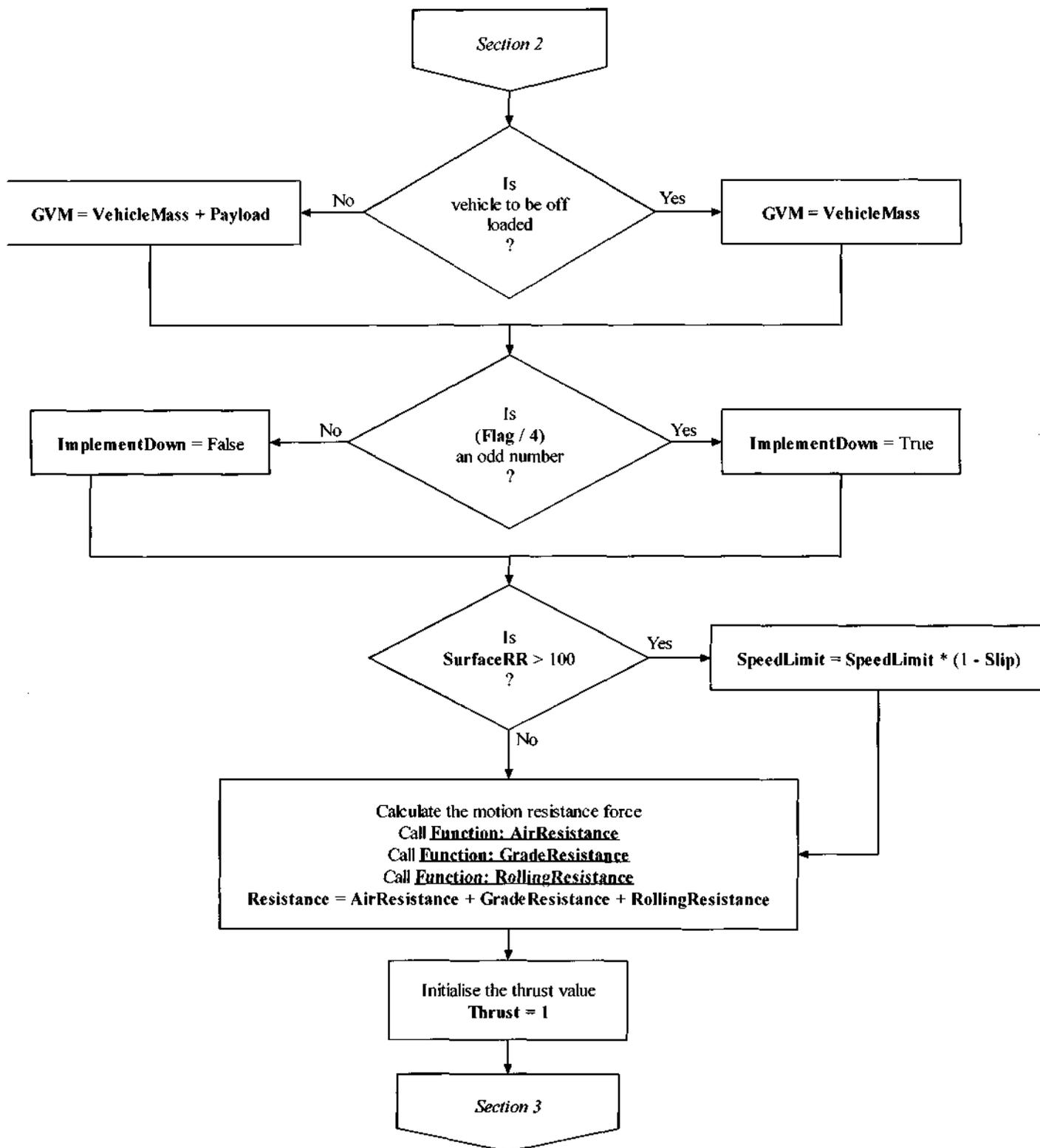


Figure B.9.2 Flowchart illustrating the algorithm for the Main procedure

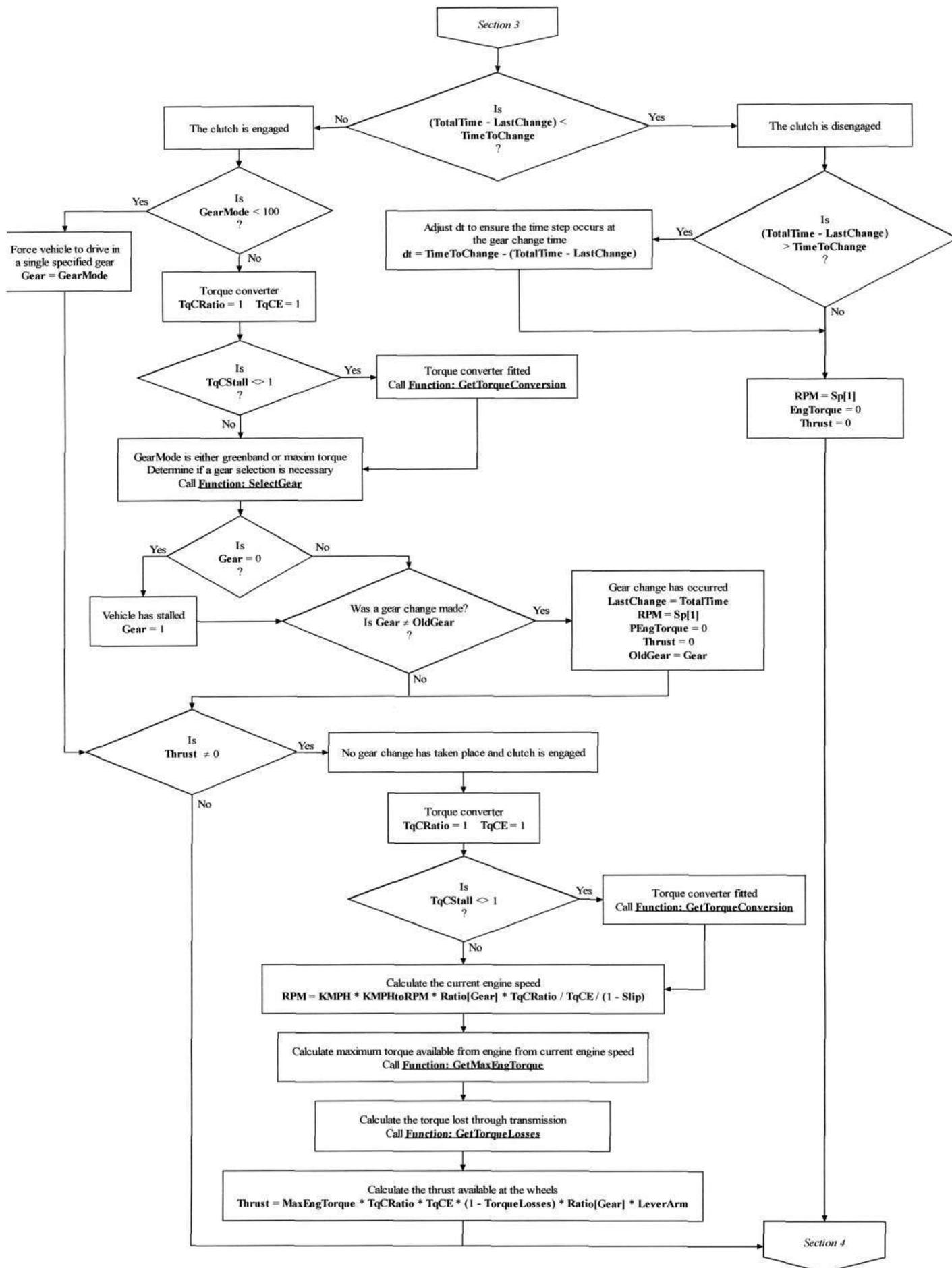


Figure B.9.3 Flowchart illustrating the algorithm for the Main procedure

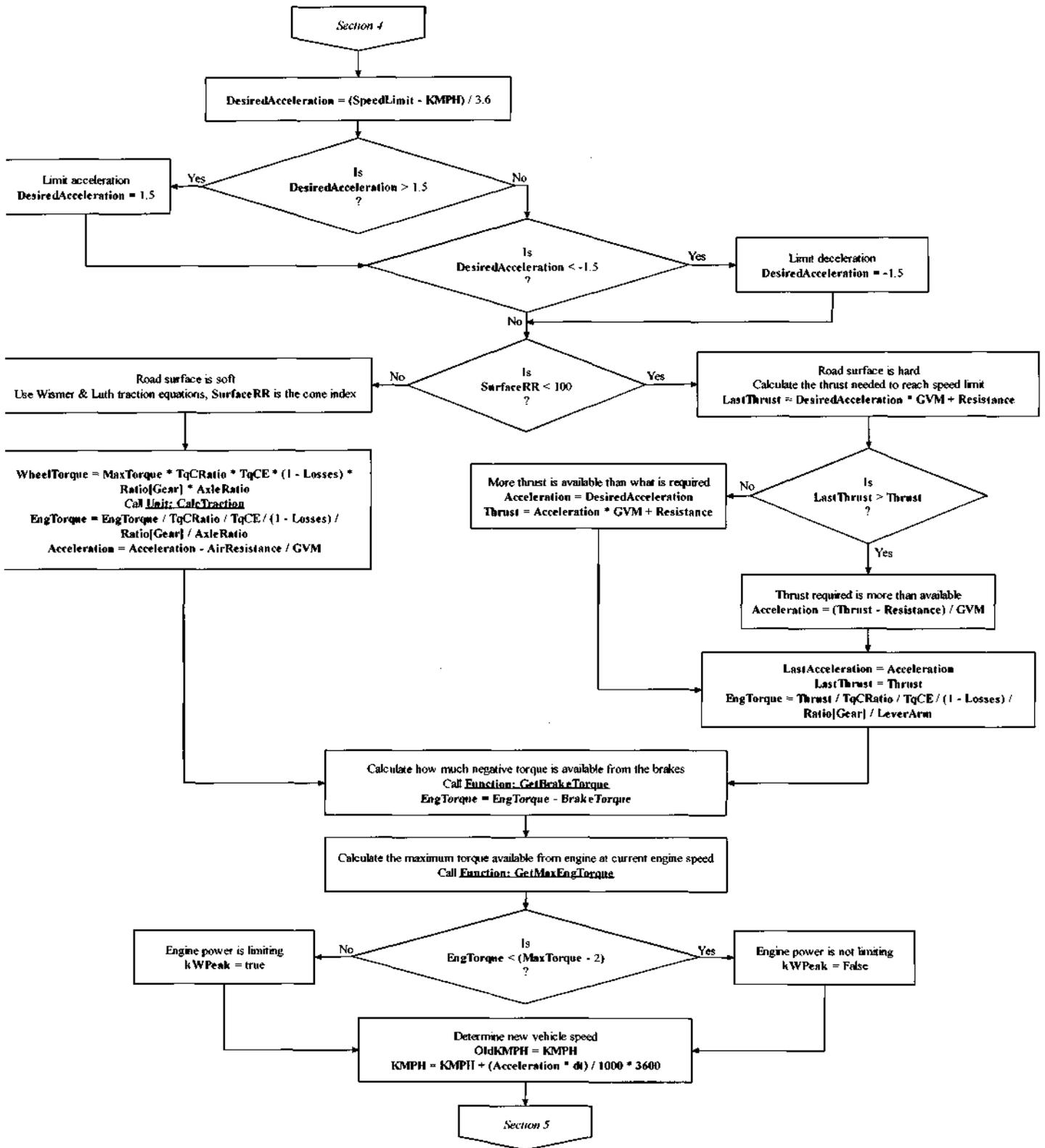


Figure B.9.4 Flowchart illustrating the algorithm for the Main procedure

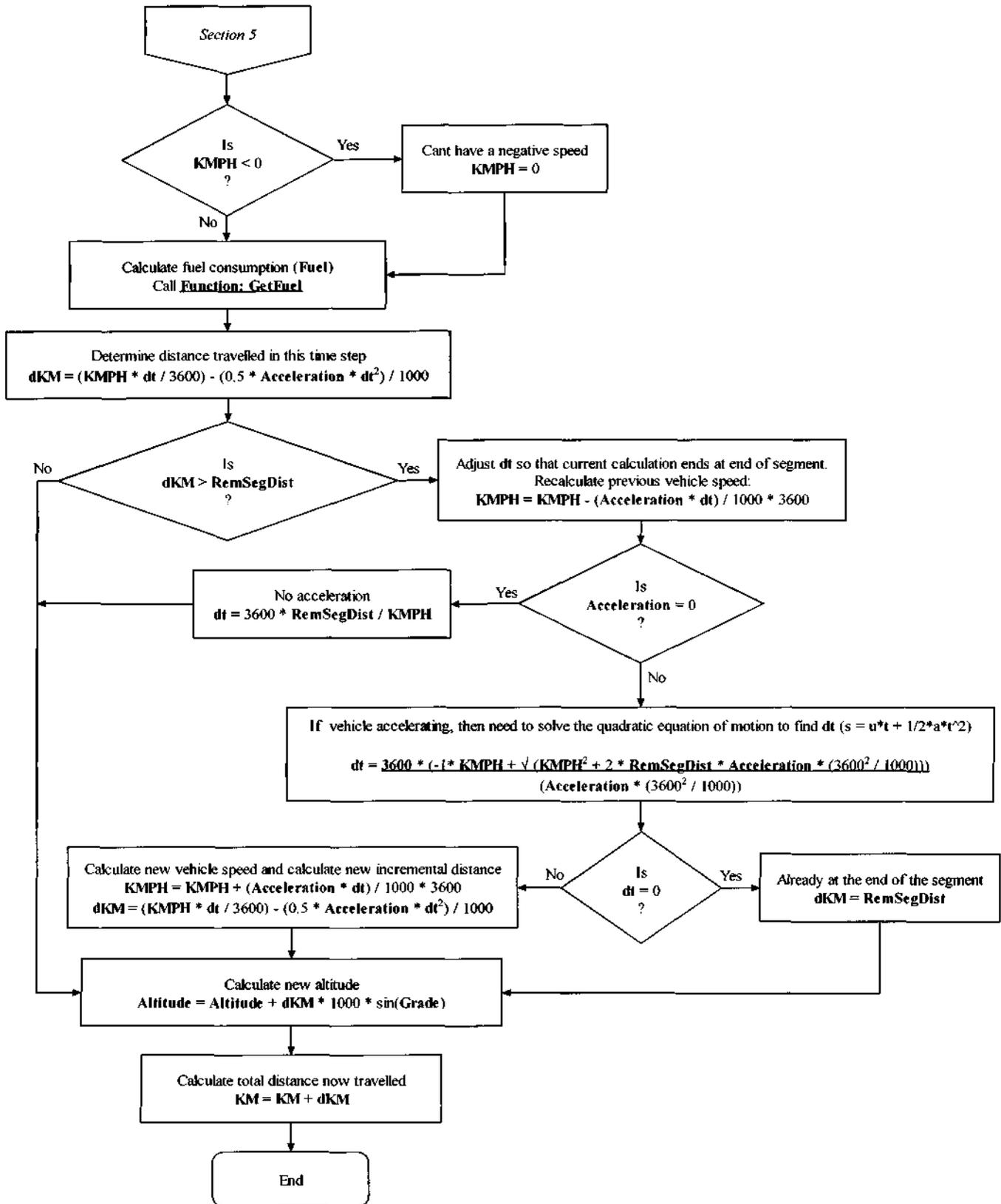


Figure B.9.5 Flowchart illustrating the algorithm for the Main procedure

APPENDIX C – FLOW METER CALIBRATION

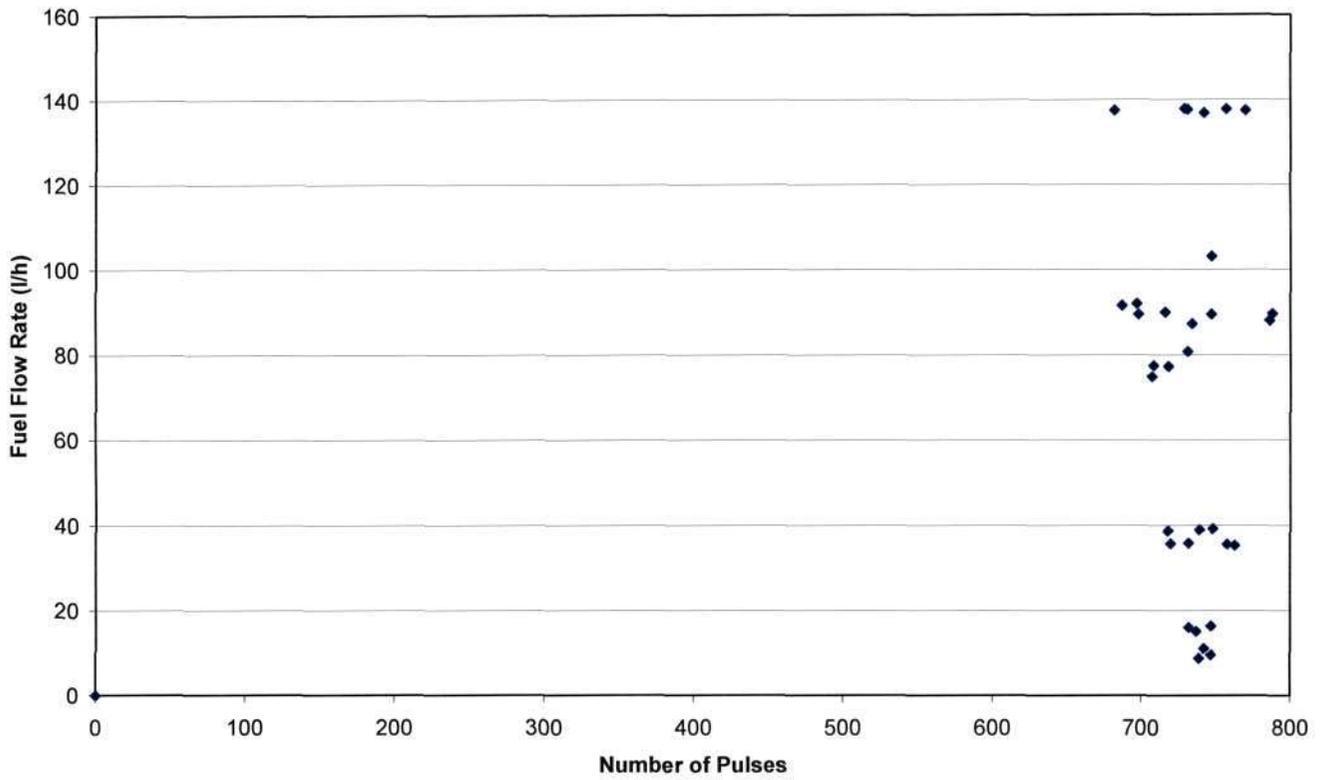


Figure C.1 Flow rate distribution plot for flow meter 83014

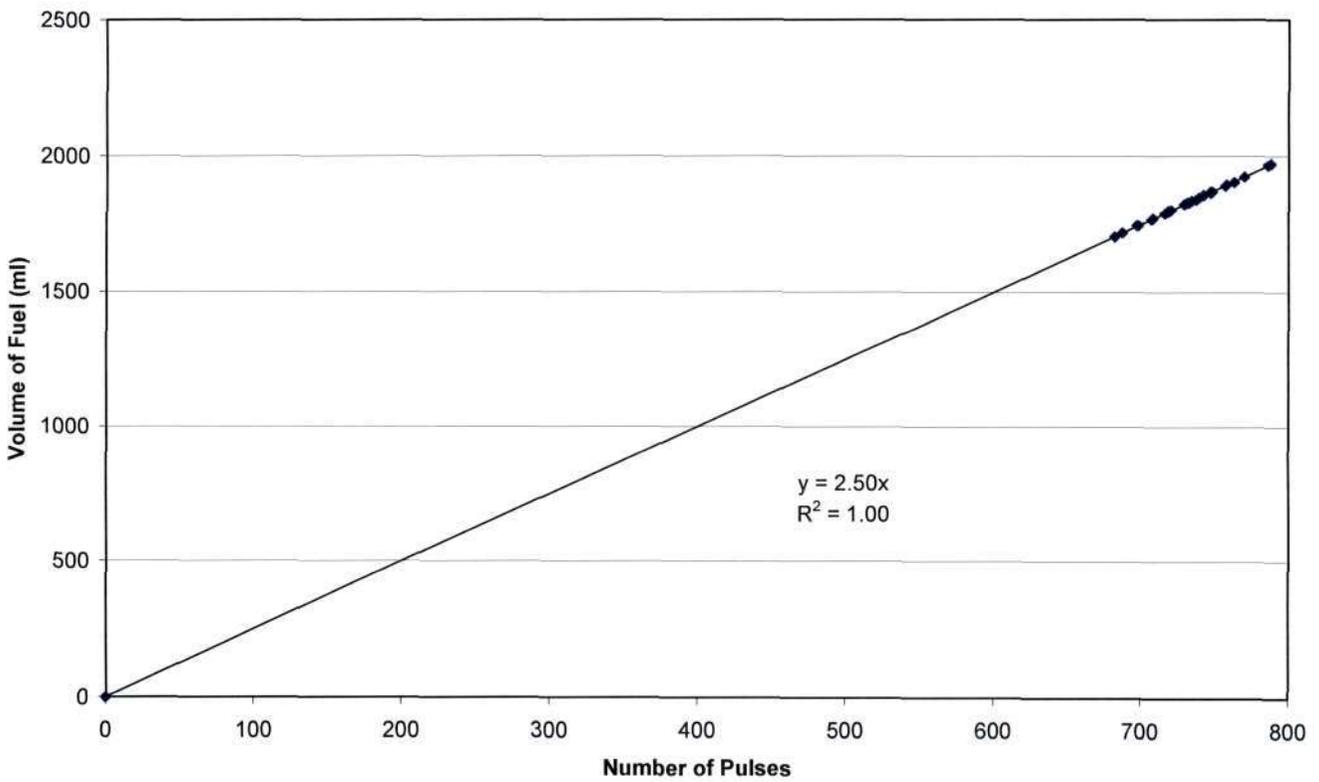


Figure C.2 Calibration plot for flow meter 83014

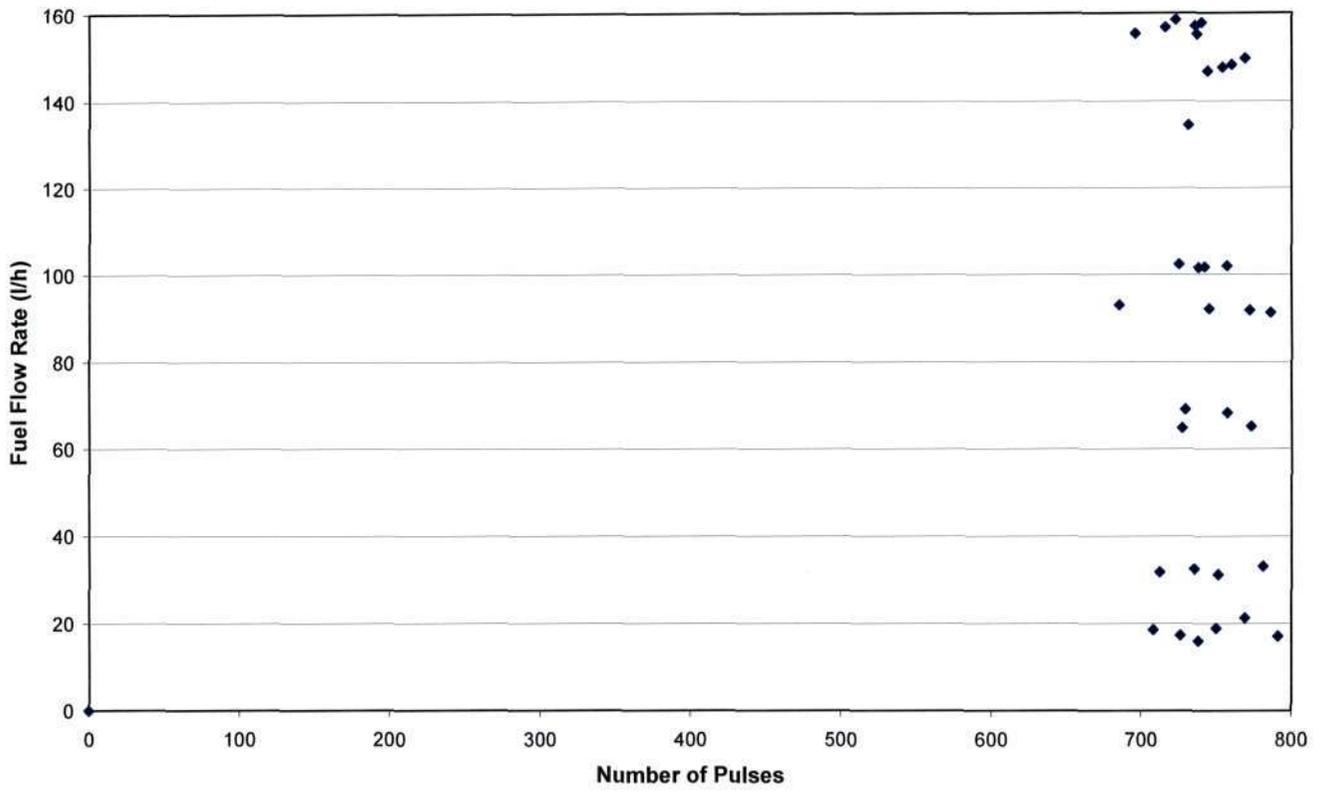


Figure C.3 Flow rate distribution plot for flow meter 83015

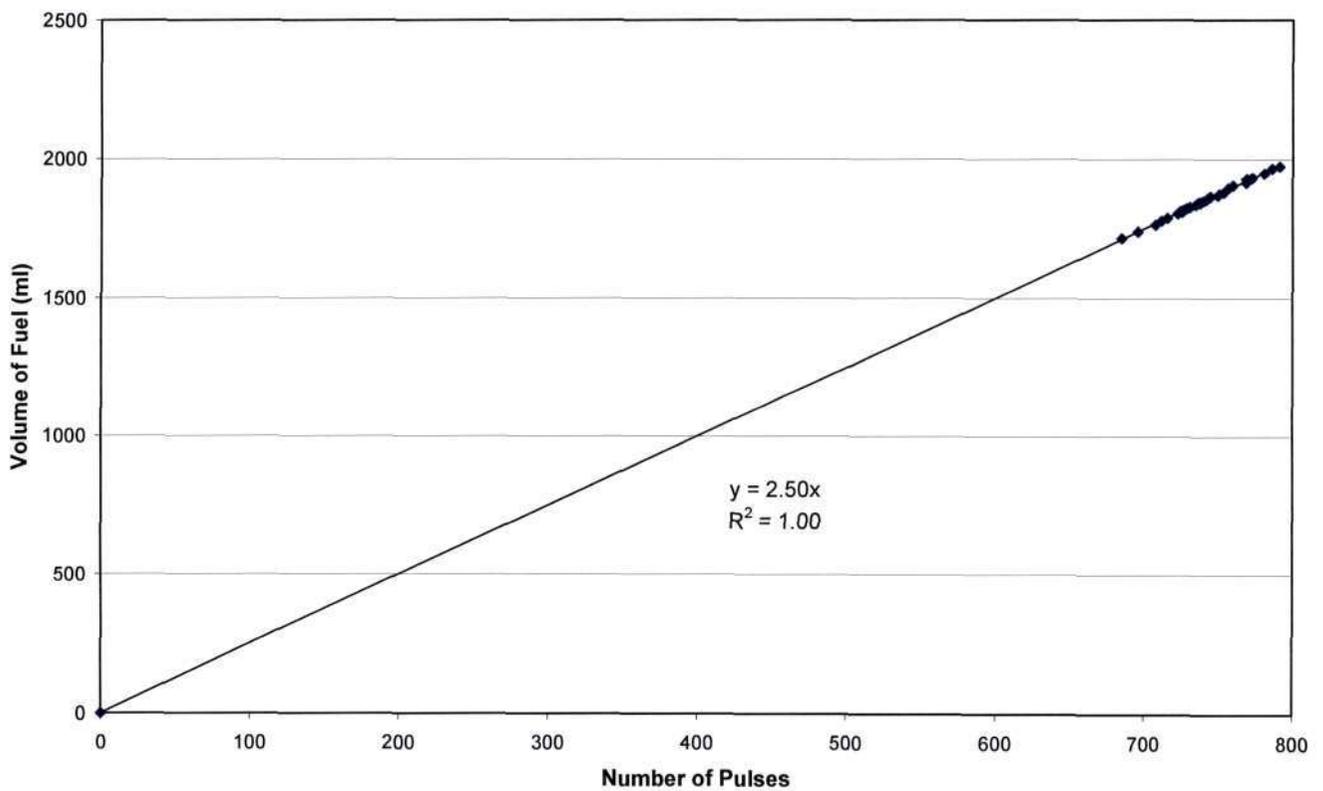


Figure C.4 Calibration plot for flow meter 83015

APPENDIX D – FUEL FLOW RATE MEASUREMENT SYSTEM TESTING

Table D.1 Observed and measured data from fuel flow tests on dynamometer

| Test # | Observed Data | | | | | Measured Data | | | | |
|--------|---------------|-------------|-------------|-----------|------------------|---------------|-------------|-------------|-----------|------------------|
| | Time (s) | Supply (ml) | Return (ml) | Used (ml) | Fuel Flow (ml/s) | Time (s) | Supply (ml) | Return (ml) | Used (ml) | Fuel Flow (ml/s) |
| 1 | 545.80 | 2000 | 525 | 1475 | 2.702 | 546.00 | 2007.5 | 520.0 | 1487.5 | 2.724 |
| 2 | 1365.20 | 2000 | 1655 | 345 | 0.253 | 1365.00 | 2002.5 | 1657.5 | 345.0 | 0.253 |
| 3 | 646.00 | 2000 | 650 | 1350 | 2.090 | 647.00 | 2002.5 | 642.5 | 1360.0 | 2.102 |
| 4 | 1153.10 | 2000 | 1216 | 784 | 0.680 | 1156.00 | 2000.0 | 1212.5 | 787.5 | 0.681 |
| 5 | 1153.10 | 2000 | 1004 | 996 | 1.015 | 985.00 | 2002.5 | 1005.0 | 997.5 | 1.013 |
| 6 | 712.35 | 2000 | 690 | 1310 | 1.839 | 712.00 | 1997.5 | 685.0 | 1312.5 | 1.843 |
| 7 | 1162.80 | 3000 | 1231 | 1769 | 1.521 | 1163.00 | 3002.5 | 1235.0 | 1767.5 | 1.520 |
| 8 | 1121.28 | 3000 | 1180 | 1820 | 1.623 | 1122.00 | 3000.0 | 1182.5 | 1817.5 | 1.620 |
| 9 | 1488.60 | 3000 | 1580 | 1420 | 0.954 | 1489.00 | 3000.0 | 1592.5 | 1407.5 | 0.945 |
| 10 | 699.75 | 2000 | - | 2000 | 2.858 | 699.00 | 2682.5 | 682.5 | 2000.0 | 2.861 |
| 11 | 1471.50 | 2000 | - | 2000 | 1.359 | 1471.00 | 3535.0 | 1537.5 | 1997.5 | 1.358 |
| 12 | 3013.91 | 2000 | - | 2000 | 0.664 | 3013.25 | 5225.0 | 3232.5 | 1992.5 | 0.661 |

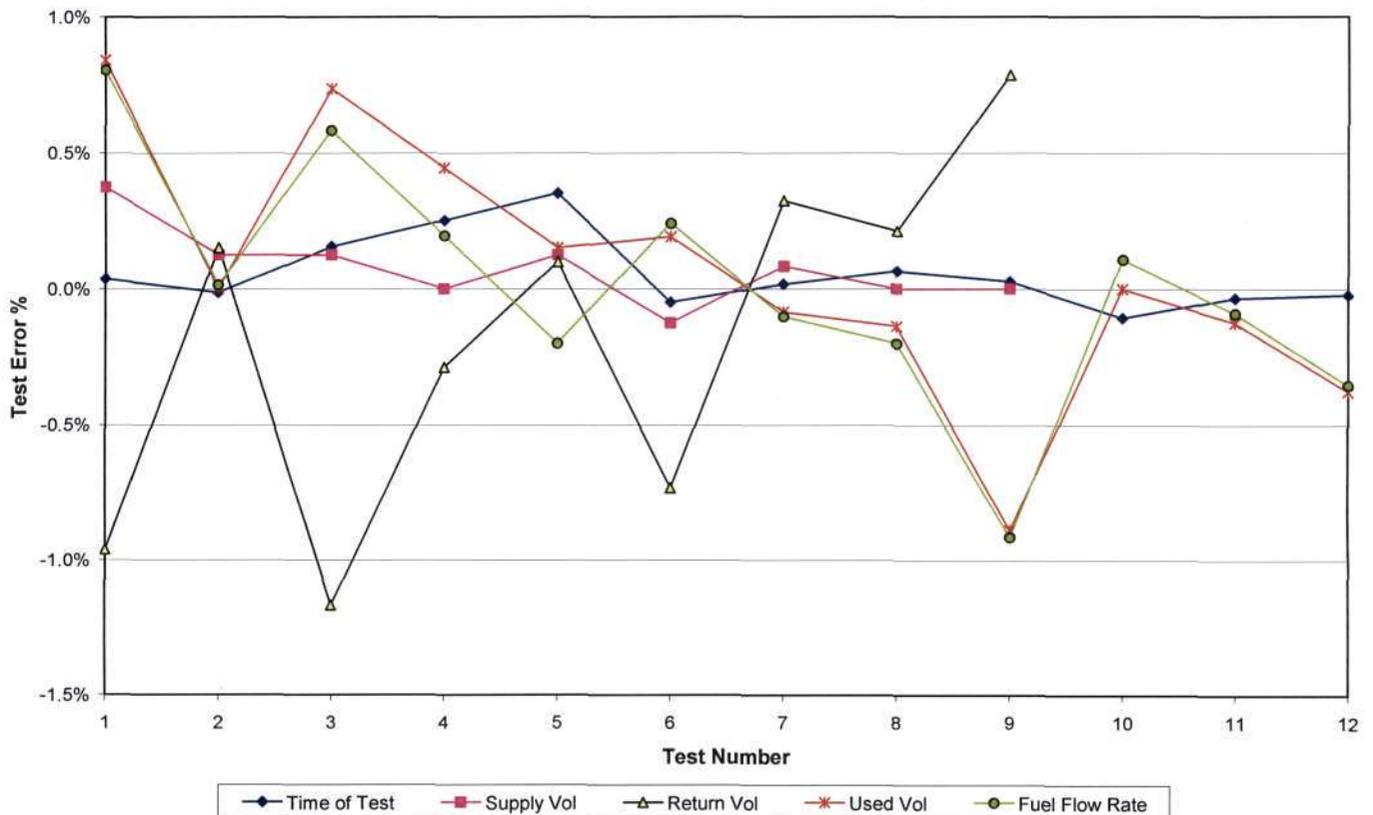


Figure D.1 Error magnitude comparing observed and measured data

APPENDIX E – ENGINE SPEED CALIBRATION

Table E.1 Measured engine speed and pulses from the tachograph

| Hand Tachometer (r/min) | CR10X reading (Tachometer output) |
|----------------------------|--------------------------------------|
| 0.0 | 0 |
| 661.0 | 292 |
| 661.0 | 292 |
| 662.8 | 292 |
| 663.0 | 293 |
| 666.0 | 293 |
| 693.4 | 305 |
| 1231.0 | 562 |
| 1262.0 | 549 |
| 1319.0 | 581 |
| 1326.0 | 603 |
| 1356.0 | 596 |
| 1360.0 | 587 |
| 2007.0 | 884 |
| 2047.0 | 902 |
| 2073.0 | 914 |
| 2083.9 | 919 |

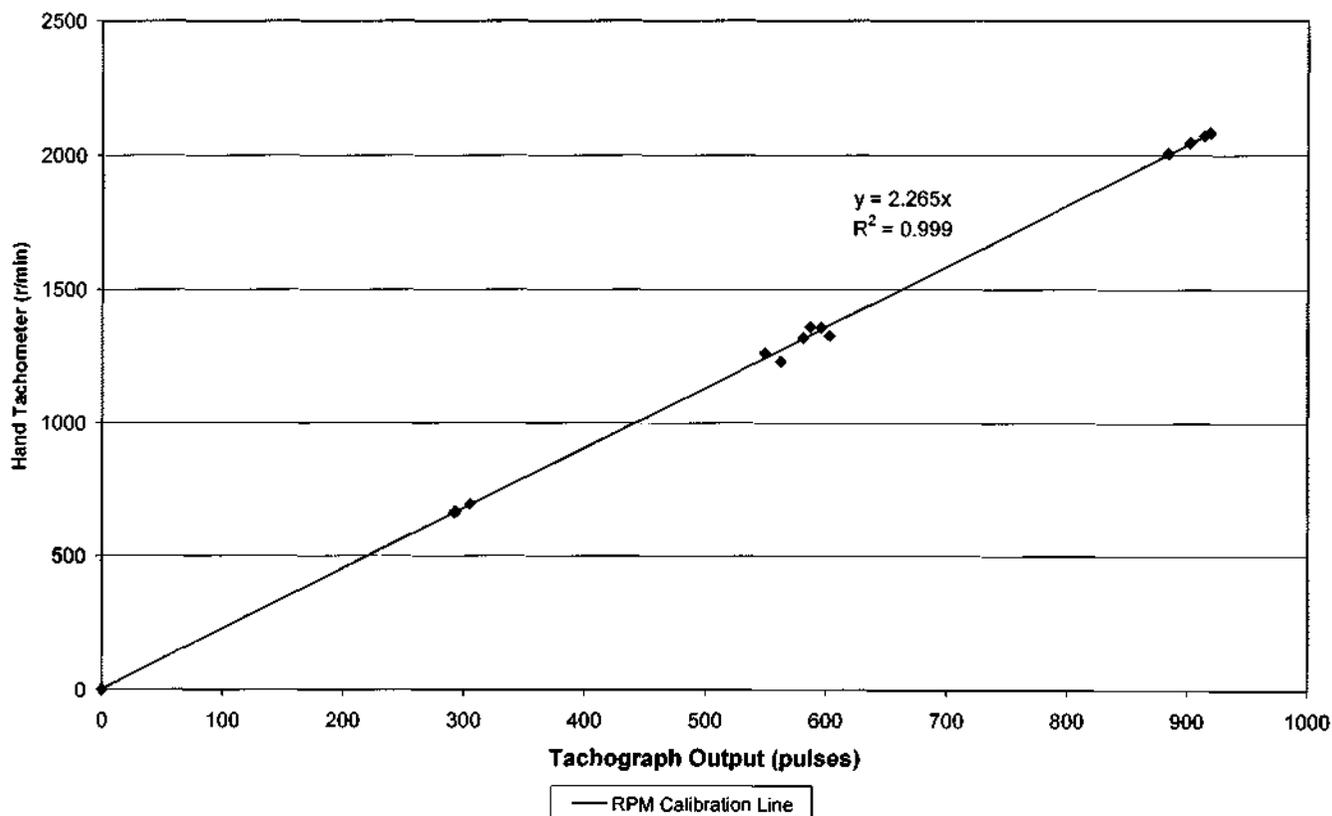


Figure E.1 Engine speed calibration graph

APPENDIX F – VEHICLE SPEED CALIBRATION

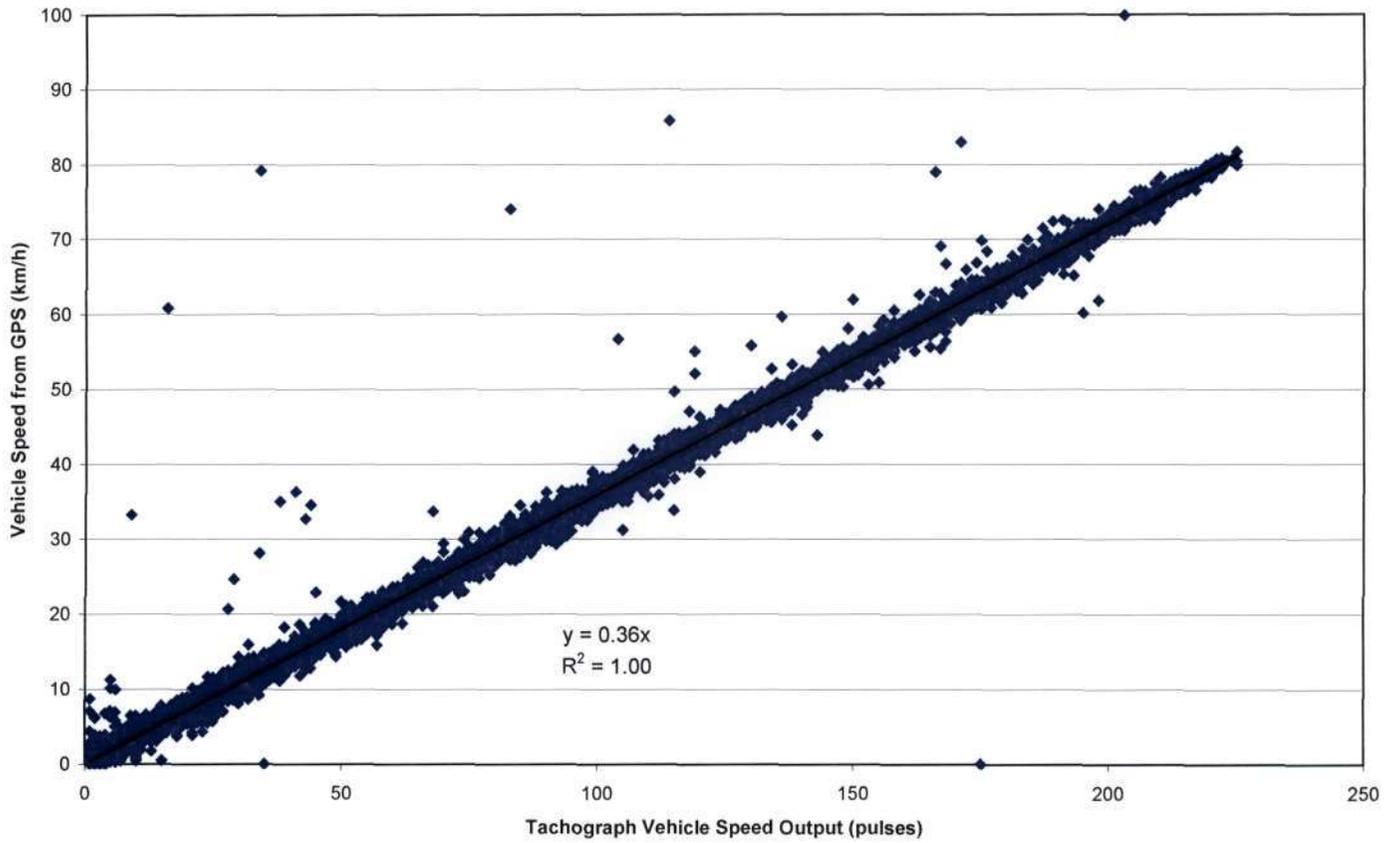


Figure F.1 Calibration plot of tachograph vehicle speed output

APPENDIX G – FLOWCHARTS ILLUSTRATING THE CR10X PROGRAM CYCLE

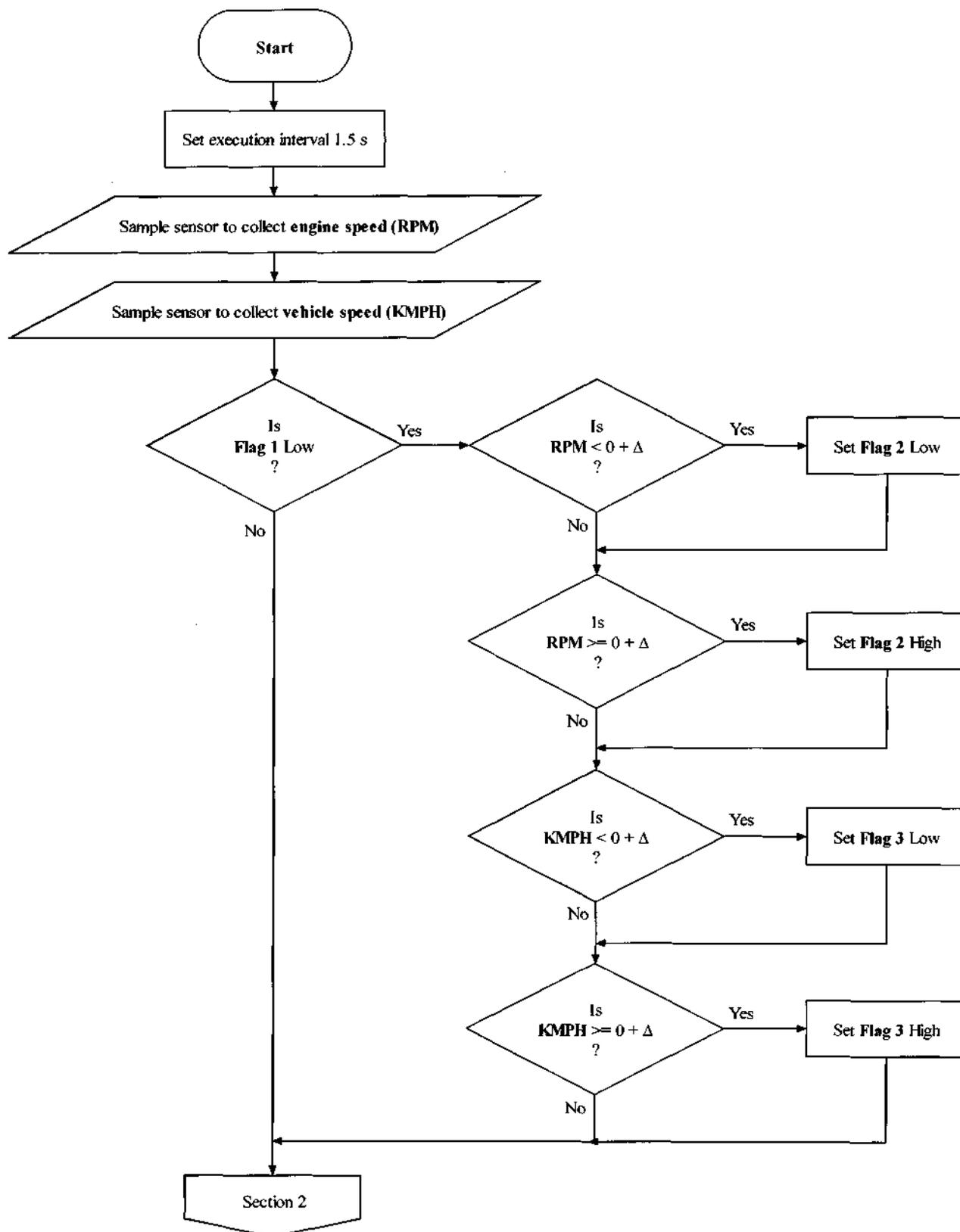


Figure G.1.1 Flowchart illustrating the algorithm for the Main procedure of the CR10X program

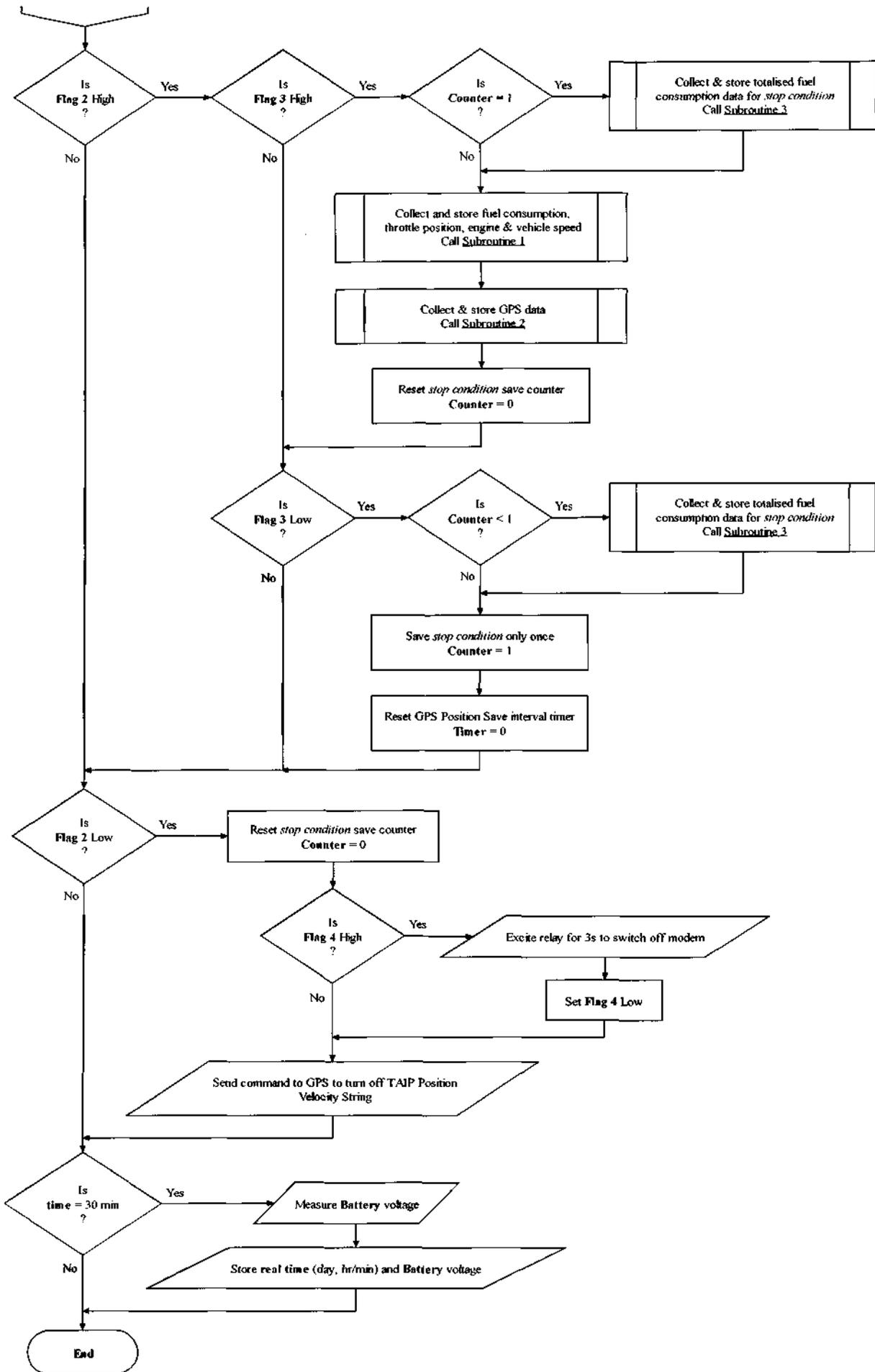


Figure G.1.2 Flowchart illustrating the algorithm for the Main procedure of the CR10X program

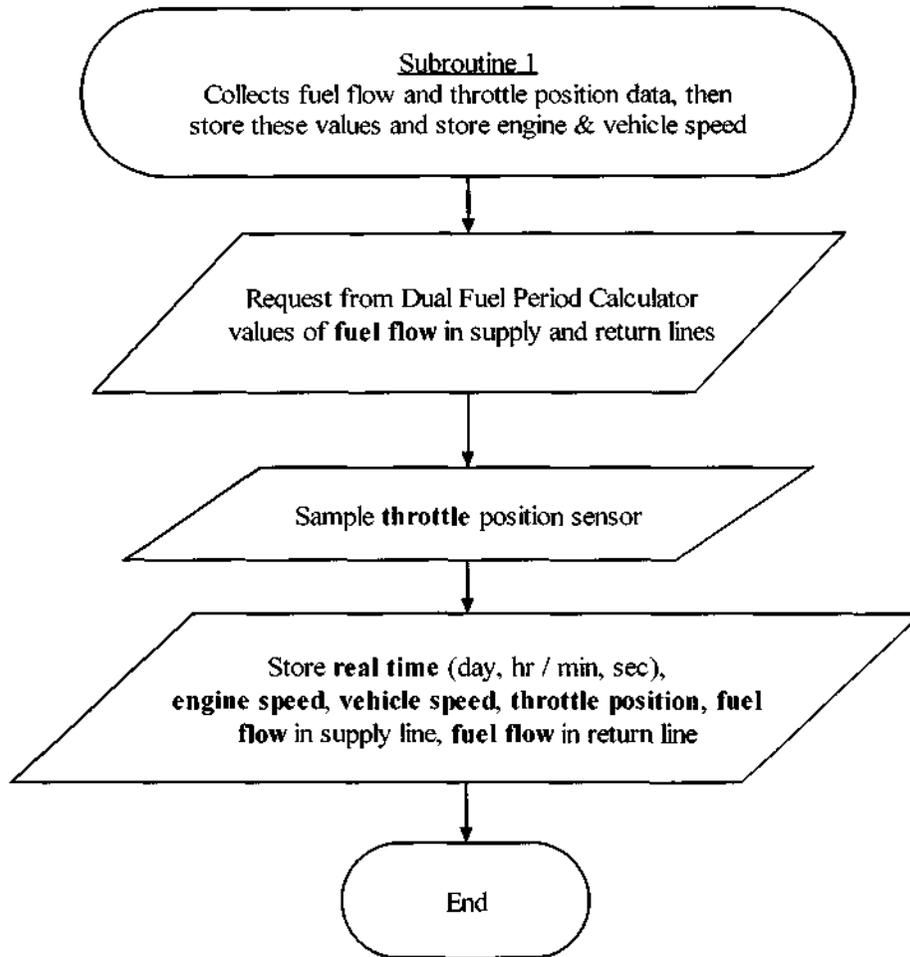


Figure G.2 Flowchart illustrating the algorithm for Subroutine 1 of the CR10X program

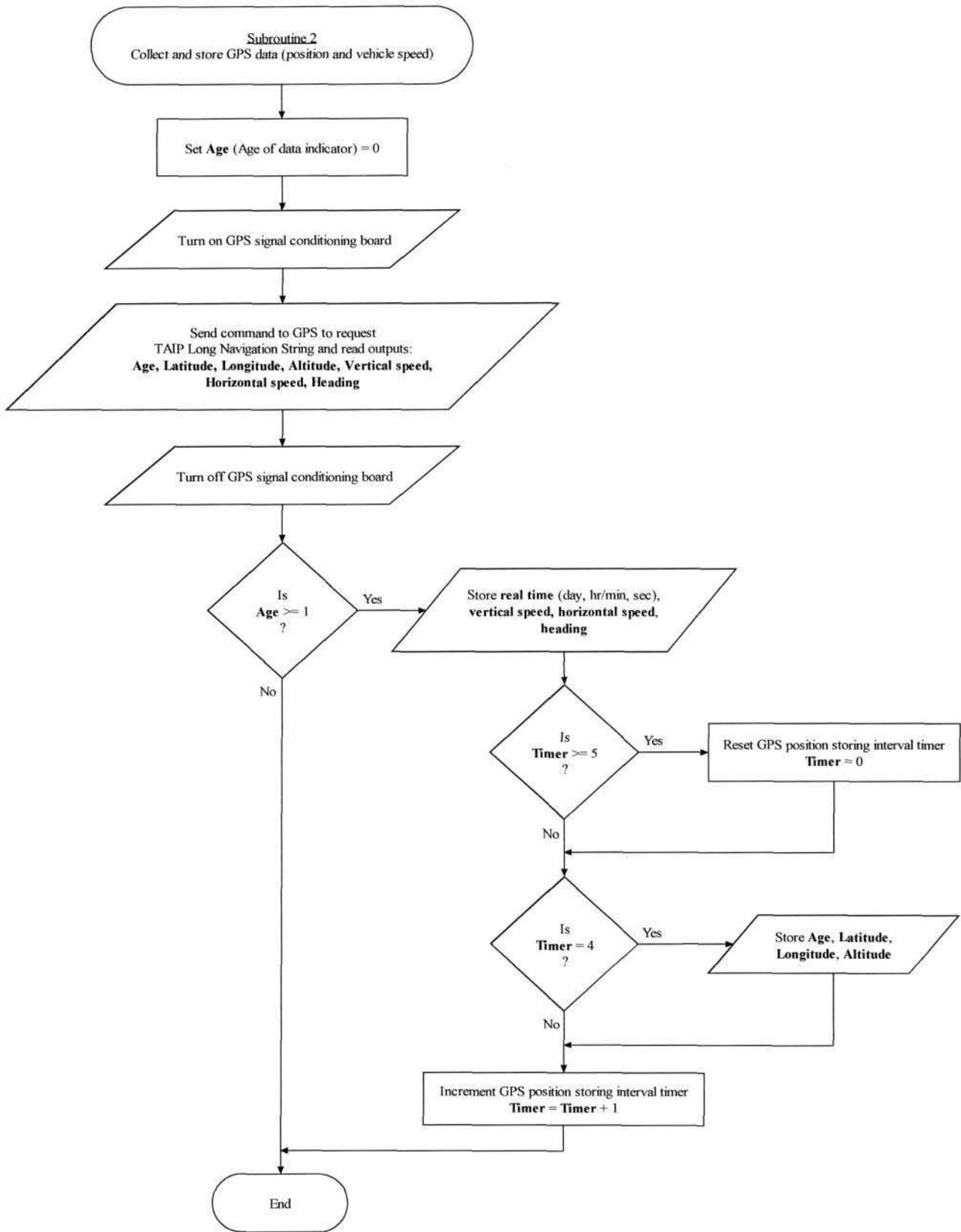


Figure G.3 Flowchart illustrating the algorithm for Subroutine 2 of the CR10X program

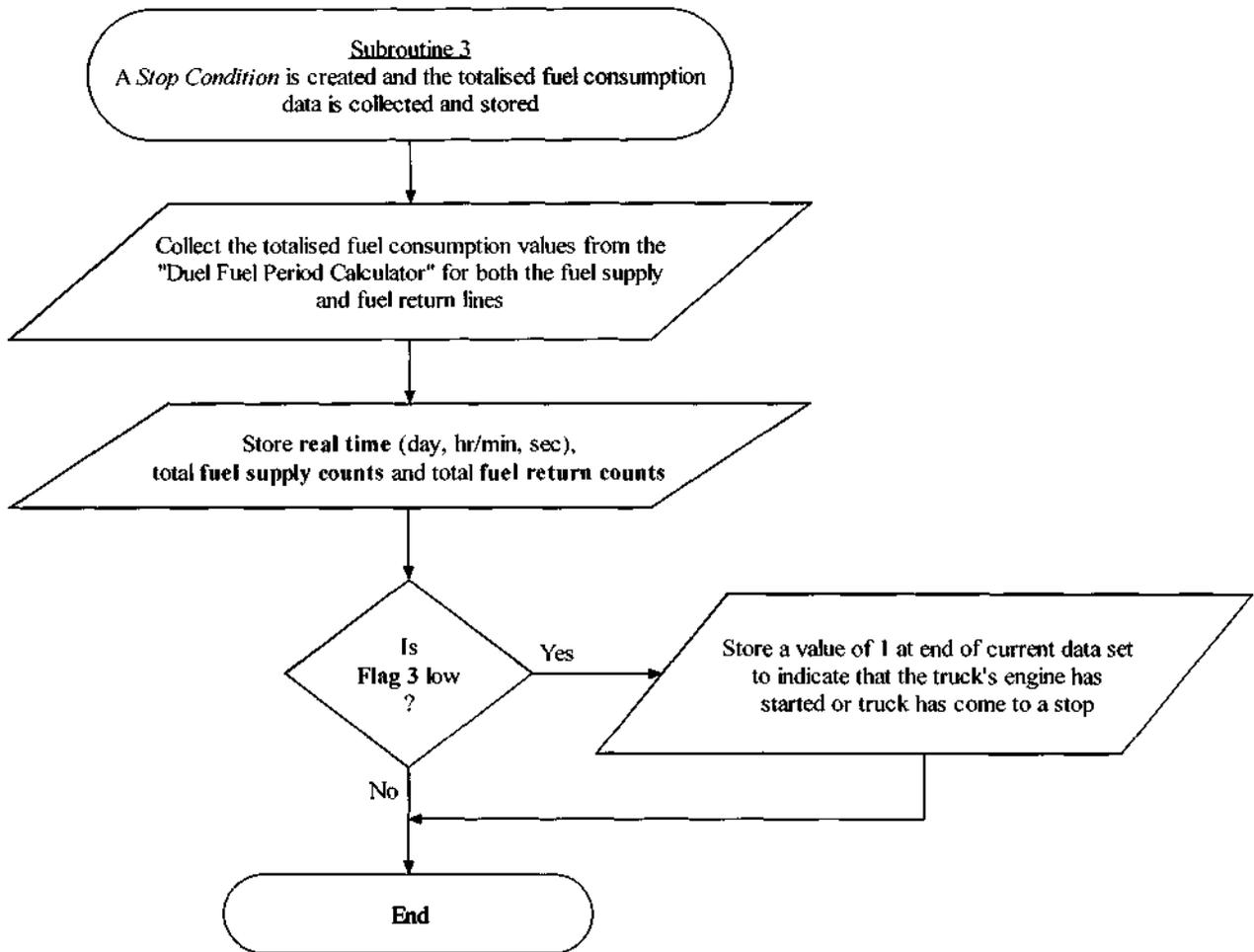


Figure G.4 Flowchart illustrating the algorithm for Subroutine 3 of the CR10X program

Figure G.5 CR10X program printout

{Note – lines beginning with “;” indicate a comment which is not processed by the CR10X}

```
;{CR10X}
;*****
;* University of Natal, School of Bioresources Engineering and Environmental Hydrology *
;* Program written by R Herbert and R van Zyl *
;* *
;* This program is used to collect data from a truck - engine & vehicle speed, fuel *
;* consumption, throttle position,real time,GPS data describing the position of the truck: *
;* latitude, longitude, *
;* age of data indicator (quality), altitude, vertical and horizontal speed (with heading) *
;* *
;*****
;
*Table 1 Program
  01: 1.5      Execution Interval (seconds)

;Engine speed is measured
1: Pulse (P3)
  1: 1      Reps
  2: 2      Pulse Channel 2
  3: 11     Low Level AC, Counts (Discard Bad Intervals)
  4: 2      Loc [ RPM      ]
  5: 1      Mult
  6: 0.0    Offset

;Vehicle speed is measured
2: Pulse (P3)
  1: 1      Reps
  2: 1      Pulse Channel 1
  3: 11     Low Level AC, Counts (Discard Bad Intervals)
  4: 3      Loc [ Speed    ]
  5: 1      Mult
  6: 0.0    Offset

;Conditional program run checking: For Diagnostic purposes one needs to disable checks, so
; program runs unconditionally. To disable checks set Flag 1 high
3: If Flag/Port (P91)
  1: 21     Do if Flag 1 is Low
  2: 30     Then Do

      ;If the truck is not running (engine r/min < 0 + delta), Flag 2 is set low
      4: If (X<=>F) (P89)
        1: 2      X Loc [ RPM      ]
        2: 4      <
        3: 2      F
        4: 22     Set Flag 2 Low

      ;When the truck is started (i.e. there is an engine speed), Flag 2 is set high
      5: If (X<=>F) (P89)
        1: 2      X Loc [ RPM      ]
        2: 3      >=
        3: 2      F
        4: 12     Set Flag 2 High

      ;If the vehicle is stationary, Flag 3 is set low (check for Vehicle speed < 0 + delta)
      6: If (X<=>F) (P89)
        1: 3      X Loc [ Speed    ]
        2: 4      <
        3: 0.5    F
        4: 23     Set Flag 3 Low

      ;Now check if the vehicle has ground speed - If vehicle is moving set Flag 3 high
      7: If (X<=>F) (P89)
        1: 3      X Loc [ Speed    ]
        2: 3      >=
        3: 0.5    F
        4: 13     Set Flag 3 High

;Ends run checks
8: End (P95)
```

```

;Executes if Flag 2 high - i.e. there is an r/min (engine is on)
9: If Flag/Port (P91)
  1: 12      Do if Flag 2 is High
  2: 30      Then Do

      ;Executes when truck is moving
10: If Flag/Port (P91)
  1: 13      Do if Flag 3 is High
  2: 30      Then Do

      ;Saves stop condition when vehicle pulls off
11: If (X<=>F) (P89)
  1: 7       X Loc [ Counter ]
  2: 1       =
  3: 1       F
  4: 3       Call Subroutine 3

      ;For fuel consumption calculation, and storing of r/min, speed & fuel data
12: Do (P86)
  1: 1       Call Subroutine 1

      ;To collect and store GPS data:
13: Do (P86)
  1: 2       Call Subroutine 2

;Clears the stop counter (the counter allows the stop condition to be recorded only once)
14: Z=F (P30)
  1: 0       F
  2: 0       Exponent of 10
  3: 7       Z Loc [ Counter ]

;Ends Flag 3 high check
15: End (P95)

;If engine running but the vehicle has stopped moving - record stop/start condition. Recorded
;as array ID 144, the day & time are then saved
16: If Flag/Port (P91)
  1: 23      Do if Flag 3 is Low
  2: 30      Then Do

      ;The time when the vehicle stops is saved to mem, but only once per stop.
      ;The fuel consumption counter is also saved
      ;The incremented counter is only cleared if the vehicle pulls off
17: If (X<=>F) (P89)
  1: 7       X Loc [ Counter ]
  2: 4       <
  3: 1       F
  4: 30      Then Do

      ;Saves time and totalised fuel consumption when vehicle stops
18: Do (P86)
  1: 3       Call Subroutine 3

;Ends stop condition check (if counter < 1 (but = 0) check)
19: End (P95)

;Set the stop counter = 1 so the stop condition is not saved again
20: Z=F (P30)
  1: 1       F
  2: 0       Exponent of 10
  3: 7       Z Loc [ Counter ]

;Reset Timer
21: Z=F (P30)
  1: 0       F
  2: 0       Exponent of 10
  3: 31      Z Loc [ Timer ]

;End Flag 3 low check (for stop condition)
22: End (P95)

;Ends Flag 2 high check
23: End (P95)

24: If Flag/Port (P91)
  1: 22      Do if Flag 2 is Low
  2: 30      Then Do

```

```

;Clears counter monitoring stop condition
25: Z=F (P30)
   1: 0      F
   2: 0      Exponent of 10
   3: 7      Z Loc [ Counter  ]

;set Flag 4 high when finished downloading data via modem and are ready to terminate call
26: If Flag/Port (P91)
   1: 14     Do if Flag 4 is High
   2: 30     Then Do

           ;a relay switches off power to the modem for 3s
27: Excitation with Delay (P22)
   1: 2      Ex Channel
   2: 300    Delay W/Ex (units = 0.01 sec)
   3: 0000   Delay After Ex (units = 0.01 sec)
   4: 2500   mV Excitation

           ;Flag 4 is set low again
28: Do (P86)
   1: 24     Set Flag 4 Low

29: End (P95) ;Flag 4 high check

;Load Command to request turn off TAIP Position Velocity String >FPV0<
;(When the GPS is powered up, sometimes the GPS receiver outputs the Position Velocity PV
;string every second, so the command >FPV0< is sent to cancel PV string being transmitted
30: Bulk Load (P65)
   1: 62     F ;>
   2: 70     F ;F
   3: 80     F ;P
   4: 86     F ;V
   5: 48     F ;0
   6: 60     F ;<
   7: 0      F ;
   8: 0      F ;
   9: 13     Loc [ QLN_1  ]

;Send string >FPV0< to turn off PV message from GPS - no characters are sent back
31: Port Serial I/O (P15)
   1: 1      Repts
   2: 1      ASCII/RS-232, 1200 Baud
   3: 1      Delay (0.01 sec units) before TX
   4: 1      First Control Port
   5: 13     Output Loc [ QLN_1  ]
   6: 6      No. of Locs to Send
   7: 42     Termination Character
   8: 0      Maximum Characters
   9: 100    Time Out for CTS (TX) and/or RX (0.01 sec units)
  10: 20     Loc [ Age      ]
  11: 1.0    Mult
  12: 0.0    Offset

;Flag 2 low check
32: End (P95)

;End of Table 1

*Table 2 Program
 02: 1800    Execution Interval (seconds)

;This section table is only executed every 30 minutes
;This table is merely to check the battery voltage

1: Batt Voltage (P10)
  1: 1      Loc [ Battery  ]

;Set output flag and write data to Final Storage in storage module
2: Do (P86)
  1: 10     Set Output Flag High (Flag 0)

3: Set Active Storage Area (P80)
  1: 1      Final Storage Area 1
  2: 101    Array ID

4: Real Time (P77)
  1: 110    Day,Hour/Minute (midnight = 0000)

```

```

;Stores the value of batt volt
5: Sample (P70)
  1: 1      Reps
  2: 1      Loc [ Battery  ]

; End of the battery voltage collection part of code (Table 2)

*Table 3 Subroutines

1: Beginning of Subroutine (P85)
  1: 1      Subroutine 1

; This subroutine collects fuel consumption data and throttle pos The subroutine
; then stores the engine speed, vehicle speed, throttle pos and fuel consumption

; The following section of the program collects serial data from the "dual fuel period
; calculator". The CR10X requests fuel data by setting C4 high for 10ms. Fuel Calculator
; waits lms. This is to allow the CR10X to prepare for receipt of data. Fuel data is
; transmitted. 2 values of 4 characters each are received on C5. The two values are comma
; delimited. A <CR> terminates the data line. A 100ms time-out allows the program to
; continue if the termination character is not received or less than 12 characters
; are received.

  2: Port Serial I/O (P15)
    1: 1      Reps
    2: 0      8-Bit, TTL ASCII, 1200 Baud
    3: 1      Delay (0.01 seconds)
    4: 4      First Control Port
    5: 12     Output Loc [ PulseDur  ]
    6: 1      No. of Locs to Send
    7: 13     Termination Character
    8: 48     Maximum Characters
    9: 25     Time Out for CTS (TX) and/or RX (0.01 sec units)
; 250ms time out if no characters received (data will equal -99999)
    10: 4     Loc [ Fuel_in  ]
    11: 0.01  Mult ;          Convert from 1/100 s to s : /10
    12: 0.0   Offset

; The period inputs from the "dual fuel period calculator" can be converted to flow rate
; (ml/min) by post-processing. This done by inverting the fuel inputs, subtracting fuel out
; from fuel in and then multiplying by the calibration factor of 2.5 ml/pulse to get ml/s
; Conversion: pulse/s * ml/pulse = ml/s

  3: Excite-Delay (SE) (P4)
    1: 1      Reps
    2: 15     2500 mV Fast Range
    3: 1      SE Channel
    4: 1      Excite all reps w/Exchan 1
    5: 1      Delay (units 0.01 sec)
    6: 2500   mV Excitation
    7: 6      Loc [ Throttle  ]
    8: 1.0    Mult
    9: 0.0    Offset

; Set output flag and write data to Final Storage
4: Do (P86)
  1: 10     Set Output Flag High (Flag 0)

5: Set Active Storage Area (P80)
  1: 1      Final Storage Area 1
  2: 11     Array ID

6: Real Time (P77)
  1: 111    Day,Hour/Minute,Seconds (midnight = 0000)

; Saves RPM, speed, fuel flow and throttle
7: Sample (P70)
  1: 5      Reps
  2: 2      Loc [ RPM      ]

; Ends Subroutine 1
8: End (P95)

```

```

9: Beginning of Subroutine (P85)
  1: 2      Subroutine 2

; Subroutine 2 is used to collect the GPS data
;
; This section of the CR10X Program uses the P15 instruction with SV8 plus Trimble Navigation
; GPS receiver. GPS outputs age of data (quality indicator), position and speed of antenna.
;
; The SV8 string is passed through a GPS black box which converts and delimitates the TAIP
; string from the GPS such that the string is useable by the CR10X.
;
; The logger sends the command to request the TAIP Long Navigation string >QLN<. The output
; string from the GPS is processed by the GPS black box which then passes the string back to
; the logger, which then in turn reads the data into the respective input locations
;
; The speed data from the GPS (horiz, vert speed & heading) is stored every 1.5 s (the exec
; interval of table 1). However, the position data from the GPS is only stored every 6s
;
; Wiring:
; GPS Power:Red to 12V, Black to GND, Yellow to 5V.
; Yellow is optional, used to backup SV8 memory
; GPS RS232: Cable has a 1K ohm resistor in
; series with the GPS Tx line, which is white
; Wire Black to C2, White to C3, clear to GND
;
; Form of LN message (from GPS black box) (Input locations used are on left)
;28 Age (1 digit) - 2 = fresh, 1 = old but still useable, 0 = no GPS signal do not use
;29 Lat_Deg (4 digit) - Lat Deg (pos north) + 1st 2 dec places (no comma separating dec) /100
;30 Long_Deg (4 digit) - Long Deg + 1st 2 dec places (no comma separating dec) (pos east) /100
;31 Alt (4 digit) - Altitude (above MSL in ft)
;32 Alt_Dec (2 digit) - decimal places of alt (in ft) /100
;33 Horiz_Spd (4 digit) - Horizontal speed of antennae (in mph) /10
;34 Heading (4 digit) - Horizontal heading in deg true north (inc eastwardly) of antennae /10
;35 Vert_Spd (5 digit) - Vertical speed of antennae (in mph) (required to be stored in high
res) /10
;36 Lat_Dec (5 digit) - Last 5 dec places of lat (required to be stored in high res) /10^5
;37 Long_Dec (5 digit) - Last 5 dec places of long (required to be stored in high res) /10^5
;* - termination character (dec equivalent = 42)
;

;Load Command to request TAIP Long Navigation String >QLN<
;However, when the GPS is powered up, sometimes the GPS receiver outputs the Position
;Velocity (PV) string every second, so the command >FPV0< is sent with the LN string to
;prevent the PV string from being transmitted
10: Bulk Load (P65)
  1: 62      F ;>
  2: 81      F ;Q
  3: 76      F ;L
  4: 78      F ;N
  5: 60      F ;<
  6: 0       F ;
  7: 0       F ;
  8: 0       F ;
  9: 13      Loc [ QLN_1      ]

;The GPS Age of Data Indicator location is stored in the last bulk load slot - as the above
;command writes a 0 over the age value - to clear the previous age value to prevent erroneous
;values accepted for age data

;Turns on GPS Black Box - save power and prevent the box from locking up if it receives a GPS
;sting other than LN
11: Do (P86)
  1: 48      Set Port 8 High

;Request TAIP Long Navigation String
12: Port Serial I/O (P15)
  1: 1       Reps
  2: 1       ASCII/RS-232, 1200 Baud
  3: 1       Delay (0.01 sec units) before TX
  4: 1       First Control Port
  5: 13      Output Loc [ QLN_1      ]
  6: 5       No. of Locs to Send
  7: 42      Termination Character
  8: 100     Maximum Characters
  9: 100     Time Out for CTS (TX) and/or RX (0.01 sec units)
  10: 20     Loc [ Age      ]
  11: 1.0    Mult
  12: 0.0    Offset

```

```

;Turns off GPS black box
13: Do (P86)
  1: 58      Set Port 8 Low

;Only write data if GPS Data is valid - i.e. Age Data Indicator >= 1 (2 = fresh)
14: IF (X<=>F) (P89)
  1: 20      X Loc [ Age      ]
  2: 3       >=
  3: 1       F
  4: 30      Then Do

;Set output flag and write data to Final Storage
15: Do (P86)
  1: 10      Set Output Flag High (Flag 0)

;Array id 21 is used for the gps info
16: Set Active Storage Area (P80)
  1: 2       Final Storage Area 2
  2: 21      Array ID

17: Real Time (P77)
  1: 111     Day,Hour/Minute,Seconds (midnight = 0000)

;Store GPS velocity data
18: Sample (P70)
  1: 3       Reps
  2: 25      Loc [ Horiz_Spd ]

;Clear timer after 6s so can count to next interval
19: If (X<=>F) (P89)
  1: 31      X Loc [ Timer    ]
  2: 3       >=
  3: 5       F
  4: 30      Then Do

  20: Z=F (P30)
    1: 0      F
    2: 0      Exponent of 10
    3: 31     Z Loc [ Timer    ]

21: End (P95)

;Only save GPS position info very 6s
22: If (X<=>F) (P89)
  1: 31      X Loc [ Timer    ]
  2: 1       =
  3: 4       F
  4: 30      Then Do

;Store GPS position data at end of gps velocity data

  23: Sample (P70)
    1: 5      Reps
    2: 20     Loc [ Age      ]

;Due to memory limitations with CR10X, must save the data with 5 digits in high res
  24: Resolution (P78)
    1: 1      High Resolution

  25: Sample (P70)
    1: 2      Reps
    2: 28     Loc [ Lat_Dec  ]

;Dont need to set Low Res here as this is the end of table 1 - when next table
;executes, the CR10X program defaults back to low res output data

;End of GPS pos data storage
26: End (P95)

;Increase timer
27: Z=Z+1 (P32)
  1: 31      Z Loc [ Timer    ]

;Ends if statement (quality control) &thus ends code section where data transferred to final
storage
  28: End (P95)

;End Subroutine 2
29: End (P95)

```

```

30: Beginning of Subroutine (P85)
  1: 3      Subroutine 3

;Subroutine 3 collects the "total number of pulses" counter values from the fuel calculator
;for the Supply and Return fuel flow meters.
;The values are 8 digits long and are split into 2 input locations each starting at Sup_Low
;(9999) then Sup_High (0000 9999) etc. The counters in the fuel computer are not reset , but
;left to roll over when they reach 9999 9999. The counters current values are downloaded when
;the vehicle starts to move and when it stops.

31: Do (P86)
  1: 47      Set Port 7 High

32: Port Serial I/O (P15)
  1: 1      Reps
  2: 0      8-Bit, TTL ASCII, 1200 Baud
  3: 1      Delay (0.01 seconds)
  4: 4      First Control Port
  5: 12     Output Loc [ PulseDur ]
  6: 1      No. of Locs to Send
  7: 13     Termination Character
  8: 96     Maximum Characters
  9: -1     Wait for RX Termination Character or Max RX Characters
; 250ms time out if no characters received (data will equal -99999)
10: 8      Loc [ Sup_Low ]
11: 1      Mult
12: 0.0    Offset

33: Do (P86)
  1: 57      Set Port 7 Low

34: Do (P86)
  1: 10     Set Output Flag High (Flag 0)

35: Set Active Storage Area (P80)
  1: 1      Final Storage Area 1
  2: 144    Array ID

;The time the vehicle stops is recorded
36: Real Time (P77)
  1: 111    Day,Hour/Minute,Seconds (midnight = 0000)

37: Resolution (P78)
  1: 1      High Resolution

; store the current values in the fuel counter registers
38: Sample (P70)
  1: 4      Reps
  2: 8      Loc [ Sup_Low ]

39: Resolution (P78)
  1: 0      Low Resolution

;Save value of 1 on end of "stop condition" string to indicate the vehicle is stopping or when
;engine started
40: If Flag/Port (P91)
  1: 23     Do if Flag 3 is Low
  2: 30     Then Do

          ;Set PulseDur = 1
41: Z=F (P30)
  1: 1      F
  2: 0      Exponent of 10
  3: 12     Z Loc [ PulseDur ]

42: Sample (P70)
  1: 1      Reps
  2: 12     Loc [ PulseDur ]

43: End (P95)

;Ends Subroutine 3
44: End (P95)

End Program

```

APPENDIX H – SPECIFICATIONS FOR MERCEDES BENZ 2637

Table H.1 Mercedes Benz 2637 V Series specifications

| | |
|-----------------------------|-----------|
| Vehicle Tare | 9.64 t |
| Trailers Tare | 11.31 t |
| Maximum GVM | 52.40 t |
| Engine Type | ADE 443N |
| Engine Displacement | 18.848 l |
| Governed Speed Range | 200 r/min |

| Fuel consumption regression equation coefficients | |
|--|----------|
| $C_1 =$ | 1.78E+00 |
| $C_2 =$ | 1.96E-06 |
| $C_3 =$ | 2.80E-06 |
| $C_4 =$ | 2.19E-09 |
| $C_5 =$ | 3.03E-09 |
| $C_6 =$ | 9.95E-06 |

| Gear Box Type: Mercedes Benz 16K-160 | |
|---|--------------|
| Gear Box efficiency | 90% |
| Gear change time | 0.75 s |
| Gear | Ratio |
| 1st | 13.68 |
| 2nd | 11.64 |
| 3rd | 9.40 |
| 4th | 8.00 |
| 5th | 6.73 |
| 6th | 5.73 |
| 7th | 4.79 |
| 8th | 4.07 |
| 9th | 3.36 |
| 10th | 2.86 |
| 11th | 2.31 |
| 12th | 1.96 |
| 13th | 1.85 |
| 14th | 1.41 |
| 15th | 1.18 |
| 16th | 1.00 |

| | | |
|------------------------------|---------------------|------------------------|
| Rear Axle Type | Planet Hub Red. | |
| Rear Axle Ratio | 4.205 | |
| Drive Axle efficiency | 90% | |
| Frontal Area | 6.07 m ² | |
| Coefficient of Drag | 0.85 | |
| Tyres | Front | Single 315 / 80 R 22.5 |
| | Rear | Dual 315 / 80 R 22.5 |
| Tyre Rolling radius | 549 mm | |

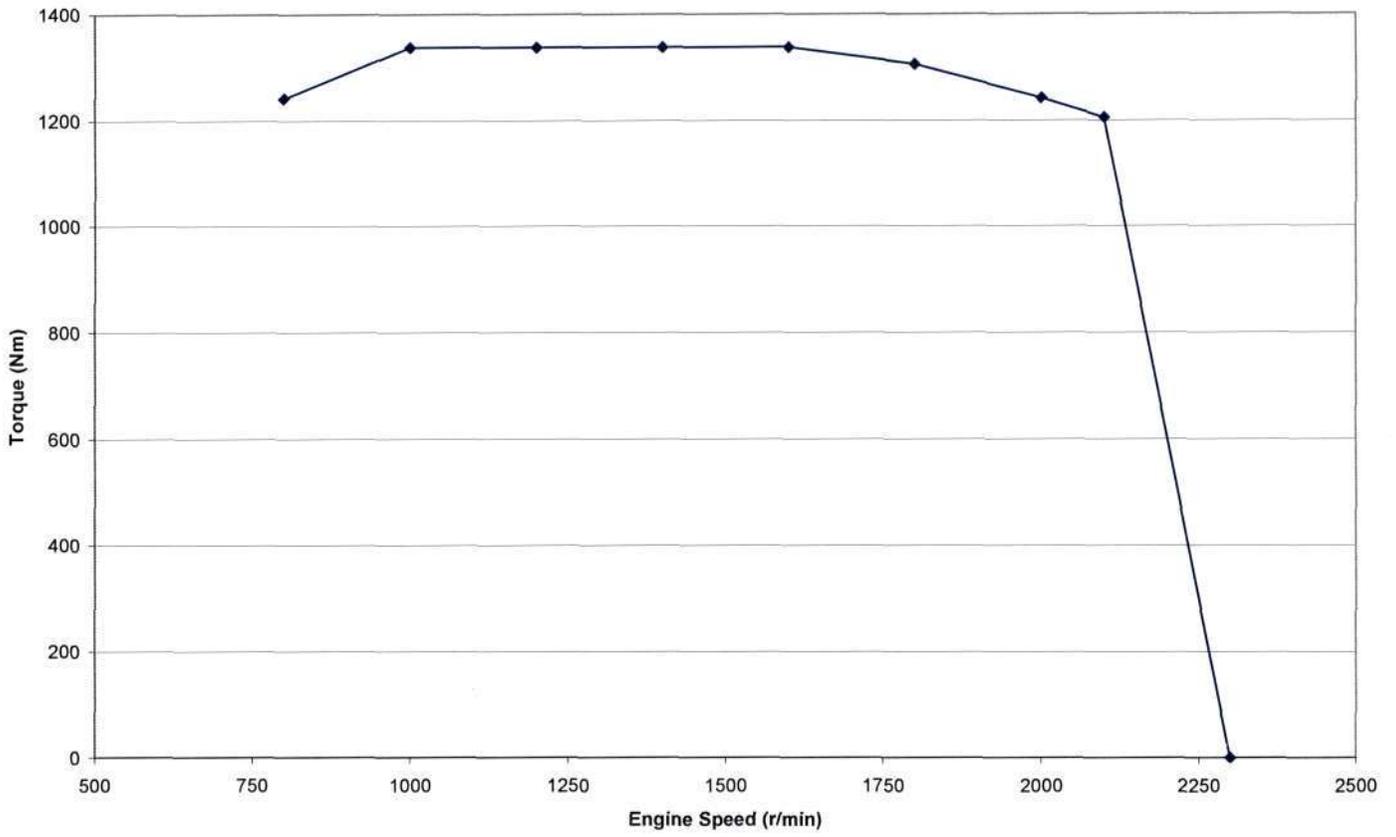


Figure H.1 Torque versus engine speed for ADE 443N engine

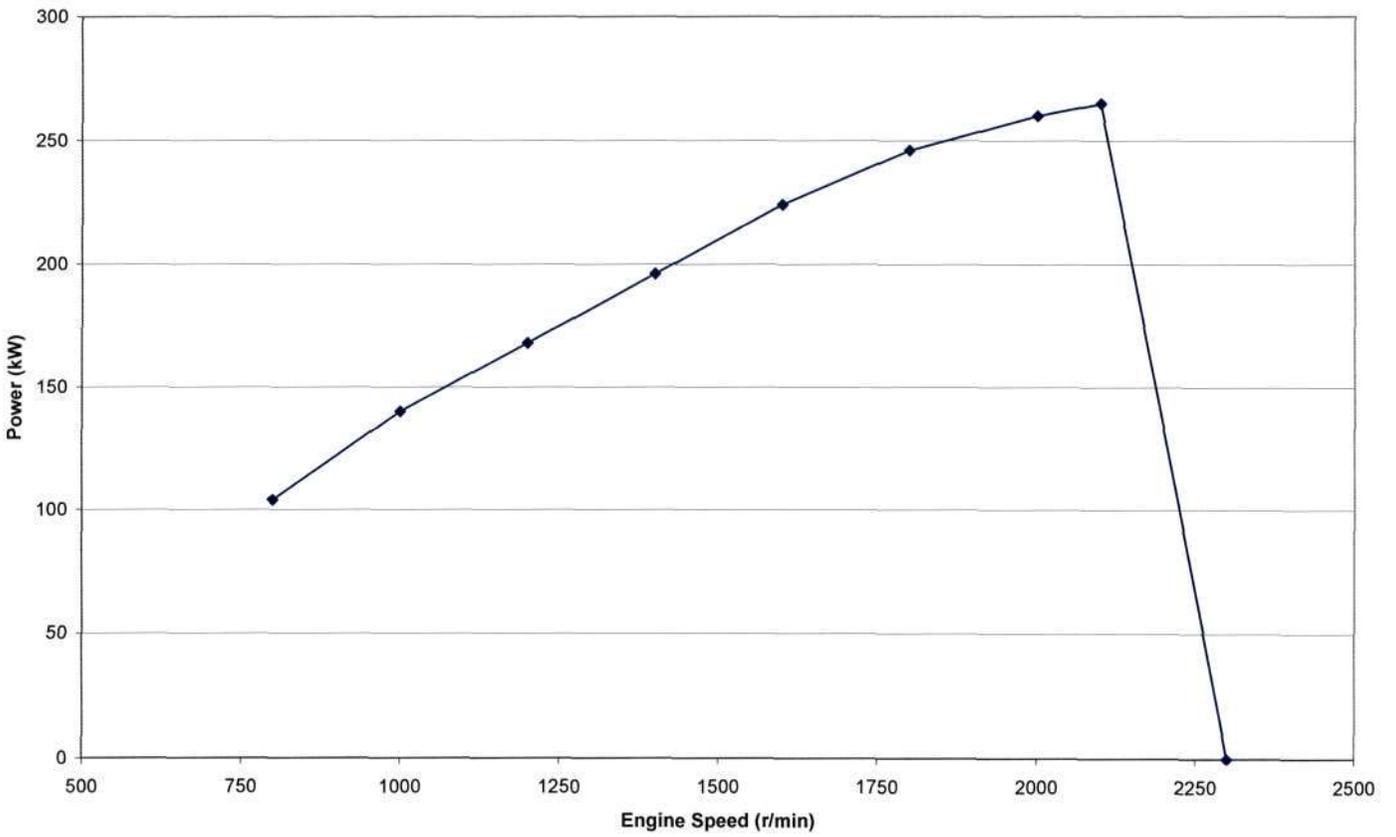


Figure H.2 Power versus engine speed for ADE 443N engine

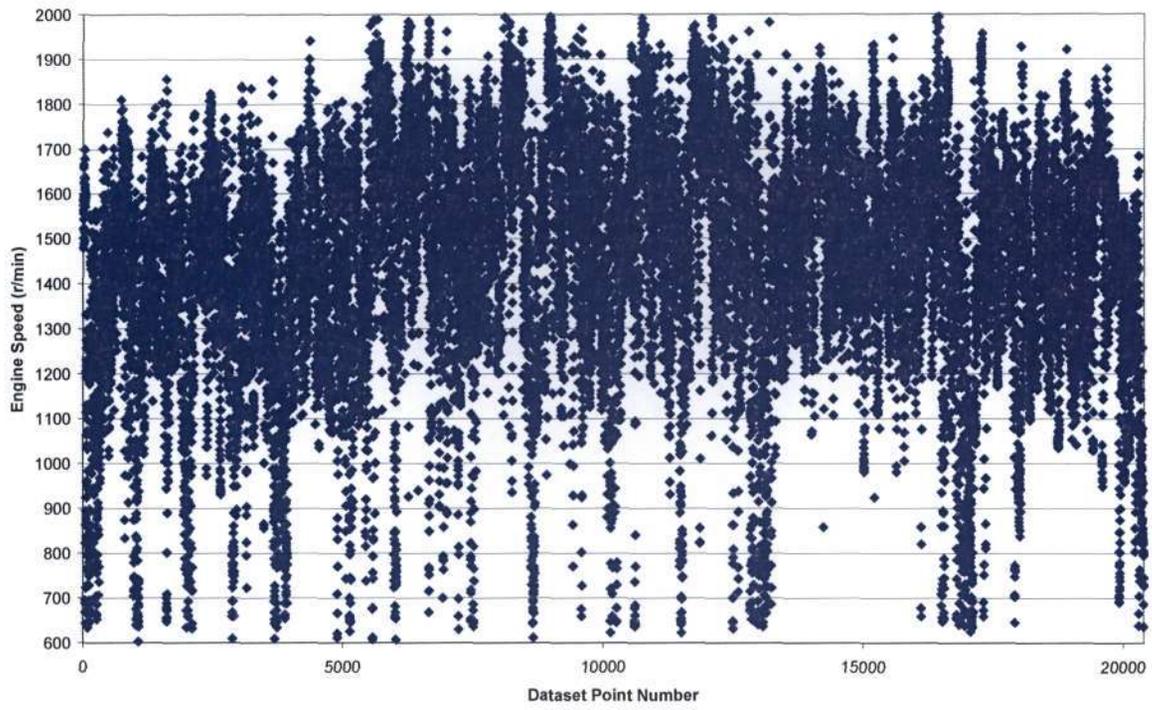


Figure H.3 Plot of the dataset of observed engine speed versus data point number of the engine speed values, as they appear in the observed dataset

APPENDIX I – FLOWCHARTS FOR MODIFIED SIMTRANS MODEL

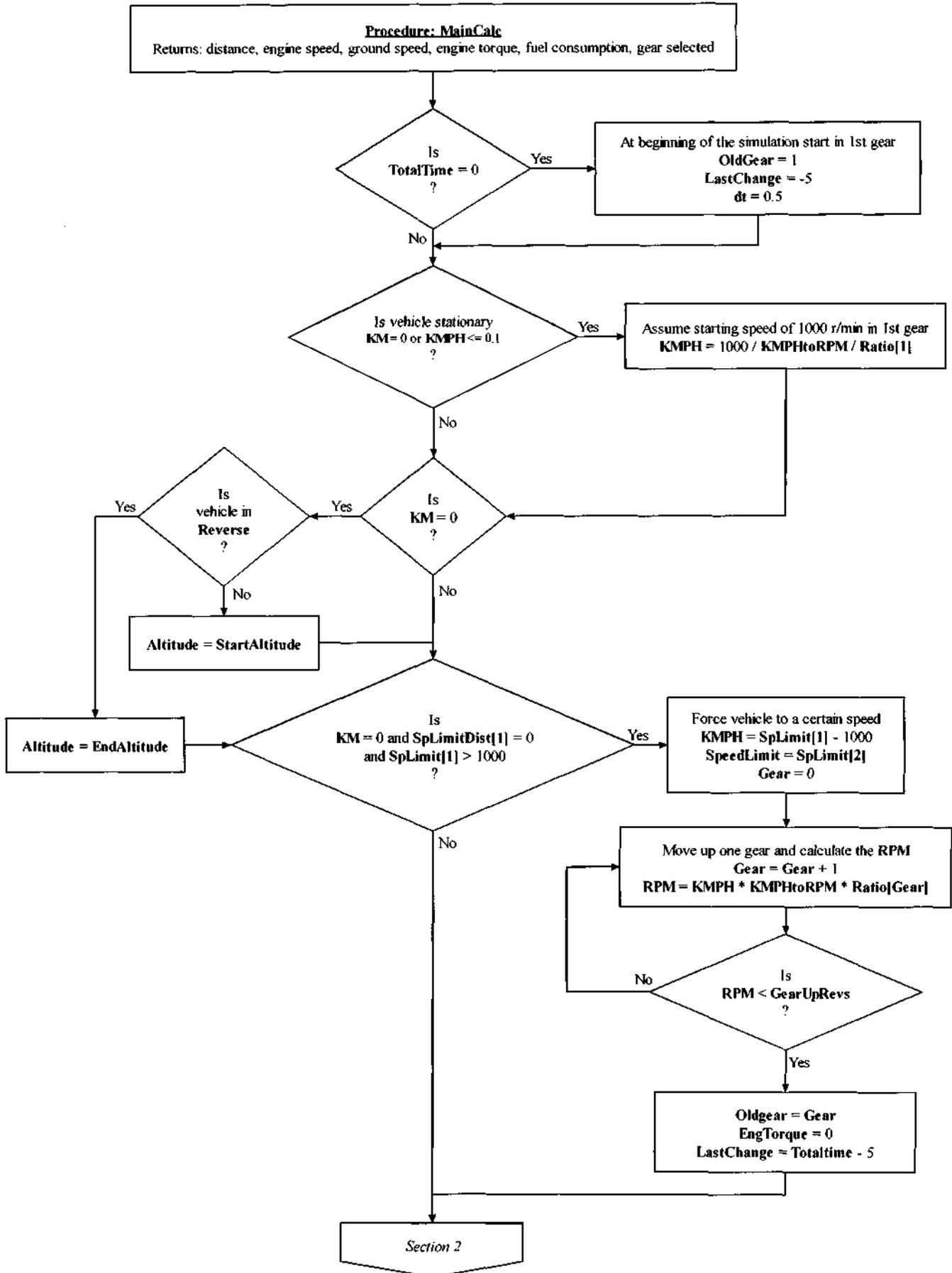


Figure I.1.1 Flowchart illustrating the algorithm for the modified Main procedure

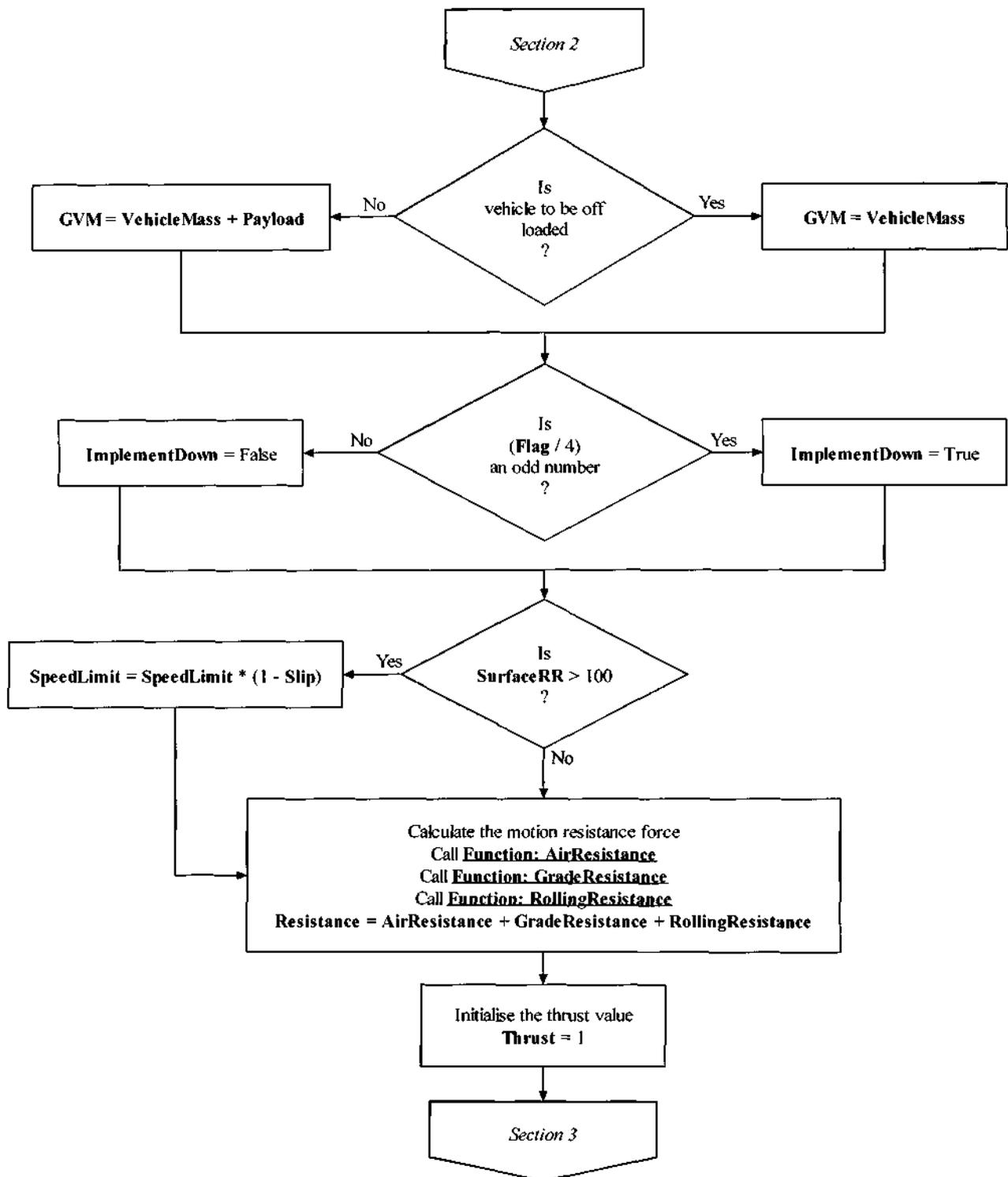


Figure I.1.2 Flowchart illustrating the algorithm for the modified Main procedure

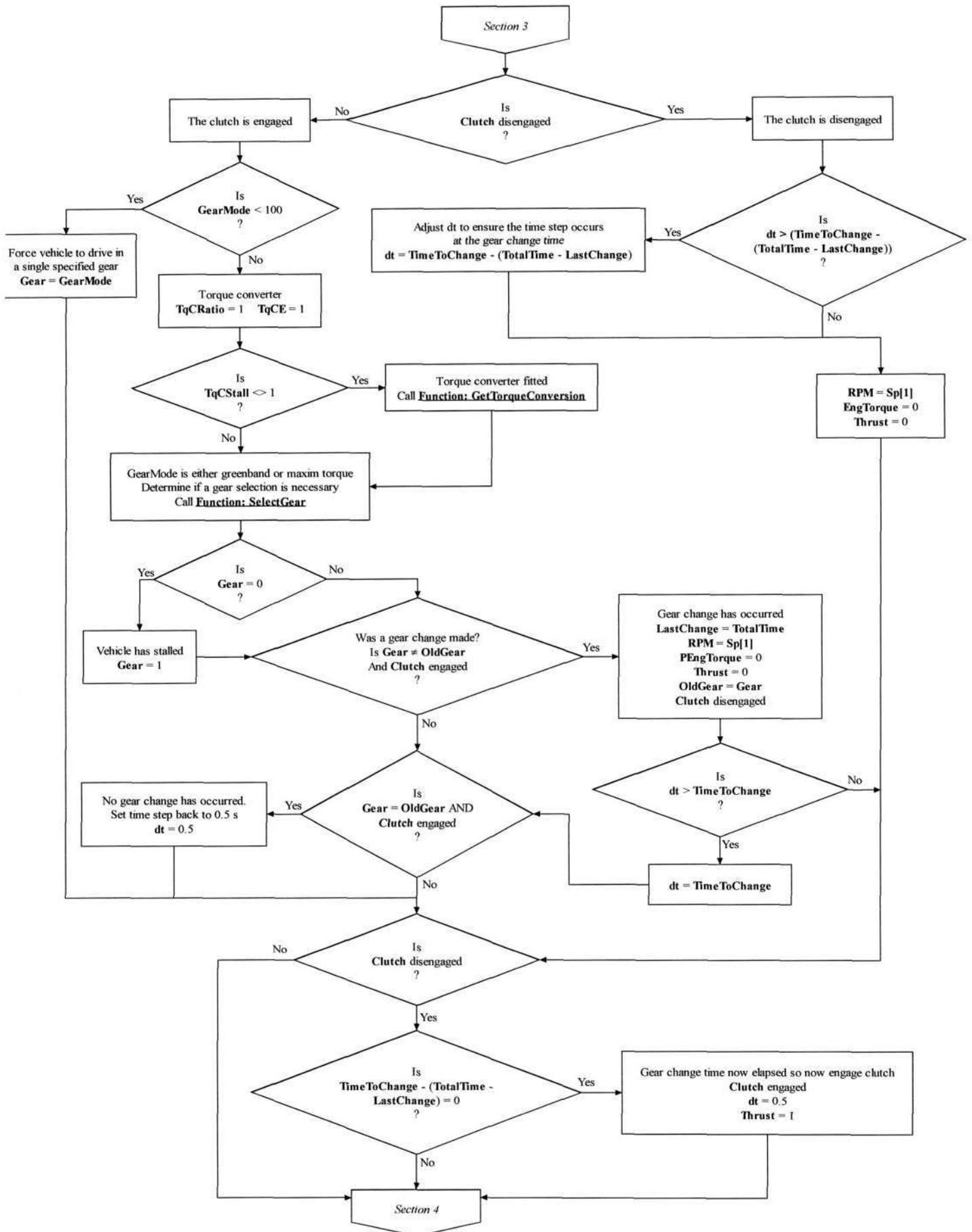


Figure I.1.3 Flowchart illustrating the algorithm for the modified Main procedure

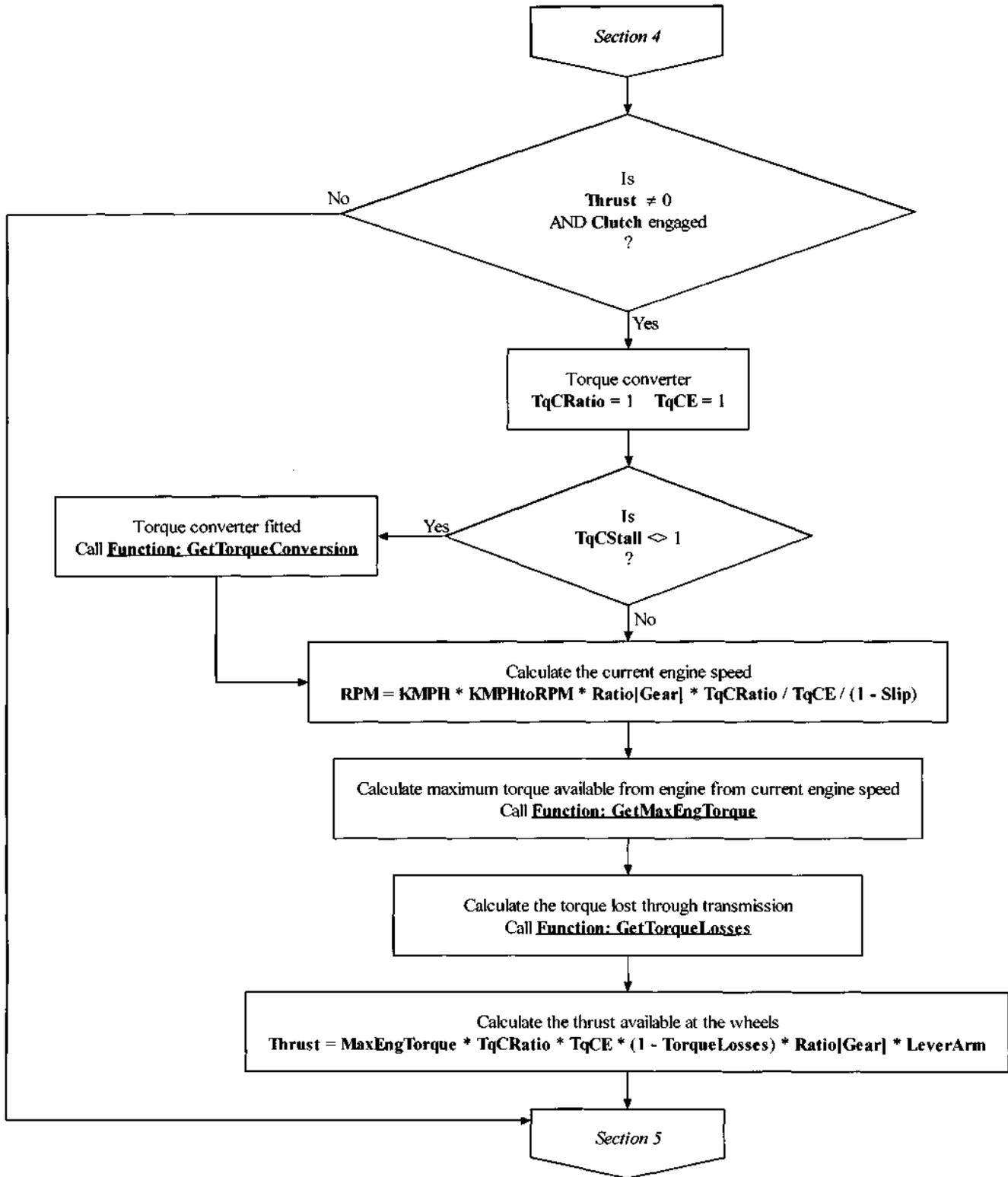


Figure I.1.4 Flowchart illustrating the algorithm for the modified Main procedure

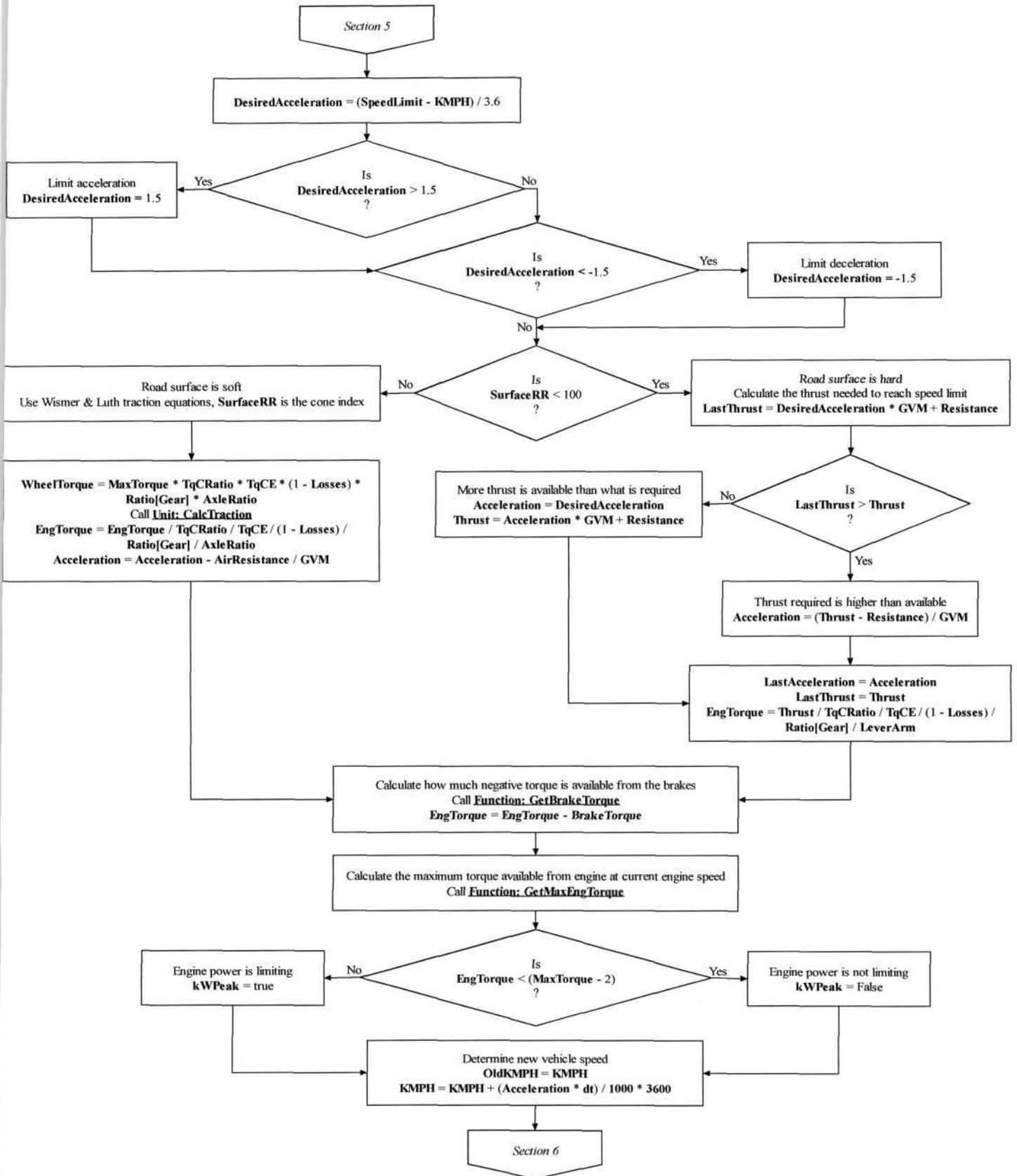


Figure I.1.5 Flowchart illustrating the algorithm for the modified Main procedure

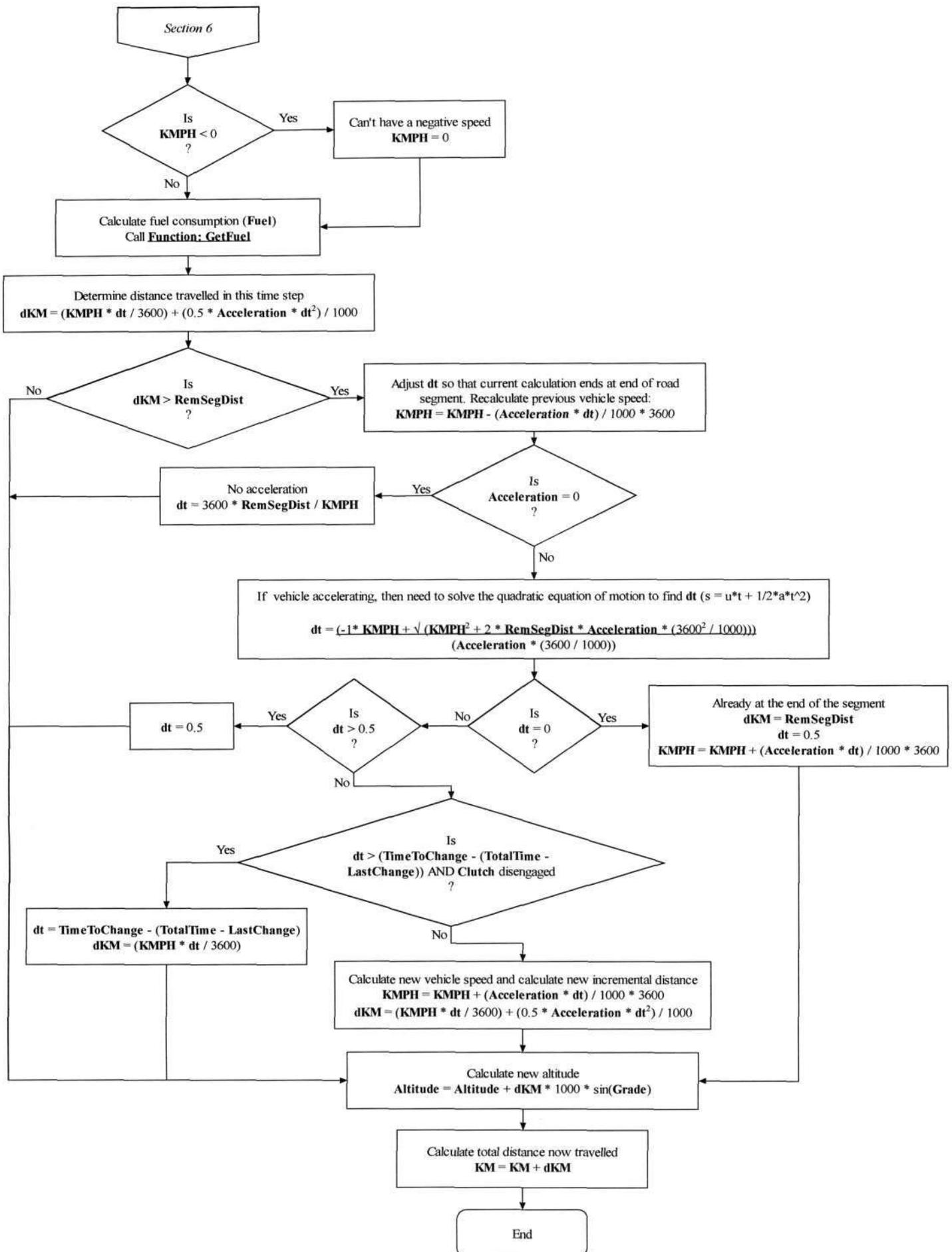


Figure I.1.6 Flowchart illustrating the algorithm for the modified Main procedure

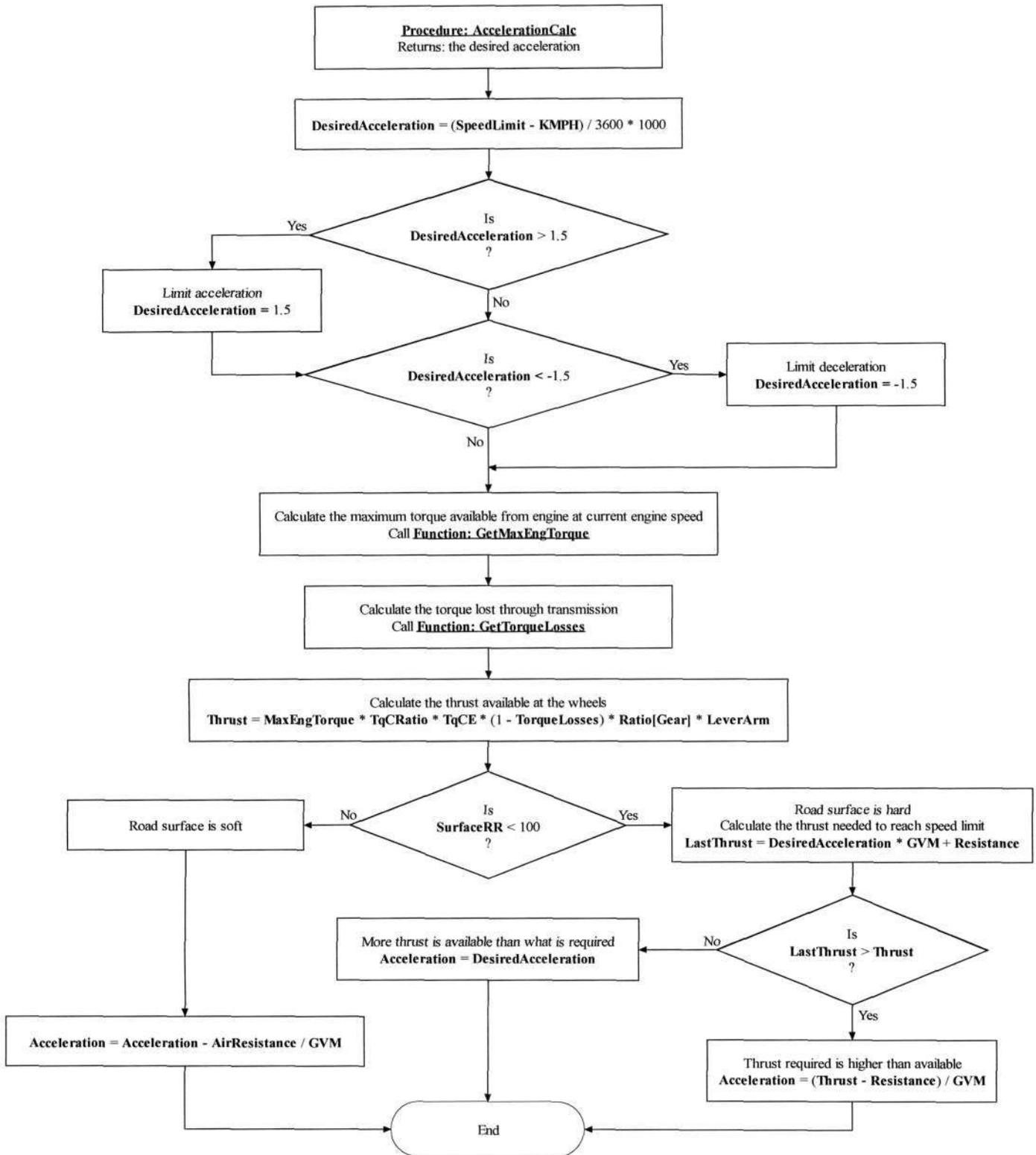


Figure I.2 Flowchart illustrating the algorithm for the AccelerationCalc procedure

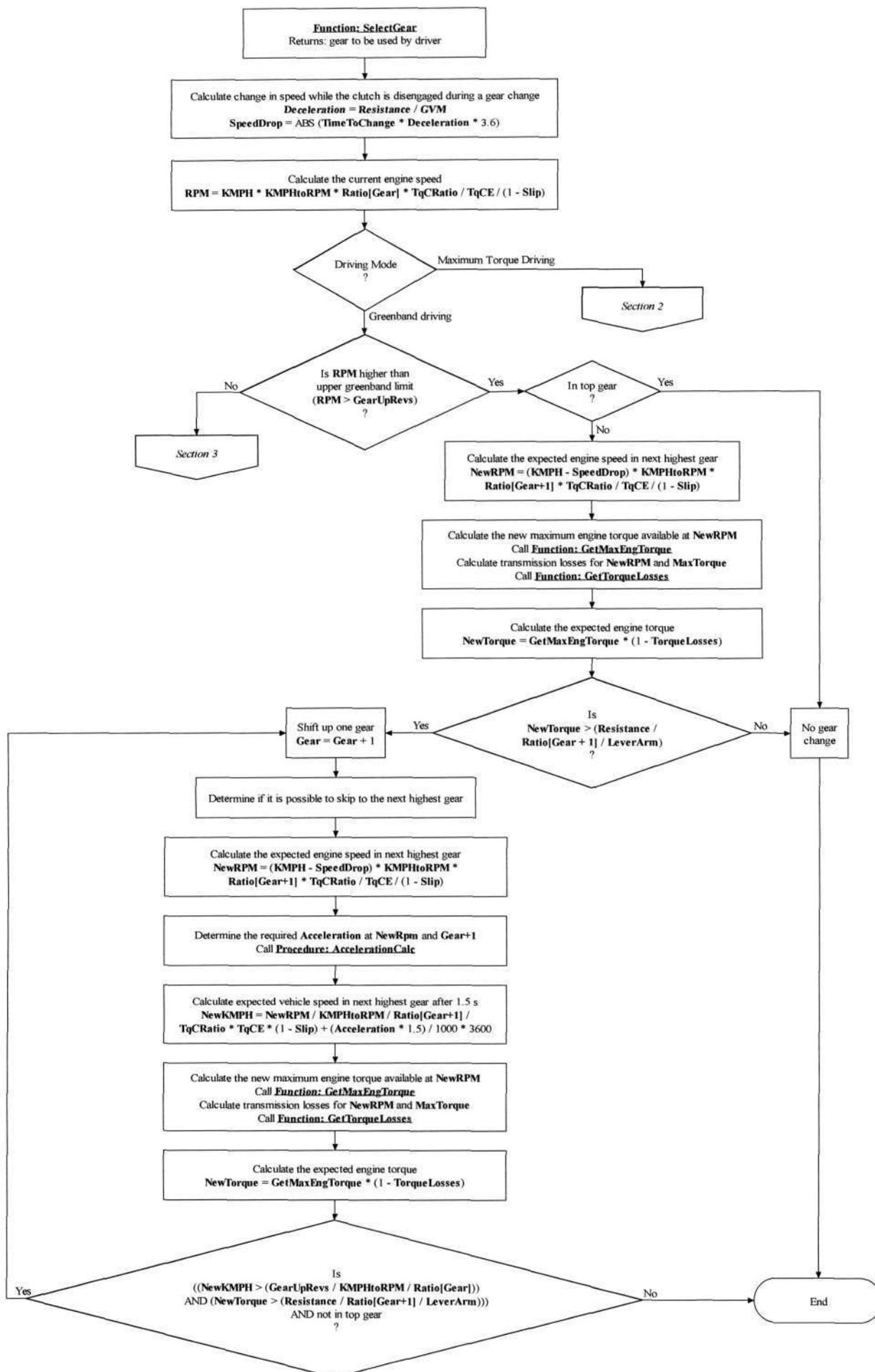


Figure I.3.1 Flowchart illustrating the algorithm for the modified SelectGear function

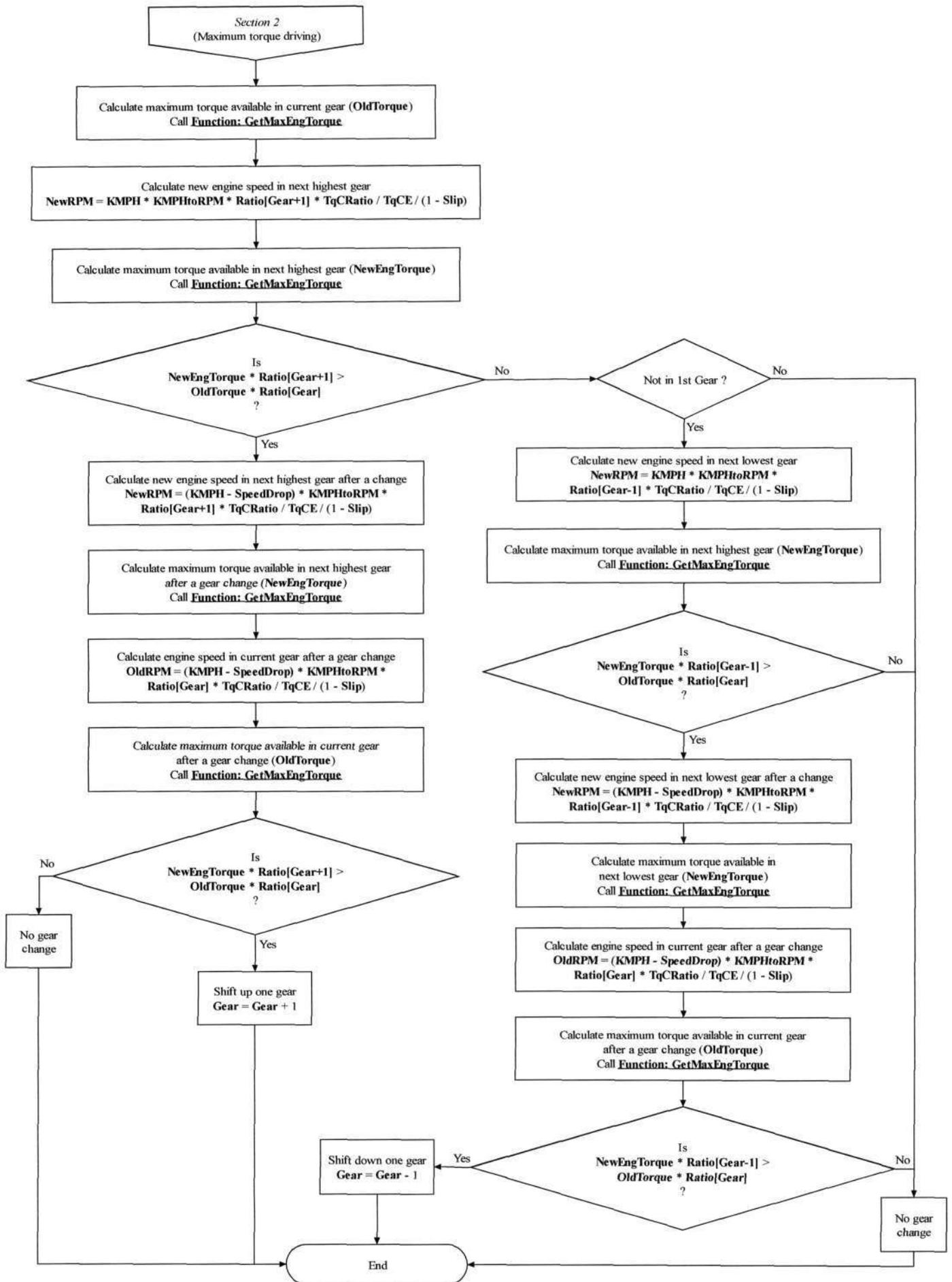


Figure I.3.2 Flowchart illustrating the algorithm for the modified SelectGear function

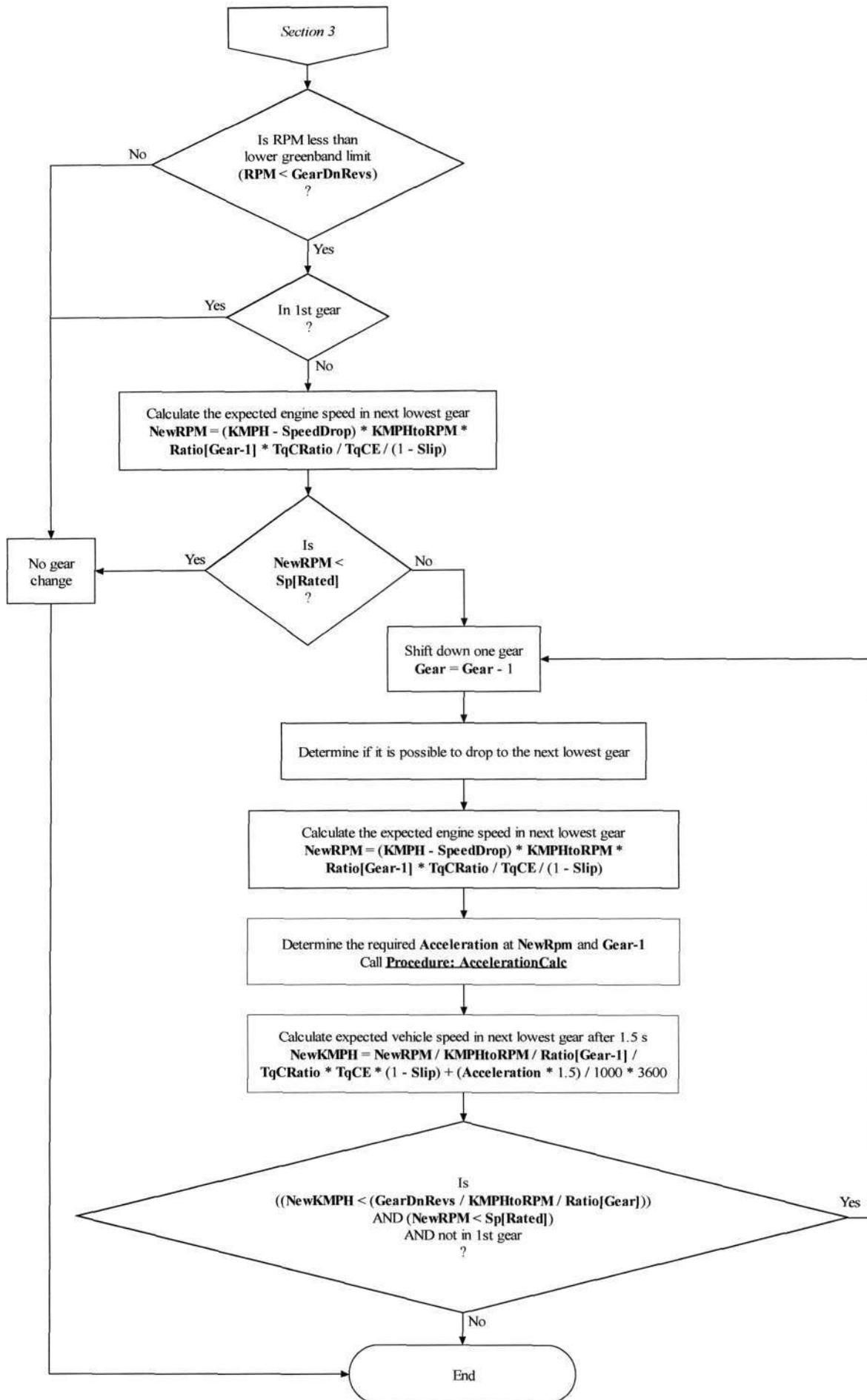


Figure I.3.3 Flowchart illustrating the algorithm for the modified SelectGear function