OXYGEN DEPLETION SHUTDOWN ALGORITHM
FOR PORTABLE GASOLINE GENERATORS

By

JENNIFER BEASLEY SMELSER

A THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in the Department of Electrical Engineering in the Graduate School of The University of Alabama

TUSCALOOSA, ALABAMA

2009
ABSTRACT

Portable gasoline generators are very useful during power outages caused by snow storms, hurricanes, tornadoes, and earthquakes. If not used outdoors and away from the home, the consequences can be deadly. Due to the fatality rate caused by the poisonous carbon monoxide (CO) gas emitted from portable gasoline generators, the need for CO reduction and automatic generator shutdown has increased. This thesis presents the Engine Management System (EMS) used to reduce CO emissions from a commercial off-the-shelf Coleman Powermate 7000 Watt generator. Although reduced emissions was accomplished, there was still a need for an automatic shutdown feature in the event the generator was operated in an enclosed environment.

Having operated in this type of environment, the generator’s poisonous exhaust depletes the environment’s oxygen (O₂) content causing the intake air temperature (IAT), fuel pulse width (FPW), and fuel pulse width correction of the engine to change. The O₂ depletion shutdown algorithm uses these variables as inputs in order to calculate a pseudo-derivative based upon a moving average for each input. A comparison is performed on each pseudo-derivative and if all three have met unacceptable limits for a predetermined period of samples, the engine will be shut down. The algorithm implemented in the engine control module (ECM) was tested under various loading conditions and proven to be successful under all loading conditions. This thesis presents the algorithm and test results from The University of Alabama and the National Institute of Standards and Technology.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFR</td>
<td>Air-Fuel Ratio</td>
</tr>
<tr>
<td>BLM</td>
<td>Block Learn Memory</td>
</tr>
<tr>
<td>CAT</td>
<td>Charge Air Temperature</td>
</tr>
<tr>
<td>CLC</td>
<td>Closed-Loop Control</td>
</tr>
<tr>
<td>CLTS</td>
<td>Oil Temperature Sensor</td>
</tr>
<tr>
<td>CO</td>
<td>Carbon Monoxide</td>
</tr>
<tr>
<td>CPSC</td>
<td>Consumer Product Safety Commission</td>
</tr>
<tr>
<td>CTB</td>
<td>Calibration Toolbox</td>
</tr>
<tr>
<td>ECM</td>
<td>Engine Control Module</td>
</tr>
<tr>
<td>EFI</td>
<td>Electronic Fuel Injection</td>
</tr>
<tr>
<td>EMS</td>
<td>Engine Management System</td>
</tr>
<tr>
<td>FPW</td>
<td>Fuel Pulse Width</td>
</tr>
<tr>
<td>Hz</td>
<td>Hertz</td>
</tr>
<tr>
<td>HC</td>
<td>Hydrocarbons</td>
</tr>
<tr>
<td>IAT</td>
<td>Intake Air Temperature</td>
</tr>
<tr>
<td>IFR</td>
<td>Injector Flow Rate</td>
</tr>
<tr>
<td>MAP</td>
<td>Manifold Absolute Pressure</td>
</tr>
<tr>
<td>$m_{\text{air}}$</td>
<td>Mass of Air</td>
</tr>
<tr>
<td>$m_{\text{fuel}}$</td>
<td>Mass of Fuel</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NO\textsubscript{x}</td>
<td>Nitrogen Oxide</td>
</tr>
<tr>
<td>O\textsubscript{2}</td>
<td>Oxygen</td>
</tr>
<tr>
<td>P</td>
<td>Pressure</td>
</tr>
<tr>
<td>PI</td>
<td>Proportional-Integral</td>
</tr>
<tr>
<td>R</td>
<td>Air Gas Constant</td>
</tr>
<tr>
<td>RPM</td>
<td>Speed</td>
</tr>
<tr>
<td>T</td>
<td>Temperature</td>
</tr>
<tr>
<td>TPS</td>
<td>Throttle Position Sensor</td>
</tr>
<tr>
<td>UA</td>
<td>University of Alabama</td>
</tr>
<tr>
<td>V</td>
<td>Volume</td>
</tr>
<tr>
<td>VE</td>
<td>Volumetric Efficiency</td>
</tr>
<tr>
<td>W</td>
<td>Watts</td>
</tr>
</tbody>
</table>
CONTENTS

ABSTRACT ...........................................................................................................................................ii

LIST OF ABBREVIATIONS ...........................................................................................................iii

LIST OF FIGURES .......................................................................................................................vii

LIST OF TABLES ...........................................................................................................................x

1  Introduction ...................................................................................................................................1

2  The Engine Management System .................................................................................................6

  2.1  Overall ECM Description .......................................................................................................10

     2.1.1 IMEC-168 ECM Description ...........................................................................................12

     2.1.2 MT05 ECM Description ..................................................................................................14

3  Oxygen Depletion Testing ............................................................................................................16

  3.1  Oxygen Depletion Test House ...............................................................................................17

  3.2  $O_2$ House Testing and Results ............................................................................................18

     3.2.1 Open Door Test Results ..................................................................................................20

     3.2.2 Closed Door Test Results ...............................................................................................25

4  Oxygen Depletion Shutdown Algorithm .....................................................................................31

  4.1  Post Processing of Initial Data ...............................................................................................35

  4.2  Generator Tests with $O_2$ Depletion Shutdown Algorithm ..................................................48

5  NIST Testing ................................................................................................................................51

6  Conclusions ....................................................................................................................................58

  6.1  Future Work ...........................................................................................................................59
LIST OF FIGURES

2.1 Crank Position Sensor and 24x Toothed Wheel...........................................7
2.2 Diagram of Delphi EMS..............................................................................8
2.3 Coleman Powermate 7000 Watt generator equipped with EMS.................9
2.4 The IMEC-168 ECM mounted to the exterior of the generator..................13
2.5 The MT05 ECM mounted inside the generator frame.................................14
3.1 O\textsubscript{2} Depletion Test House-Exterior.............................................17
3.2 O\textsubscript{2} Depletion Test House-Exterior..................................................18
3.3 Measured sensor signals and O\textsubscript{2} vs. time, open-door, 0 W...............21
3.4 Fueling parameters and O\textsubscript{2} vs. time, open-door, 0 W....................21
3.5 Emissions data, open-door, 0 W.................................................................22
3.6 Measured sensor signals and O\textsubscript{2} vs. time, open-door, 1500 W.......22
3.7 Fueling parameters and O\textsubscript{2} vs. time, open-door, 1500 W...............23
3.8 Emissions data, open-door, 1500 W...........................................................23
3.9 Measured sensor signals and O\textsubscript{2} vs. time, open-door, 5500 W.........24
3.10 Fueling parameters and O\textsubscript{2} vs. time, open-door, 5500 W.............24
3.11 Emissions data, open-door, 5500 W.........................................................25
3.12 Measured sensor signals and O\textsubscript{2} vs. time, closed-door, 0 W...........26
3.13 Fueling parameters and O\textsubscript{2} vs. time, closed-door, 0 W...................27
3.14 Emissions data, closed-door, 0 W.............................................................27
3.15 Measured sensor signals and O\textsubscript{2} vs. time, closed-door, 1500 W.....28
3.16 Fueling parameters and O₂ vs. time, closed-door, 1500 W ................................. 28
3.17 Emissions data, closed-door, 1500 W ................................................................. 29
3.18 Measured sensor signals and O₂ vs. time, closed-door, 5500 W ....................... 29
3.19 Fueling parameters and O₂ vs. time, closed-door, 5500 W .............................. 30
3.20 Emissions data, closed-door, 5500 W ................................................................. 30
4.1 Post processed-measured signals, open-door, 0 W ............................................. 36
4.2 Post processed-fail hits, open-door, 0 W .............................................................. 36
4.3 Post processed-trip flags, open-door, 0 W ............................................................ 37
4.4 Post processed-shutdown signal, open-door, 0 W .............................................. 37
4.5 Post processed-measured signals, closed-door, 0 W .......................................... 38
4.6 Post processed-fail hits, closed-door, 0 W ........................................................... 38
4.7 Post processed-trip flags, closed-door, 0 W ........................................................ 39
4.8 Post processed-shutdown signal, closed-door, 0 W .......................................... 39
4.9 Post processed-measured signals, open-door, 1500 W .................................... 40
4.10 Post processed-fail hits, open-door, 1500 W ..................................................... 40
4.11 Post processed-trip flags, open-door, 1500 W .................................................. 41
4.12 Post processed-shutdown signal, open-door, 1500 W ..................................... 41
4.13 Post processed-measured signals, closed-door, 1500 W .................................. 42
4.14 Post processed-fail hits, closed-door, 1500 W ................................................. 42
4.15 Post processed-trip flags, closed-door, 1500 W ............................................... 43
4.16 Post processed-shutdown signal, closed-door, 1500 W ................................... 43
4.17 Post processed-measured signals, open-door, 5500 W ................................. 44
4.18 Post processed-fail hits, open-door, 5500 W ................................................... 44
4.19  Post processed-trip flags, open-door, 5500 W

4.20  Post processed-shutdown signal, open-door, 5500 W

4.21  Post processed-measured signals, closed-door, 5500 W

4.22  Post processed-fail hits, closed-door, 5500 W

4.23  Post processed-trip flags, closed-door, 5500 W

4.24  Post processed-shutdown signal, closed-door, 5500 W

4.25  O₂ depletion parameters, closed-door, 0 W

4.26  Algorithm fail counters and trip flag, closed-door, 0 W

4.27  O₂ depletion parameters, closed-door, 5500 W

4.28  Algorithm fail counters and trip flag, closed-door, 5500 W

5.1  NIST Indoor Air Quality Test House

5.2  O₂ depletion parameters, open-door, warm start, 5500 W

5.3  Algorithm fail counters and trip flag, open-door, warm start, 5500 W

5.4  O₂ depletion parameters, closed-door, warm start, 5500 W

5.5  Algorithm fail counters and trip flag, closed-door, warm start, 5500 W

5.6  O₂ depletion parameters, door open 24 in., warm start, 5500 W

5.7  Algorithm fail counters and trip flag, door open 24 in., warm start, 5500 W

5.8  O₂ depletion parameters, closed-door, warm start, 2500 W

5.9  Algorithm fail counters and trip flag, closed-door, warm start, 2500 W

5.10 O₂ depletion parameters, door open 24 in., warm start, 2500 W

5.11 Algorithm fail counters and trip flag, door open 24 in., warm start, 2500 W
LIST OF TABLES

2.1 Input and output signals of the ECM.........................................................6
3.1 Load Bank Load Points.............................................................................19
4.1 O₂ depletion algorithm variables...............................................................33
4.2 O₂ depletion algorithm process variables ..................................................34
Portable gasoline generators are primarily used when power is needed in remote locations or during periods of power outages caused by hurricanes, snow storms, etc. These generators emit poisonous carbon monoxide (CO) in their exhaust, so if not used properly they can be very deadly tools. CO is an odorless, colorless, and toxic gas that will cause headaches, dizziness, nausea, and fatigue depending on concentration level, length of exposure, and the person’s age and overall health [1]. Ultimately, CO poisoning will cause death if an individual is exposed to high concentration levels for a substantial period of time.

In 2007, the Consumer Product Safety Commission (CPSC) reported 334 non-fire CO poisoning deaths due to portable gasoline generators. Eighty percent of these deaths occurred in homes, including single-family homes, apartments, townhouses, mobile homes, and garages or sheds at homes [2]. As the number of CO poisoning related deaths associated with portable gasoline generators increases, the concern for reducing the generator’s CO emissions also increases. This concern led the CPSC to contract with The University of Alabama to reduce the CO emissions from portable gasoline generators and install an automatic shutdown feature. Although generators should never be run indoors, the low CO generator with automatic shutdown would increase the rate of survivability in the event it did occur.
Through literature searches in Society of Automotive Engineers, Institute of Electronics and Electrical Engineers, and other independent archival journals and proceedings, the author has determined that little literature directly related to the study of an automatic shutdown feature for portable gasoline generators exists. However, literature has been found on on-board diagnostics (OBD) of sensors and fault detection methods, which could be used to detect oxygen depletion scenarios, rather than sensor failures. Some of the current fault technology methods being used today are model based, knowledge based, signal based, and data based type strategies [9, 10]. Model based strategies determine faults through deviations between a theoretical model and the physical process, and knowledge based strategies determine when a fault has occurred by using prior knowledge of the physical process. A signal based strategy simply analyzes, or filters a signal to obtain further information that can be used to detect a fault. An example of this strategy is the detection of broken rotor bars in induction motors by monitoring the spectral components of the motor current produced by the resulting magnetic field disturbance [11]. However, this process is focused on detecting a parametric variation rather than a plausible operating condition with an active closed-loop control system. The data based strategy uses a neural network to program a process model that can be used to compare against the physical process to determine fault conditions. The primary focus for this project was to use a low CO portable gasoline generator for testing under oxygen depleted conditions, in order to create an automatic shutdown feature in the event a generator was operated in an enclosed environment.

The low CO generator project used a commercial off-the-shelf, Coleman Powermate 5500 Watt portable gasoline generator as its test platform. The generator
came equipped with a carbureted Honda GX390, single cylinder, four-stroke, 11 Hp engine. An Environmental Protection Agency (EPA) 6-mode emissions test was performed on the generator to establish a baseline emissions output before modifications were made to reduce the output. In order to reduce this output, the generator’s carburetor was replaced with an Engine Management System (EMS) and three-way catalyst. The purpose of the EMS was to reduce emissions without sacrificing engine performance and fuel economy, which can be achieved by controlling the amount of fuel delivered to the engine and the spark timing under various load conditions \[3\]. The three-way catalyst, which is most effective when the engine uses a stoichiometric fuel mixture, uses a formulation of platinum, palladium, and rhodium to reduce all three major emissions, Nitrogen Oxide (NO\textsubscript{x}), Hydrocarbons (HC), and CO. A stoichiometric fuel mixture is a ratio of the mass of air to the mass of fuel, 14.6 to 1 air-fuel ratio (AFR). The mixture has just enough air to completely burn the available fuel or complete the combustion process. If the fuel mixture is lower than 14.6 to 1 AFR, the mixture is considered to be rich, which improves the reduction of NO\textsubscript{x} at the expense of CO and HC. When the mixture is greater than 14.6 to 1 AFR, the mixture is considered to be lean, which improves the oxidation of CO and HC at the expense of NO\textsubscript{x} \[4, 5\].

The generator’s EMS includes a Delphi IMEC-168 Engine Control Module (ECM) and various sensors that monitor and control the engine’s operation. The IMEC-168 ECM was used to calibrate the generator’s engine for operating in closed-loop at a stoichiometric fuel mixture for all engine load points. Closed-loop operation uses the voltage signal provided by an oxygen sensor to determine if the air/fuel ratio is rich or lean, so that closed-loop control can compensate and drive the air/fuel ratio back to
stoichiometric. Due to the addition of the Delphi EMS and three-way catalyst, the output of CO emissions from the 5500 Watt generator was reduced by 97%, which exceeded the project goal of 90% reduction.

It was the goal of the efforts from creating, testing, and implementing an automatic shutoff feature for the Coleman Powermate 7000 Watt portable gasoline generator from which this thesis resulted. This generator, equipped with the same engine as the 5500 Watt generator, was originally calibrated with the Delphi IMEC-168 ECM to mimic the operation of the 5500 Watt generator. In order to complete the tasks of this thesis, the obsolete IMEC-168 ECM was replaced by Delphi’s MT05 ECM and sensors specific to the new system. The MT05 system, which will be covered in the following chapter of this thesis, provides increased capability over the previous system, user-friendly calibration tools, and the capability for future upgrades. Delphi provided training on the MT05 ECM, the Delphi ITS Calibration Toolbox (CTB), and engine calibration using the new system. Upon completion of Delphi training, the 7000 Watt generator was calibrated with the upgraded EMS and then tested under oxygen depleted conditions.

Initial tests, which will be discussed in Chapter 3, were performed on the generator to determine which engine variables and engine signals reacted to oxygen depleted conditions. It was determined that the intake air temperature (IAT), block learn memory (BLM) correction factor, and fuel pulse width (FPW) reacted significantly during the oxygen depletion tests. These variables were used to develop an algorithm that would shut down the generator after a specified threshold had been crossed. Upon post-processing the data through a simulation, it was determined for specific cases that
the engine would automatically. Due to the proprietary nature of Delphi’s system, the
data and Oxygen (O₂) depletion shutoff algorithm were sent to Delphi for implementation
in the MT05 software. The O₂ depletion shutoff algorithm and test results will be
covered in Chapter 4, and the final tests, which were performed in a more realistic
scenario at the National Institute of Standards and Technology (NIST) in Gaithersburg,
Maryland will be discussed in Chapter 5.

The goal of thesis was ultimately obtained by taking a 7000 Watt generator
already modified for low CO emissions and implementing a software-based automatic
shutoff feature. The shutoff feature works very well under medium and high load test
conditions and although success has been achieved at light low load conditions, the
amount of time it takes to shut off could be greatly improved.
CHAPTER 2  

The Engine Management System (EMS)

The EMS is designed to help meet exhaust and emissions requirements for small gasoline engines without sacrificing engine performance and fuel consumption [6]. The Delphi EMS for this application uses an ECM, ignition coil, electronic fuel injection (EFI), a fuel pump and pressure regulator, and various sensors to continuously monitor the engine’s performance. The ECM is an electronic device that collects data from engine sensors, performs calculations based on the received data, and provides output control to optimize engine operation under a variety of conditions. The following table shows a list of inputs and outputs to the Delphi ECM.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Input/Output</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxygen Sensor</td>
<td>Input</td>
<td>Analog</td>
</tr>
<tr>
<td>Intake Air Temperature</td>
<td>Input</td>
<td>Analog</td>
</tr>
<tr>
<td>Oil Temperature</td>
<td>Input</td>
<td>Analog</td>
</tr>
<tr>
<td>Battery Voltage</td>
<td>Input</td>
<td>Analog</td>
</tr>
<tr>
<td>Crank Position</td>
<td>Input</td>
<td>Pulse</td>
</tr>
<tr>
<td>Manifold Absolute Pressure</td>
<td>Input</td>
<td>Analog</td>
</tr>
<tr>
<td>Injector</td>
<td>Output</td>
<td>Pulse</td>
</tr>
<tr>
<td>Spark Timing</td>
<td>Output</td>
<td>Pulse</td>
</tr>
</tbody>
</table>
The ignition coil, EFI, fuel pump, and fuel pressure regulator allow for fuel delivery and spark timing to ignite the fuel. An oil temperature sensor is used to monitor the temperature of the oil, and an intake air temperature sensor is used to monitor the temperature of the air entering the engine. The signals from these sensors are used as look-up values for other parameters which determine engine operating modes. The crank position sensor and 24x toothed wheel, shown in Figure 2.1, are used by the ECM to determine speed and provide a crank position reference for reading the manifold absolute pressure (MAP), fuel delivery, and spark timing.

![Crank Position Sensor and 24x Toothed Wheel.](image)

Figure 2.1: Crank Position Sensor and 24x Toothed Wheel.

The crank sensor is excited by the 24x wheel, which provides a pulse train signal that can be used by the ECM to calculate speed. The 24x wheel is aligned on the engine so that the falling edge of the 9th tooth after the gap corresponds to the position of the piston at top dead center (TDC). The crank sensor uses the missing tooth on the wheel as a reference point to find TDC so the MAPReadCrankAngle can be read at the minimum
MAP point on the engine’s intake stroke. The MAP read crank angle can be found in a look-up table, based upon speed (RPM) and vacuum, in the calibration software.

Vacuum is simply the difference between the barometric pressure (BARO) and MAP.

The ECM reads the BARO only at key-on using the MAP sensor, since the MAP is equal to the BARO before the engine is started. A diagram of the EMS is shown below in Figure 2.2.

Figure 2.2: Diagram of Delphi EMS.
Two different Delphi engine controllers have been used during the course of this project, the IMEC-168 ECM and the MT05 ECM. The IMEC-168 ECM, now obsolete, was used for the original calibration and testing of the Honda Coleman Powermate 7000 W portable gasoline generator shown in Figure 2.3. This ECM along with a 3-way catalyst helped reduce the CO emissions output from this portable gasoline generator by 97%. Upon upgrading the ECM, the crank position sensor, oxygen sensor, and ignition coil specific to the old controller had to be replaced with MT05 system counterparts. The new controller and sensors have increased functionality over the previous system, more user-friendly calibration tools, and the capability for future upgrades.

Figure 2.3: Coleman Powermate 7000 Watt generator equipped with EMS.
2.1 Overall ECM Description

Although two different Delphi engine controllers were used for this project, the basic principles are the same. Both systems use a MAP based load variable and engine speed as primary inputs to control fuel delivery and spark timing. In order to deliver a balanced fuel mixture to the engine, the controller must calculate the amount of air entering the engine by means of the speed-density method. The volumetric efficiency (VE), shown in Equation 2.1, is defined as the actual air mass divided by the theoretical air mass in the cylinder and can be found in a look-up table, based upon RPM and MAP, in the calibration software.

\[ \eta_{VE} = \frac{\frac{m_{air/cycle}}{\rho_{ref} * V_D}}{V} \]  

Where engine displacement volume is \( V_D \) and \( \rho_{ref} \) is the reference density. The \( \rho_{ref} \), shown in Equation 2.3, is proportional to the pressure and inversely proportional to the temperature of the intake manifold, and can be calculated from the manifold conditions using the ideal gas law shown in Equation 2.2. [4].

\[ P * V = m * R * T \]  

\[ \rho_{ref} = \frac{m}{V} \]  

Where \( V \) is volume, \( P \) is pressure, \( T \) is temperature, \( m \) is the mass, and \( R \) is the air gas constant. By combining Equations 2.1 and 2.2, the \( \rho_{ref} \) can be calculated as shown in Equation 2.3, where \( P_{man} \) and \( T_{man} \) are pressure and temperature of the manifold:

\[ \rho_{ref} = \frac{P_{man}}{R * T_{man}} \]
Equation 2.5 calculates the mass of air ($m_{\text{air}}$) entering the engine by using Equations 2.1, which compensates for deviations in the ideal gas law, and Equation 2.3. Equation 2.5 can be related to the current system by the following: $P_{\text{man}} = \text{MAP (kPa)}$, $V_D = \text{cylinder volume, 389 (cc)}$, $T_{\text{man}} = \text{charge air temperature (CAT) (°C)}$, and $R = 287.05 \ (J/[kg*K])$. The CAT is an estimated temperature based upon RPM, MAP, IAT, and oil temperature (CLTS).

\[
m_{\text{air}} = \frac{P_{\text{man}} \cdot V_D \cdot \eta_{\text{VE}}}{R \cdot T_{\text{man}}} \tag{2.4}
\]

The controller then uses the calculated $m_{\text{air}}$ and the desired AFR, 14.6, to determine the mass of fuel ($m_{\text{fuel}}$) to be delivered to the engine. In order to deliver the $m_{\text{fuel}}$ calculated in Equation 2.5, the controller must use the injector flow rate (IFR), 1.4 (g/s), $m_{\text{fuel}}$, and other transient fuel parameters to calculate the FPW of the injector.

\[
m_{\text{fuel}/cycle} = \frac{m_{\text{air}/cycle}}{AFR} \tag{2.5}
\]

Another important feature of the ECM is closed-loop control (CLC), because it uses an oxygen sensor as feedback to control the AFR around stoichiometric, 14.6 to 1 AFR. Upon engine startup, the system will run in open-loop, which uses the VE and an open-loop AFR table, until the CLC enable temperature has been reached. Once CLC has been activated, the oxygen sensor provides the controller with the rich/lean status of the fuel mixture in the engine. CLC uses a correction factor and a proportional-integral (PI) controller to drive the AFR back towards 14.6 to 1 AFR, which will either increase or decrease the FPW based upon the rich/lean status of the fuel mixture. In order for the controller to keep the AFR near 14.6 to 1, the integral term will increase when the fuel mixture is lean and decrease when it is rich. The proportional term of this controller uses
the feedback from the oxygen sensor to toggle the status of the fuel mixture between rich and lean. These corrections are applied to the FPW, and the process starts all over again. Although the concept is the same for each controller, the logic and functionality are quite different. Brief descriptions of each controller covered in the subsequent sections of this chapter highlight those differences.

2.1.1 IMEC-168 ECM Description

This controller uses a MAP sensor internal to the ECM, which can be connected to the intake port of the engine using a small vacuum tube. The setback to this is that gasoline is occasionally drawn into the vacuum tube causing erratic MAP readings, which will ultimately cause unstable engine operation. The controller also uses a virtual throttle position (TPS) signal to identify different modes of engine operation, and since the engine does not use a TPS sensor, a TPS look-up table was created based upon RPM and MAP. The controller uses this table in conjunction with other tables and variables, so if not calibrated properly, the engine will begin to run in an unstable condition.

The fuel delivery logic for this controller uses $m_{\text{air}}$ and $m_{\text{fuel}}$ as calculated previously to calculate the FPW. The exception is that the FPW calculation must also account for CLC, battery state, and other transient fueling conditions. One major setback to this controller occurs when calibrating the engine and fine tuning the calibration look-up tables, most of which are based upon RPM and MAP, for optimized engine performance. The axes of these look-up tables are not tunable; meaning if the engine has more than one RPM/MAP break point, the engine must be tuned to meet in the middle of those points. This, in turn, sacrifices the ECM’s ability to control the engine as well as
the engine’s overall performance. Placement of the ECM is also an issue, since it has an internal MAP sensor. The ECM must be mounted to the generator in such a way that the MAP tube connected from the ECM to the engine is at least 5 degrees above the horizontal; this will help prevent moister from building up in the tube. The IMEC-168 ECM is shown in Figure 2.4.

![Image of IMEC-168 ECM](image)

**Figure 2.4: The IMEC-168 ECM mounted to the exterior of the generator.**

The features for the IMEC-168 ECM shown in the figure above are listed below:

- Microcontroller: Motorola 68HC08 family
- Microprocessor: 8-bit
- EEPROM Memory Space: 512 bytes
- Flash EEPROM Memory Space: 32 Kbytes
- Communication Link: Serial RS232, 9-pin connection to Laptop
- Software: Delphi Specific
  - ePS Flash Tool
  - Excel based Pixcal Calibration Tool
2.1.2 MT05 ECM Description

The MT05 controller, shown in Figure 2.5, is used to assist the control of engine management systems for one and two cylinder small engine applications by instantaneously measuring multiple events, enabling real-time adjustments of fuel, air, and spark [7]. The slim design of this controller, which no longer uses an internal MAP sensor, allowed for the mounting location to be on the inside of the generator frame away from accidental damage. An external MAP sensor was installed on the generator above the existing intake MAP port on the engine by means of a MAP tube. The new sensor and its location increase the reliability of the MAP signal over the previous system, since the signal is used to calculate many of the parameters that are used for the fuel control logic.

![Figure 2.5: The MT05 ECM mounted inside the generator frame.](image)

The fuel control logic for this controller calculates \( m_{\text{air}} \) and \( m_{\text{fuel}} \), as previously shown in this chapter, the controller then uses \( m_{\text{fuel}} \), the injector flow rate, 1.4 (g/s), and other transient fuel parameters to calculate the FPW delivered to the engine. CLC improves the FPW further by means of BLM, which uses a correction factor as long-term
adjustment to the FPW. This feature decreases the work load of the CLC correction term and improves the overall FPW delivered to the engine. When the engine begins to run rich, the FPW is increasing and the BLM decreases. If the engine begins to run lean, the FPW is decreasing and the BLM is increasing. The ideal operating point for BLM is 1, so below 1 indicates rich and above 1 indicates lean.

The features and advantages of the MT05 ECM are listed below [7]:

- Microprocessor: 16-bit
- EEPROM Memory Space: 512 bytes
- Flash EEPROM Memory Space: 256 Kbytes
- Communication Link: Controller Area Network (CAN)
  - Kvaser Leaf Light High Speed CAN to USB Device
  - Connects ECM to Laptop
- Software: Delphi Calibration Toolbox (CTB)
  - Real-time Data Logger
  - Data Playback Tool
  - Data File Export Tool
- External MAP Sensor
- Heated Oxygen Sensor
- Ability to tune look-up table axes
  - The distance between RPM/MAP break points can be increased
  - Improves engine performance
CHAPTER 3

Oxygen Depletion Testing

When the generator is running in an open environment, outside in moving air, oxygen is readily available and the intake air temperature of the generator will operate close to ambient conditions. Due to calibrations performed on the engine, it will run in closed-loop around 14.6 to 1 AFR, with very little deviations in engine variables. If the generator is running in an enclosed environment, specific engine variables will change rapidly to accommodate for the lack of oxygen. The engine will still operate in closed-loop, but as the exhaust builds in the space, the oxygen decreases, forcing the engine to run richer than 14.6 to 1 AFR. Closed-loop control in the ECM must adjust for the rich conditions by decreasing the amount of fuel being delivered to the engine. In order to identify the engine variables that react to an oxygen depleted environment, the generator had to be tested under open and closed-door conditions. This chapter will cover the facility used to initially test the generator, what tests were performed, and the results from those tests.
3.1 Oxygen Depletion Test House

A wooden shed 8 ft. x 4 ft. x 9 ft. 6 in., shown in Figures 3.1 and 3.2, was constructed for testing the generator. During testing, the generator’s engine variables and exhaust emissions were monitored from inside a lab away from any poisonous exhaust fumes.

Figure 3.1: O$_2$ Depletion Test House-Exterior.
3.2 $O_2$ House Testing and Results

Real-time data from the ECM, along with emissions data, were collected during each test. Since the ECM’s real-time data logger captures many different engine variables, which include measured parameters from sensors and calculated parameters, the focus for the initial testing was to analyze the parameters that were being measured by a sensor and fueling parameters. This data was then analyzed to determine how the engine performed, what engine parameters reacted to oxygen depletion, and how much the oxygen was being depleted from the test house. In order to vary the load applied to the generator, a Simplex 10 kW, single phase, 60 Hertz (Hz) load bank was used for each test. Table 3.1 shows the load settings applied to the generator.
Table 3.1
Load Bank Load Points

<table>
<thead>
<tr>
<th>Mode</th>
<th>Load (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5500</td>
</tr>
<tr>
<td>2</td>
<td>4750</td>
</tr>
<tr>
<td>3</td>
<td>3500</td>
</tr>
<tr>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

The initial test scenarios performed on the generator were open-door and closed-door settings under warm start conditions. A warm start condition, meaning the generator had to run for at least ten minutes before beginning the test scenario, was used to ensure the engine was warm and running in closed-loop. The purpose of having CLC enabled for all test scenarios was to ensure that BLM learning was taking place and that the engine was trying to run at 14.6 to 1 AFR. After engine startup, the engine oil temperature must reach 60 °C before BLM learning will occur.

During open-door testing, after warmup, real-time and emissions data was collected from the engine for modes 1, 4, and 6 listed in the Table 3.1. After the open-door tests were completed, the engine was shutdown for a substantial period of time to cool down. This process was used for each set of tests performed. The closed-door scenarios were performed in the same manner, but with one exception; after the engine was warm, the door was adjusted to the closed position, load was applied to the generator, and the data was collected. Once the test was complete, the door was adjusted to the open position to allow the test house to return to an ambient state before proceeding to the next load setting. The following section discusses the test results from each test scenario performed on the generator.
3.2.1 Open-Door Test Results

The expectation for open-door conditions was that the engine would run at conditions directly related to the load setting. The oil temperature (VCLTS) steadily continues to increase during engine operation under all load conditions, until a steady state temperature has been reached, but the MAP signal remains constant relative to the load. Although some minor changes were recorded, the IAT and O$_2$ content remain fairly constant throughout each load test. The open-door tests results show that from 0 W to 1500 W, the oxygen content of the test house did not drop below 20.0%, but when tested at 5500 W the O$_2$ content dropped just below 20.0%.

The following figures show the measured sensor signals and O$_2$ content vs. time, fueling parameters and O$_2$ content vs. time, and emissions output vs. time for three load conditions. Listed in the order tested, each graph displays the test condition and load point. The measured sensor signals and fueling parameters are oil temperature (VCLTS), intake air temperature (IAT), and manifold absolute pressure (MAP), fuel pulse width (FPW), and block learn memory correction (FBLMCOR). The emissions output displays percent oxygen (O$_2$), and carbon monoxide (CO), hydrocarbons (HC), and nitrogen oxide (NO$_x$) in parts per million (ppm).
Figure 3.3: Measured sensor signals and O$_2$ vs. time, open-door, 0 W.

Figure 3.4: Fueling parameters and O$_2$ vs. time, open-door, 0 W.
Figure 3.5: Emissions data, open-door, 0 W.

Figure 3.6: Measured sensor signals and O₂ vs. time, open-door, 1500 W.
Figure 3.7: Fueling parameters and $O_2$ vs. time, open-door, 1500 W.

Figure 3.8: Emissions data, open-door, 1500 W.
Figure 3.9: Measure sensor signals and \( \text{O}_2 \) vs. time, open-door, 5500 W.

Figure 3.10: Fueling parameters and \( \text{O}_2 \) vs. time, open-door, 5500 W.
3.2.2 Closed-Door Test Results

The expectations for closed-door conditions were that the engine would begin to run rich after some period of time, due to the exhaust depleting the O$_2$. The VCLTS would steadily continue to increase under all load conditions, until a steady state temperature was reached, just as it did in open-door conditions. As the speed continues to decrease, the governor forces the throttle open, which in turn causes the MAP signal to increase. As the engine runs under closed-door conditions, the exhaust output increases thereby decreasing the O$_2$ content and significantly increasing the IAT. The closed-door test results show that at 0 W the O$_2$ content of the test house dropped to 18.2% in 6 minutes. At 1500 W the O$_2$ content of the test house dropped to 18.1% in 4 minutes, and at 5500 W the O$_2$ content dropped just below 17.5% in 3 minutes. The following figures
show the measured sensor signals and O₂ content vs. time, fueling parameters and O₂ content vs. time, and emissions output vs. time for these three load points. The test condition and load point are specified for each graph and they are listed in the order they were tested.

Figure 3.12: Measured sensor signals and O₂ vs. time, closed-door, 0 W.
Figure 3.13: Fueling parameters and O$_2$ vs. time, closed-door, 0 W.

Figure 3.14: Emissions data, closed-door, 0 W.
Figure 3.15: Measured sensor signals and $O_2$ vs. time, closed-door, 1500 W.

Figure 3.16: Fueling parameters and $O_2$ vs. time, closed-door, 1500 W.
Figure 3.17: Emissions data, closed-door, 1500 W.

Figure 3.18: Measured sensor signals and O₂ vs. time, closed-door, 5500 W.
Figure 3.19: Fueling parameters and $O_2$ vs. time, closed-door, 5500 W.

Figure 3.20: Emissions data, closed-door, 5500 W.
CHAPTER 4

Oxygen Depletion Shutdown Algorithm

The O$_2$ depletion shutdown algorithm, described in this chapter, uses existing engine variables from the ECM to detect if the engine is running in an O$_2$ depleted environment. After analyzing the data in the previous chapter, it was decided that intake air temperature (IAT), fuel pulse width (FPW), and block learn memory correction (BLM) were the engine variables that reacted significantly to O$_2$ depleted conditions. When the generator is running in an enclosed environment, where the O$_2$ is being depleted, the IAT and FPW will gradually begin to increase, and the BLM will begin to decrease based on the FPW. The IAT increases due to the heat being emitted from the generator’s exhaust. The FPW, on the other hand, increases due to the AFR of the engine being rich, which means in an enclosed environment the engine is using more fuel because it has less oxygen. As the FPW increases, the CLC in the ECM begins compensating for the richness by correcting the FPW with the BLM. When the FPW becomes too rich, the BLM will decrease in order to compensate and drive the AFR back towards 14.6 to 1 by decreasing the FPW.
The algorithm is based upon a moving average of the collected data to calculate a pseudo-derivative for each signal. The pseudo-derivative is then used in the comparison logic to establish the trend of each signal. IAT and FPW will show increasing trends, and hence a positive pseudo-derivative, and the BLM will show a decreasing trend, and hence a negative pseudo-derivative, under closed-door conditions. Since a derivative is employed on physical signals with inherent noise, the input signal is filtered through the fixed-width window moving average. Although the algorithm does not actually calculate the average, the following set of equations illustrates how the concept of the moving average is used in the pseudo-derivative calculation. Equation 4.1 shows an average calculation of an array \( x \) over \( n \) points. As a new data point enters the array; the window slides, and a new average can be calculated according to Equation 4.2.

\[
\hat{x}_n = \frac{x_1 + x_2 + x_3 + \Lambda + x_n}{n} \quad (4.1)
\]

\[
\hat{x}_{n+1} = \frac{x_2 + x_3 + x_4 + \Lambda + x_{n+1}}{n} \quad (4.2)
\]

Equations 4.3 and 4.4 illustrate how the pseudo-derivative, \( \Delta x \), is calculated from Equations 4.1 and 4.2.

\[
\Delta x = \hat{x}_{n+1} - \hat{x}_n = \frac{x_2 + x_3 + x_4 + \Lambda + x_{n+1}}{n} - \frac{x_1 + x_2 + x_3 + \Lambda + x_n}{n} \quad (4.3)
\]

\[
\Delta x = \frac{x_{n+1}}{n} - \frac{x_1}{n} = \frac{x_{n+1} - x_1}{n} \quad (4.4)
\]

The calculation of Equation 4.5 shows the pseudo-derivative, \( \delta x \), as an approximation of the derivative over a number of samples. Note that the term pseudo-derivative is employed since the difference is numerically calculated and the time difference is not considered. Ultimately, stated more physically, if a sample taken now is higher than a
sample taken much earlier, then the signal is increasing. Furthermore, the magnitude of the derivative is not of interest, just the sign to indicate a trend.

\[ \Delta x = x_{n+1} - x_1 \] 

(4.5)

In order for this algorithm to work in the existing ECM software, new variable initializations had to be made for the IAT, FPW, and BLM signals, which can be found in Table 4.1. The variables listed in Table 4.2 are used to process the IAT, FPW, and BLM signals through the algorithm, and they must be restored to zero upon system restart.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>BufferSize</td>
<td>Integer</td>
<td>The number of points in the averaging window for IAT, FPW, and BLM arrays</td>
<td>512</td>
</tr>
<tr>
<td>Flagcnt</td>
<td>Integer</td>
<td>The size of the sample window over which the IAT, FPW, and BLM counters are compared to their respective counters</td>
<td>128</td>
</tr>
<tr>
<td>IAT_FailCounter</td>
<td>Integer</td>
<td>The number of hits that must be exceeded by the IAT comparison logic in the sample window to trip the IATflag</td>
<td>32</td>
</tr>
<tr>
<td>FPW_FailCounter</td>
<td>Integer</td>
<td>The number of hits that must be exceeded by the FPW comparison logic in the sample window to trip the FPWflag</td>
<td>64</td>
</tr>
<tr>
<td>BLM_FailCounter</td>
<td>Integer</td>
<td>The number of hits that must be exceeded by the BLM comparison logic in the sample window to trip the FPWCflag</td>
<td>64</td>
</tr>
<tr>
<td>IAT_FailThreshold</td>
<td>Floating</td>
<td>The change in IAT between samples that must be exceeded to result a hit in the IAT comparison logic</td>
<td>0.5</td>
</tr>
<tr>
<td>COen</td>
<td>Binary</td>
<td>The enable bit for the O₂ depletion algorithm</td>
<td>1</td>
</tr>
<tr>
<td>EngineTempThreshold</td>
<td>Integer</td>
<td>The minimum oil temperature in, °C, that allows the O₂ depletion algorithm to be enabled</td>
<td>50</td>
</tr>
</tbody>
</table>
Once initialization of the above variables has occurred, the algorithm buffers the incoming data from the IAT, FPW, and BLM signals into three arrays, respectively, until 512 data points have been collected. The algorithm continues to buffer the incoming data through the array by pushing the current data point in, popping out the first element, and computing the pseudo-derivative. The comparison logic for each signal uses the respective pseudo-derivative to perform a comparison, and if the comparison is true, the signal’s fail counter will be incremented by one count. The signal’s fail counters are then compared to their respective thresholds listed in Table 4.1, and if all three counters have reached their threshold point before FODSC resets to zero, the FODDET flag will be set to true resulting in engine shutdown. Once shutdown has occurred, the ECM must be turned off to clear the FODDET flag before restarting the engine. Upon restart, the variables used to process the IAT, FPW, and BLM signals will be reset to zero. The O$_2$ depletion algorithm pseudo-code is shown below:

### Table 4.2

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
<th>Restore Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FODBC</td>
<td>Integer</td>
<td>Used to buffer the IAT, FPW, and BLM signal arrays</td>
<td>0</td>
</tr>
<tr>
<td>FODSC</td>
<td>Integer</td>
<td>Used in the logic for which the IAT, FPW, and BLM counters are compared to their respective thresholds</td>
<td>0</td>
</tr>
<tr>
<td>FODIATC</td>
<td>Integer</td>
<td>Counter used in the IAT comparison logic and increments when the comparison is true</td>
<td>0</td>
</tr>
<tr>
<td>FODFPWC</td>
<td>Integer</td>
<td>Counter used in the FPW comparison logic and increments when the comparison is true</td>
<td>0</td>
</tr>
<tr>
<td>FODBLMC</td>
<td>Integer</td>
<td>Counter used in the BLM comparison logic and increments when the comparison is true</td>
<td>0</td>
</tr>
<tr>
<td>dVIAT</td>
<td>Floating</td>
<td>Pseudo-derivative for the IAT</td>
<td>0</td>
</tr>
<tr>
<td>dFPW2</td>
<td>Floating</td>
<td>Pseudo-derivative for the FPW</td>
<td>0</td>
</tr>
<tr>
<td>dFBLMCOR</td>
<td>Floatin</td>
<td>Pseudo-derivative for the BLM</td>
<td>0</td>
</tr>
<tr>
<td>FODDET</td>
<td>Binary</td>
<td>Flag used to shut down the engine</td>
<td>0</td>
</tr>
</tbody>
</table>
1. Set the buffer counter, FODBC=0, variables listed in Table 4.2 already initialized
2. Upon update of VIAT, FPW2, and BLMCOR
   a. FODBC++
   b. Begin initial buffer of VIAT, FPW2, and BLMCOR (BufferSize long, FIFO)
      i. IAT(FODBC)=VIAT
      ii. FPW(FODBC)=FPW2
      iii. BLM(FODBC)=FBLMCOR
   c. If FODBC<BufferSize
      i. GO TO 2
3. FODSC=0
4. Upon update of VIAT, FPW2, and BLMCOR
   a. FODSC++
   b. dVIAT=VIAT-IAT(1)
   c. dFPW2=FPW2-FPW2(1)
   d. dBLM=FBLMCOR-BLM(1)
   e. PUSH current VIAT value into the buffer
   f. PUSH current VIAT value into the buffer
   g. PUSH current VIAT value into the buffer
   h. If dVIAT>IAT_FailThreshold
      i. FODIATC++
      i. If dFPW2>0
         i. FODFPWC++
   j. If dFBLMCOR<0
      i. FODBLMC++
   k. If FODSC=Flagcnt
      i. If (FODIATC>IAT_FailCounter)&(FODFPWC>FPW_FailCounter) 
         & (FODBLMC>BLM_FailCounter) 
         1. FODDET=1, Disable Fuel Injection (SHUT DOWN ENGINE)
      ii. Else
         1. FODSC=0
         2. FODIATC=0
         3. FODFPWC=0
         4. FODBLMC=0
   l. GO TO 4

4.1 Post Processing of Initial Data

The following figures represent the results from post processing the open-door and closed-door data, from the previous chapter, through a simulation of the O₂ depletion shutdown algorithm. A copy of this code can be found in Appendix B of this thesis. The
open door test data did not produce any false trips and the closed door data tripped just as expected.

![Figure 4.1: Post processed-measured signals, open-door, 0W.](image)

![Figure 4.2: Post processed-fail hits, open-door, 0W.](image)
Figure 4.3: Post processed-trip flags, open-door, 0 W.

Figure 4.4: Post processed-shutdown signal, open-door, 0W.
Figure 4.5: Post processed-signals, closed-door, 0W.

Figure 4.6: Post processed-fail hits, closed-door, 0W.
Figure 4.7: Post processed-trip flags, closed-door, 0W.

Figure 4.8: Post processed-shutdown signal, closed-door, 0W.
Figure 4.9: Post processed-signals, open-door, 1500 W.

Figure 4.10: Post processed-fail hits, open-door, 1500 W.
Figure 4.11: Post processed-trip flags, open-door, 1500 W.

Figure 4.12: Post processed-shutdown signal, open-door, 1500 W.
Figure 4.13: Post processed-sIGNALS, closed-door, 1500 W.

Figure 4.14: Post processed-fail hits, closed-door, 1500 W.
Figure 4.15: Post processed-trip flags, closed-door, 1500 W.

Figure 4.16: Post processed-shutdown signal, closed-door, 1500 W.
Figure 4.17: Post processed-signals, open-door, 5500 W.

Figure 4.18: Post processed-fail hits, open-door, 5500 W.
Figure 4.19: Post processed-trip flags, open-door, 5500 W.

Figure 4.20: Post processed-shutdown signal, open-door, 5500 W.
Figure 4.21: Post processed-signals, closed-door, 5500 W.

Figure 4.22: Post processed-fail hits, closed-door, 5500 W.
Figure 4.23: Post processed-trip flags, closed-door, 5500 W.

Figure 4.24: Post processed-shutdown signal, closed-door, 5500 W.
4.2 Generator Tests with O$_2$ Depletion Shutdown Algorithm

Due to the propriety nature of the ECM’s software, the algorithm created had to be sent to the Delphi software engineers in China for implementation. The code and test data was reviewed by Delphi and then implemented in the existing software for the MT05 controller. Upon receiving the new software, it was programmed into the memory of ECM and then tested to ensure the algorithm was properly coded in the software. The figures below represent the success of the O$_2$ depletion shutdown algorithm when tested at the UA test facility under high and low loading conditions.

Figure 4.25: O$_2$ depletion parameters, closed-door, O W.
Figure 4.26: Algorithm fail counters and trip flag, closed-door, O W.

Figure 4.27: O₂ depletion parameters, closed-door, 5500 W.
Figure 4.28: Algorithm fail counters and trip flag, closed-door, 5500 W.
CHAPTER 5

NIST Testing

Once the UA testing of the O₂ depletion shutdown algorithm was complete, the next phase of testing could begin. The generator was sent to the National Institute of Standards and Technology (NIST) in Gaithersburg, Maryland for testing in more realistic test scenarios. NIST is equipped with an Indoor Air Quality testing facility, which can be seen in Figure 5.1. The garage, located on left end of the test house from this view, is where the generator was setup for testing.

![Figure 5.1: NIST Indoor Air Quality Test House.](image)

The following list shows the different scenarios tested in the NIST test house:

- Warm start, open bay door, 5500 W
- Warm start, closed bay door, 5500 W
- Warm start, closed bay door, 2500 W
- Warm start, bay door open 24 in., 2500 W
- Warm start, bay door open 24 in., 5500 W
The tests listed above, worked as expected, the open bay door tests did not produce any false trips, and closed bay door tests caused the engine to trip as expected. The algorithm still tripped the engine during the tests where the bay door was open 24 in., since the amount of clean air moving into the garage was not significant enough to filter out the exhaust. The 5500 W test tripped faster than the 2500 W test, due to the high load, which produces more heat. When the engine is operating at mid-to-low loads, the algorithm does not respond as quickly as it does at high loads. It was discover at the NIST facility that under no load and lightly loaded conditions, the engine will run for 3 hours before the algorithm shuts down the engine. The following figures illustrate the tests results from the NIST facility.

Figure 5.2: O₂ depletion parameters, open-door, warm start, 5500 W.
Figure 5.3: Algorithm fail counters and trip flag, open-door, warm start, 5500 W.

Figure 5.4: $\text{O}_2$ depletion parameters, closed-door, warm start, 5500 W.
Figure 5.5: Algorithm fail counters and trip flag, closed-door, warm start, 5500 W.

Figure 5.6: O₂ depletion parameters, door open 24 in., warm start, 5500 W.
Figure 5.7: Algorithm fail counters and trip flag, door open 24 in., warm start, 5500 W.

Figure 5.8: $O_2$ depletion parameters, closed-door, warm start, 2500 W.
Figure 5.9: Algorithm fail counters and trip flag, closed-door, warm start, 2500 W.

Figure 5.10: O$_2$ depletion parameters, door open 24 in., warm start, 2500 W.
Figure 5.11: Algorithm fail counters and trip flag, door open 24 in., warm start, 2500 W.
CHAPTER 6

Conclusions

This thesis has described the Engine Management System and the components that it uses to control the AFR around 14.6 to 1 for all operating conditions of the Coleman Powermate 7000 Watt generator. It was the goal of the efforts from testing this generator, to identify the engine variables that detect $O_2$ depleted conditions. An automatic shutdown algorithm was developed and tested based upon the IAT, FPW, and BLM engine variables identified during initial testing. Upon completion of the algorithm, the initial test data was then post processed through the algorithm to verify the algorithm would shutdown the engine in an enclosed environment.

After verification, the algorithm was sent to Delphi for implementation into the software of the MT05 ECM. Once the software was received from Delphi, it was loaded into the ECM’s memory and tested to ensure the algorithm was properly coded into the software. The final testing took place at the NIST facility in Gaithersburg, Maryland, where more realistic test scenarios were available. The testing results from the NIST facility, that were discussed in this thesis, show that when run in an enclosed environment the generator will shutdown as programmed, and in open-door settings the generator will run at 14.6 to 1 AFR, producing little amounts of CO in the garage. It was also discovered that during lightly loaded conditions, the engine would run for several hours before the algorithm would trip the engine.
6.1 Future Work

Further modifications can be made to the shutdown algorithm, to improve its response time when operating in an enclosed environment. Each load condition behaves differently, so modifications can be made to account for the change in MAP. This would require the use of a lookup table based upon RPM and MAP, where the value in the table would be change a magnitude of the change in MAP. As testing continues, more doors could be opened to future ways of improving upon the existing algorithm documented in this thesis.
REFERENCES


APPENDIX

O₂ Depletion Shutdown Algorithm Simulation Code

%Oxygen Depletion Algorithm
%CPSC - University of Alabama
%Simulation of ECU
%Workspace Variables
%   DatTab=[ts IAT FPW FPWC]
%       time, intake air temp, fuel pulse width, fuel pulse width
correction
[nsamp,m]=size(DatTab);
%Flash variables
Npts=512;
Flagcnt=128;
Fthresh=64;
FIATthresh=32;
Tlim=0.001;
Tlim=Tlim*Npts;
COen=1;
%Initialization
dVIAT=0;
dFPW2=0;
dFBLMCOR=0;
FPWpos=0;
FPWCneg=0;
IATpos=0;
FPWflag=0;
FPWCflag=0;
IATflag=0;
COflag=0;
%Begin simulation
PCNT=0;
while PCNT<(Npts); %Loop models the initial buffering (step 2)
    PCNT=PCNT+1;
    VIAThold(PCNT)=DatTab(PCNT,2);
    FPW2hold(PCNT)=DatTab(PCNT,3);
    FBLMCORhold(PCNT)=DatTab(PCNT,4);
end
WINCNT=0;
for i=Npts+1:nsamp; %Loop models the continuing incoming data (step 4)
    WINCNT=WINCNT+1;
    dVIAT=DatTab(i,2)-VIAThold(1);
    dFPW2=DatTab(i,3)-FPW2hold(1);
    dFBLMCOR=DatTab(i,4)-FBLMCORhold(1);
    for k=1:Npts-1; %Loop models buffering actions
        VIAThold(k)=VIAThold(k+1);
        FPW2hold(k)=FPW2hold(k+1);
FBLMCORh0ld(k)=FBLMCORh0ld(k+1);
end
VIATh0ld(Npts)=DatTab(i,2);
FPW2h0ld(Npts)=DatTab(i,3);
FBLMCORh0ld(Npts)=DatTab(i,4);
if dVIAT>Tlim;
    IATpos=IATpos+1;
end
if dFPW2>0;
    FPWpos=FPWpos+1;
end
if dFBLMCO<0;
    FPWCneg=FPWCneg+1;
end
if WINCNT==Flagcnt;
    if FPWpos>Fthresh;
        FPWflag=1;
    else
        FPWflag=0;
    end
    if FPWCneg>Fthresh;
        FPWCflag=1;
    else
        FPWCflag=0;
    end
    if IATpos>FIATthres;
        IATflag=1;
    else
        IATflag=0;
    end
    COflag(i)=FPWflag&FPWCflag&IATflag;
WINCNT=0;
FPWpos=0;
IATpos=0;
FPWCneg=0;
if COflag&COen;
    DatTab(i,1)
    break
end
IAThits(i)=IATpos;
FPWhits(i)=FPWpos;
FPWCchits(i)=FPWCneg;
IATtripcnt(i)=IATflag;
FPWtripcnt(i)=FPWflag;
FPWCtripcnt(i)=FPWCflag;
end
%Plot simulation output data
%Plot input variables signals
figure
subplot(3,1,1)
plot(DatTab(:,1),DatTab(:,2))
xlabel('Time (s)')
ylabel('Degrees C')
title('Intake Air Temp')
subplot(3,1,2)
plot(DatTab(:,1),DatTab(:,3))
xlabel('Time (s)')
ylabel('ms')
title('Fuel Pulse Width')
subplot(3,1,3)
plot(DatTab(:,1),DatTab(:,4))
xlabel('Time (s)')
title('Fuel Pulse Width Correction')

% Plot hit counts
figure
subplot(3,1,1)
plot(IAThits)
xlabel('Time (s)')
ylabel('counts')
title('Intake Air Temperature Hits')
subplot(3,1,2)
plot(FPWhits)
xlabel('Time (s)')
ylabel('counts')
title('Fuel Pulse Width Hits')
subplot(3,1,3)
plot(FPWChits)
xlabel('Time (s)')
ylabel('counts')
title('Fuel Pulse Width Correction Hits')

% Plot trip flags
figure
subplot(3,1,1)
plot(IATtripcnt)
xlabel('Time (s)')
title('IAT Trip Flag')
subplot(3,1,2)
plot(FPWhittripcnt)
xlabel('Time (s)')
title('FPW Trip Flag')
subplot(3,1,3)
plot(FPWChittripcnt)
xlabel('Time (s)')
title('FPWC Trip Flag')

% Plot shutdown signal
figure
plot(COflag)
xlabel('Time (s)')
title('Shutoff Signal')