

Andrew Palardy

CE-499

December 7, 2017

Design of Common Rail Diesel Control System

This paper describes the design and implementation of a Common Rail Diesel control system for a Mercedes/Smart OM660 engine. The OM660 engine is a three cylinder, 800cc, inline engine featuring a Bosch common rail injection system.

The control system is based on the Woodward MotoHawk development system. MotoHawk provides a Simulink development environment and control hardware, designed for rapid prototype and low to medium volume production solutions. The Simulink environment prioritizes a model-based design methodology. MotoHawk is used for engine and aftertreatment controls in many off highway and marine applications.

The Woodward control system has been used successfully on previous Kettering Motorsports vehicles, starting with the Formula SAE car in 2013. The methods for developing Motohawk models have been refined for both Formula SAE and Clean Snowmobile since then. Based on the experience gained in the previous models, the Common Rail Diesel model was developed without carrying over previous code structure, as the previous structure was shown to be inadequate over time.

As the Woodward modules do not contain the high voltage electronics required for driving DI injectors, an additional box was built to contain the power electronics required. The DI driver box was designed using an NXP MC33816 development kit, with modifications to drive the injectors from spark outputs of the Woodward module.

1. Software Overview

The Common Rail Diesel control software is organized into subsystems, each with an abbreviated code. The subsystems are grouped in Simulink into the following categories:

- Sensors
- Gas Flows (GF)
- Combustion Manager (CM)
- Engine Torque Management (ETRQ)
- Fuel Injection Equipment (FIE)
- Thermal Management

Within Motohawk, the root of the model does not contain executable code. Motohawk requires all executable model code to be triggered using function-call subsystems. The root of

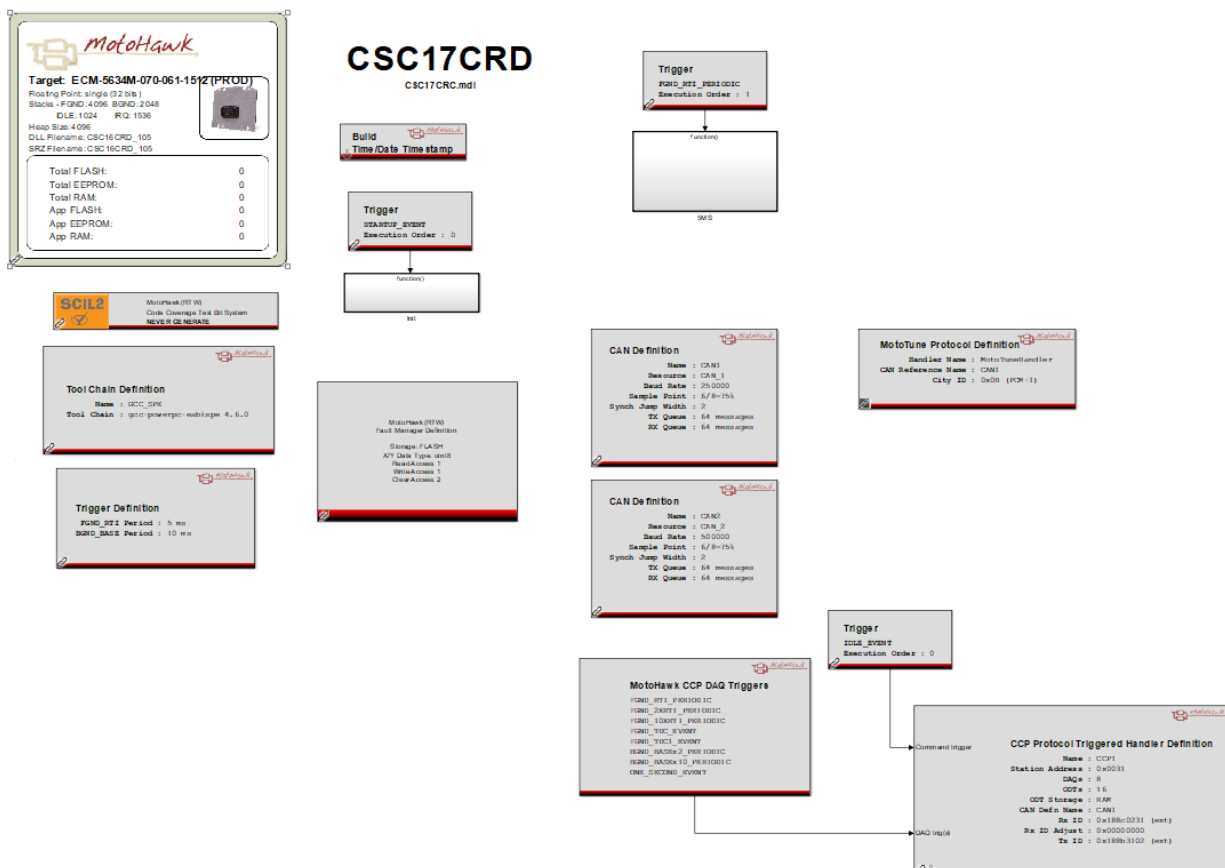


Figure 1: CSC 17 CRD Model Root

the Diesel model is shown below, along with the 5ms primary control task. Additionally, a graphic describing the torque control path is shown.

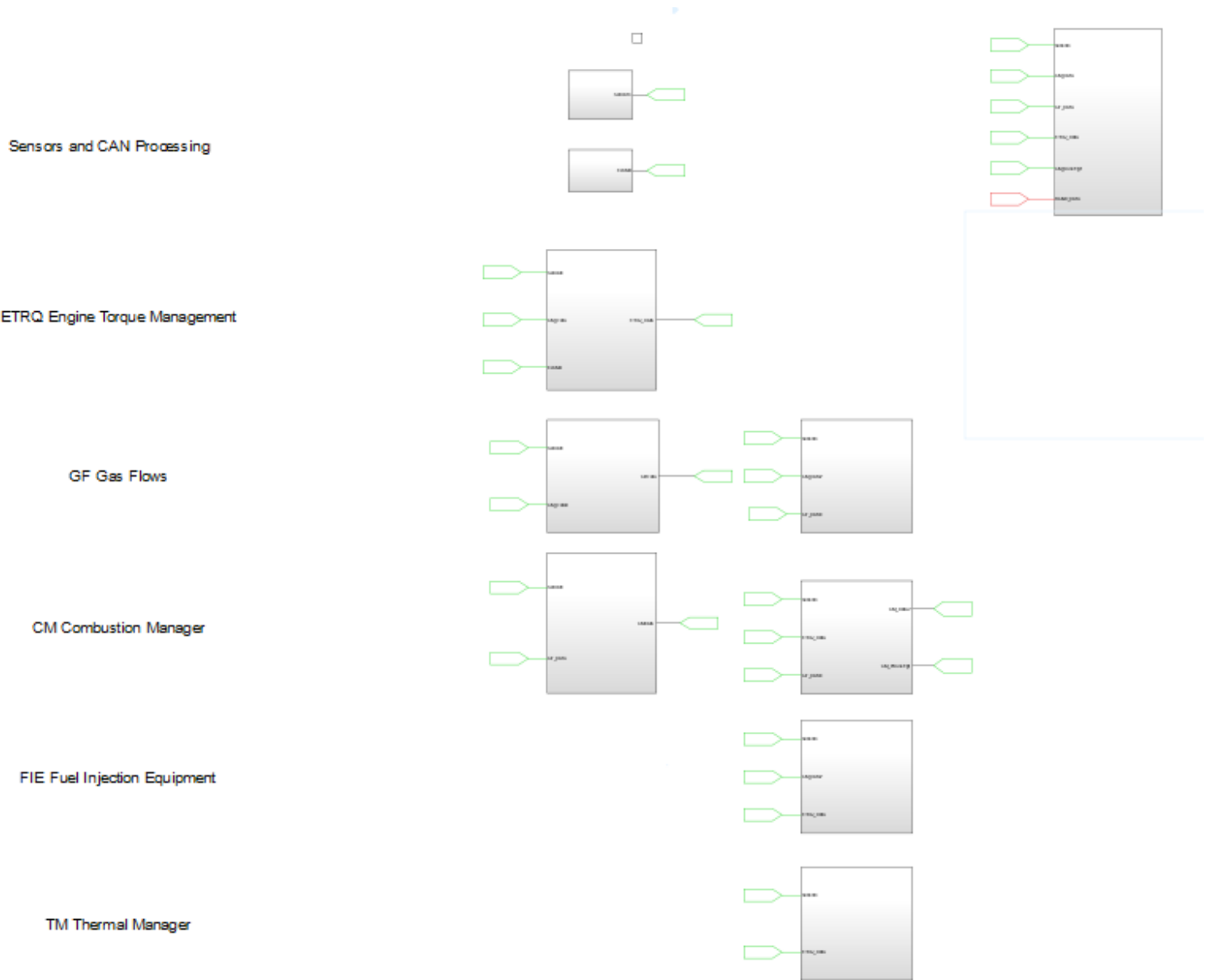


Figure 2: CSC 17 CRD 5ms Task

TORQUE CONTROL PATH

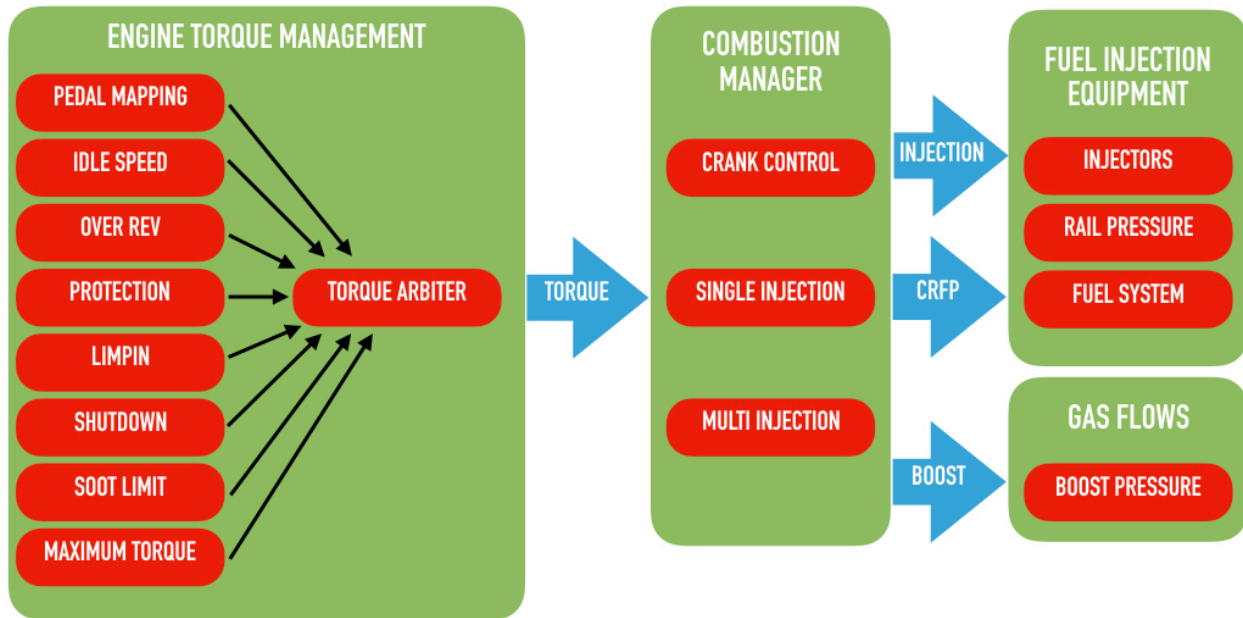


Figure 3: Torque Control Path

2. Sensor Processing

2.1. Sensor Overview

All sensors are processed in the Sensors subsystem. Each sensor contains a unique subsystem dedicated to input processing for that sensor. Each sensor is assigned an abbreviation, which is used as the prefix for all associated labels. As many sensors are implemented using a masked subsystem and are identical in design, only a few will be described in detail. The following sensors are implemented, although not all are used by algorithm code:

- Battery Voltage (VBatt)
- Key Sense Voltage (VECU)
- Sensor Supply Voltage (VRef)
- Engine Coolant Temperature (ECT)
- Engine Oil Temperature (EOT)
- Engine Oil Pressure (EOP)
- Common Rail Fuel Pressure (CRFP)
- Air Charge Temperature (ACT)
- Intake Air Temperature (IAT)
- Intake Mass Air Flow (MAF)
- Diesel Particulate Filter Differential Pressure (DPFP)
- Crankshaft Position and Camshaft Position Encoder
- Vehicle Speed Signal (VSS)
- Pedal Voltage Sense (PVS), redundant inputs

2.2. ECU Internal Voltage Sensing

The ECU is powered by two separate external power supplies, and one internally generated supply. A similar subsystem is used to process all three power supply voltages, and provide fault detection appropriate for each. The three supplies are referred to by Motohawk as 'VECU' (ECU Key Voltage Sense), 'DRVP' (ECU Driver Power input), and 'XDRP' (Transducer Power). The 'VECU' signal is labeled 'VECU' in the model, the DRVP signal is labeled 'VBatt', and the 'XDRP' signal is labeled 'VRef'. The DRVP input is fed from the engine auto shut-down (ASD) relay, and cannot sense when the relay is switched off. DRVP was selected to supply VBatt because VECU contains a diode which affects the voltage measurement accuracy, however the model will use VECU for VBatt when the ASD relay is switched off. Each sensor supply subsystem is similar and contains the following components:

- (1) Motohawk analog input block
- (2) Sensor Scaling via Gain Only
- (3) Sensor Low-Pass Filtering
- (4) Sensor Scaling Adjustment (VRef only)
- (5) ASD Active Sense (VBatt only)
- (6) Sensor Fault Detection (only applicable cases)
- (7) Sensor Shared Index

2.3. Engine Coolant Temperature (ECT)

The Engine Coolant Temperature (ECT) sensor is representative of most simple analog input sensors used in the control system. The ECT sensor subsystem contains the following key components:

- (1) Motohawk analog input block
- (2) Sensor Scaling via Look-Up Table
- (3) Sensor Low-Pass Filtering
- (4) Sensor Fault Detection for High and Low range faults (i.e. open and short circuits)
- (5) Sensor Limp-In replacement

- (6) Sensor Shared Index

2.4. Vehicle Speed Signal (VSS)

The Vehicle Speed Signal (VSS) is representative of most simple frequency input sensors used in the control system. The VSS sensor subsystem contains the following key components:

- (1) Motohawk frequency input block
- (2) Sensor Scaling via Gain Only
- (3) Sensor Low-Pass Filtering
- (4) Sensor Shared Index
- (5) Note, Fault detection is not possible as the VSS signal may legitimately report 0 Hz.

2.5. Synthesized Barometric Pressure (BARO)

The barometric pressure is currently set to a fixed calibration value equivalent to an altitude of approximately 1000 feet above sea level. The Synthesized Baro subsystem allows an algorithm to be implemented in the future to capture and estimate the barometric pressure using the MAP (Manifold Absolute Pressure) sensor.

2.6. Pedal Voltage Sensor (PVS)

The Pedal Voltage Sensor (PVS) is more complex than the ECT and other simple analog sensors because it is fully redundant. A single subsystem is used to process both sensors (PVS1 and PVS2), which have unique output curves designed to improve the ability to detect a wide range of circuit and sensor faults. The PVS system can detect and accommodate individual sensor faults by selecting the healthy sensor, and detect cross-sensor or common mode faults with a default value replacement. The PVS system also allows a single PVS sensor to be used with alternate scaling and no fault detection during dyno testing, where a potentiometer with a knob is used. The PVS sensor subsystem contains the following key components:

- (1) Subsystems for PVS 1 and PVS 2 input processing, containing:
 - (1) Motohawk analog input block
 - (2) Sensor Scaling via Look-Up Table
 - (3) Alternate Scaling for Dyno Test Usage (PVS 1 only)
 - (4) Sensor Fault Detection for High and Low range faults (i.e. open and short circuits)
- (2) Active sensor selection
- (3) Sensor Low-Pass filtering
- (4) Sensor Fault Detection for Cross-Sensor Rationality faults
- (5) Sensor Limp-In replacement
- (6) Sensor Shared Index

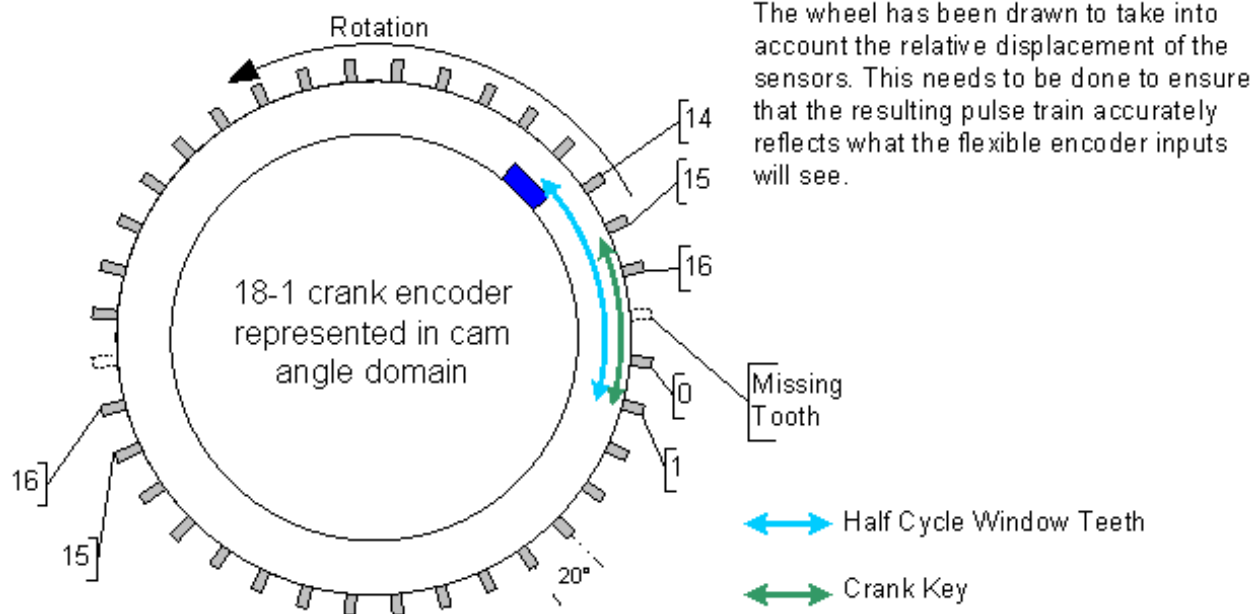
2.7. Engine Synchronization Encoder

While most sensor subsystems focus on processing the data received by a single sensing point, the Engine Synchronization Encoder subsystem processes the data from two unique sensors (Crank and Cam position) to allow the engine control to establish engine synchronization for angular events.

Mechanically, the engine cam shaft spins at 1/2 crank speed, and one complete cam rotation is one full engine cycle. The cam shaft is equipped with a single tooth Hall-effect sensor and the crank shaft is equipped with a 60 tooth (with two missing teeth) variable reluctance (VR) sensor. The two sensors are used to establish a 1/16th crank degree resolution timer which is synchronized to the full engine cycle and used to drive angular events.

The operation of the encoder is done by Motohawk using the eTPU hardware and the Flexible Encoder simulink block set. Both sensors are edge-synchronized using the falling edge only. An example timing chart is provided below, which shows a missing tooth crank with 18 teeth (one missing) showing a full engine cycle in the cam angle domain. The OM660 engine is identical but with a 60 tooth crank sensor.

Motohawk provides a number of crank related status flags and faults, and these are used to detect loss of crank, loss of cam, reversed rotation, and incorrect number of detected teeth. Usually, all of these faults are indicative of a wiring error on the cam or crank sensor, although they could also be caused by reversed polarity of the VR crank sensor.



2.8. Auto Shut-Down Control (ASD)

The Auto Shut-Down (ASD) system is responsible for managing the ECU's power on hold and reset status. When voltage is applied to the key switch input, the power supply automatically turns on and the processor boots up. The processor will turn off when the key switch voltage is removed unless the processor holds the power supply on.

The ASD system measures the voltage on the Key switch input (VECU) to sense the current keys status, and holds the power supply on once the ECU has finished booting. The ASD system will continue to hold the power supply on after the key voltage is removed, to allow the ECU to complete a logical power-down and save non-volatile memory. In addition, the ASD system controls the ASD relay, which provides power to the engine loads.

The ASD system also provides calibration hooks to allow the ASD relay to be forced on, and the ECU to remain powered up even when the key switch input is removed. These features are used during engine dyno testing, allowing the key switch input to shut down the engine without losing communication to the ECU. In addition, a calibration hook is provided to allow the user to manually force the non-volatile data to be written to EEPROM.

3. Gas Flows (GF)

3.1. Charge Calculation

The Charge Calculation is performed on several inputs, as shown later. The Charge calculation calculates the mass entering a single cylinder, over a single engine cycle. The calculation is based on the ideal gas law and the molecular weight of dry air.

$$\frac{P_{man} V_{disp} \eta_{vol} M_{air}}{T_{man} R_{air}} = m_{charge}$$

The equation is solved for the mass m_{charge} at the current intake manifold pressure P_{man} and temperature T_{man} , along with the displacement per cylinder V_{disp} . The resulting charge is then corrected based on the volumetric efficiency η_{vol} of the engine, which is mapped in a lookup table indexed by the engine speed and the intake manifold pressure ratio $\frac{MAP}{BARO}$.

The charge calculation block also calculates the mass flow rate into the engine, using the calculated charge, engine speed, and number of cylinders.

3.2. Gas Flow Model

The Charge calculation is performed at two conditions, the current intake manifold conditions and the predicted max load conditions (using the current air temperature and maximum boost pressure at the current engine speed). The maximum boost pressure ratio achievable is mapped with respect to the current engine speed.

3.3. Boost Pressure Control

The Gas Flows system is responsible for controlling boost pressure based on the demand pressure calculated by the combustion manager. Boost pressure is controlled

mechanically using a wastegate, which allows exhaust gas to bypass the turbine to reduce turbocharger input energy. The wastegate is actuated using a pneumatic diaphragm and spring, which is connected to the intake manifold through a solenoid controlled valve. If the solenoid valve is not actuated, the diaphragm will automatically regulate the boost to a fixed pressure determined by the spring in the wastegate actuator. If the solenoid valve is actuated, the actuator input pressure is reduced to ambient air pressure and the spring causes the wastegate to close. The solenoid valve is duty cycle controlled at low frequency to modulate the pressure 'seen' by the wastegate actuator, resulting in the appropriate boost pressure regulation, as long as the target boost pressure is greater than the wastegate spring pressure and can be achieved with the wastegate fully closed.

The target boost pressure is calculated by the Combustion Manager as an absolute pressure. To calculate the open loop wastegate solenoid duty cycle, Gas Flows converts the pressure into a target gauge pressure by subtracting the current barometric pressure, and then calculate the fraction between the wastegate spring gauge pressure and the target boost gauge pressure. The resulting duty cycle may be inverted through calibration, depending on the polarity of the wastegate solenoid valve.

$$DC_{wg} = \frac{P_{spring}}{P_{obj} - P_{baro}}$$

4. Engine Torque Management (ETRQ)

4.1. Pedal Torque Request

The pedal torque request is determined by the current pedal position, the eco mode, and the N/V ratio. The N/V ratio is calculated as the ratio between the engine speed (N) in RPM and the vehicle speed (V) in kph, limited to 1024 and protected against division by zero. The pedal torque percent is mapped using a lookup table indexed by the pedal position and N/V ratio. Multiple pedal maps are provided to allow for map testing, and for dyno use.

After the pedal torque percent is determined, the pedal torque request is calculated linearly between the minimum and maximum torque requests determined by torque arbitration. When in dyno raw mode, the results of the dyno pedal table are used directly, without scaling between minimum and maximum torque. The final torque result is presented to the torque arbiter.

4.2. Idle Speed Control

The target idle speed is set using the base idle speed calibration, with modifiers for coolant temperature and air temperature. The usage of modifiers is left up to the calibrator, it may not be necessary to use all of them.

Once the target idle speed is determined, the idle speed torque request is determined by adding friction torque to the idle speed torque table, indexed by idle speed error. There are plans to replace this table with a feedback controller in the future. To allow a feedback controller to be used in the future, the idle controller submits two torque requests to the arbiter. The first is the nominal torque request, which is used to schedule the pedal. The second is the instantaneous torque requests, which is actually requested on the engine.

A future improvement project is planned to add closed-loop torque control to the idle speed controller, and allow the idle speed controller to be used for constant-speed dyno testing.

4.3. Over Rev Control

The over rev speed is set to a scalar speed by calibration. The over rev torque request is set to positive infinity, reduced to zero at the over rev speed using a calibratable speed error gain. This prevents the engine from producing torque at and above the over rev speed.

4.4. Maximum Torque

The maximum torque request is a table indexed by RPM. This table may be used to shape the engine torque curve as needed.

4.5. Engine Protection

Engine Protection encompasses engine torque limits as a result of engine coolant temperature, engine coolant temperature, engine oil pressure, and the generic Limp-In path. The minimum of individual engine protection limits is sent to the torque arbiter.

Temperature based torque limits are all implemented identically. All temperature limits include a torque limit curve (vs RPM), protection temperature, and protection gain. The protection linear torque is calculated as the difference between the measured temperature and the protection value, multiplied by the protection gain, and limited to a number between 0 and 128. When the thermal limit is approached, the protection linear torque approaches 0. When the thermal limit is exceeded, the protection linear torque is 0. The protection torque limit rule is added to the linear torque to determine the torque request for a specific temperature limit. In this way, the engine torque curve may be shaped when the limit is fully applied, and the limit is blended in as the limit is approached. Limits are provided for both high and low coolant and oil temperature, each with unique torque limit curves and gains.

Oil pressure based torque limit is determined similarly to the temperature based torque limits, except the protection pressure is mapped in a table indexed by engine speed and oil temperature. This allows the protection to be disabled or modified based on these two

conditions, as determined by the calibrator. In most cases, the oil pressure expected is reduced at low temperature, but not removed entirely.

The final engine protection torque is the Limp-In path. Motohawk Fault Actions are used to map faults to the Limp-In path. Two actions are defined, 'Level 1' and 'Level 2'. Level 2 Limp-In is assumed to be more severe. Each level contains a protection torque table which may be defined by the calibrator. These tables are intended to be used for severe circuit or control faults which do not require engine shutdown.

4.6. Soot Limiting Torque

The limiting of soot during steady-state and transient conditions is very important for Diesel engines. Soot limiting is calculated by Combustion Manager for steady state and instantaneous conditions based on the current and maximum airflow and a maximum phi (fuel to air ratio) limit. A future improvement project is planned to calculate the desired boost based on the target phi value, further improving soot control.

4.7. Torque Arbitration

The final step in the torque management path is to determine the actual torque request on the engine. The torque arbiter is run in two steps, the initial arbiter determines the requests not including the pedal, and the final arbiter determines the final torque request.

The Initial torque arbiter determines the minimum and maximum torques for pedal mapping. The minimum is equal to the greater of idle nominal torque and zero. The maximum is equal to the lesser of the steady state soot limit torque and the maximum torque.

The final torque arbiter determines the torque request on the engine, and also the ID of the torque requestor currently controlling the torque path. The torque request is initially set to zero. Each subsequent requestor is treated as a minimum or maximum. First, the maximum of all minimum requestors is determined. Minimum requestors are idle control and pedal torque. Second, the minimum torque of the previous step as well as all maximum requestors is

determined. Maximum requestors are soot limit, maximum torque, over rev torque, protection torque, and shutdown torque.

5. Combustion Manager (CM)

5.1. Maximum Boost Pressure

The maximum boost pressure is mapped as a table from engine speed. This pressure is used to calculate the maximum charge.

5.2. Friction Torque

The engine friction torque is mapped using a lookup table indexed by engine speed and coolant temperature. The engine friction torque model subsystem allows additional sources of friction to be estimated in the future, such as accessory load torques.

5.3. Soot Limit Torque

The soot limit is calculated from the maximum fuel to air equivalence ratio (ϕ). The equivalence ratio is mapped in a table indexed by engine speed and coolant temperature, with unique tables for steady state and instantaneous soot limits. The instantaneous soot limit must be equal to or higher than the steady state limit. The instantaneous soot limiting torque is calculated from the current charge, stoichiometric fuel to air ratio, and instantaneous equivalence ratio. The steady state soot limiting torque is calculated similarly, but from the maximum charge and steady state equivalence ratio.

5.4. Combustion Definitions

For the purposes of combustion control, many quantities are calculated. The primary quantity is known as 'QObj', also known as the 'Torque Quantity', and is equal to the torque requested by ETRQ. This quantity is defined as the quantity required to produce the requested torque using a single optimized injection. In addition to the torque quantity, the total quantity is also calculated. This is the sum of all fuel delivered in a single engine cycle, including injections

and injection fractions which do not produce torque. Total quantity is used to calculate fuel flow rate and fuel economy.

In addition to quantities, several injections are named for the purposes of combustion control. Four injections are currently defined, named 'Pilot', 'Main', and 'Post'. The Pilot injection is advanced of the main and normally used to initiate combustion and reduce the pressure spike during the rapid burn down phase of combustion. The Main injection is used to deliver the bulk of the torque quantity. The Post injection is related to the torque quantity and may be used by the calibrator to either implement a split main injection, or an additional delayed injection to burn off excess soot.

5.5. Stall Injection Mode

No injections are scheduled during stall mode. However, the injection requests are set to zero during this case.

5.6. Crank Injection Mode

Torque quantity is not used during cranking mode. The quantity for the pilot and main injection, as well as the angles, are mapped.

5.7. Single Injection Mode

Single Injection Mode is only used for calibration. It provides calibrations for rail pressure and injection angle for a single torque injection, with all other injections disabled.

5.8. Multi Injection Mode

Multi Injection Mode is the primary set of tables used to calibrate engine operation. Multi Injection Mode provides control of all three injections to allow optimal efficiency and emissions, at the discretion of the calibrator. Multi Injection Mode provides maps based on engine speed and torque quantity to control all combustion parameters. All three injections contain maps for start of injection angle. The pilot and post injection angles are calculated as

offsets from the main injection angle. The pilot and post injections contain maps for delivery factors, which are calibrated to represent the torque produced as a result of those injections (i.e. 0 represents no torque, 1 represents full torque for their quantity). The quantity for the main injection is calculated from the torque quantity, minus the pilot quantity times its delivery factor, minus the after quantity times its delivery factor.

5.9. Planned Enhancements

One planned improvement is to restructure the phi and airflow calculations in Combustion Manager and Gas Flows. This would move the maximum boost calibration to Gas Flows. Additionally, it would calculate a target phi in combustion manager, with Gas Flows calculating the required boost to meet the target phi.

A second planned improvement is to support a fourth injection which is not torque related, which increases the exhaust gas temperature for the purpose of regenerating the diesel particulate filter.

6. Fuel Injection Equipment (FIE)

6.1. Fuel Injector Pulse Width

The injector pulse width is mapped based on injection quantity and common rail pressure. The injector flow curves must be determined using an injector flow bench. In addition, all injections less than a fixed threshold are dropped entirely.

6.2. Schedule Injections

Resulting injector pulses are scheduled using Motohawk Multiple PSP (Periodic Synchronous Pulse) behavior blocks, one for each cylinder. The blocks are configured with the appropriate TDC angle for the respective cylinder. The same injections are scheduled for all cylinders.

6.3. Rail Pressure Control Methods

Common rail pressure is generated using an engine mounted high pressure piston pump. The pressure in the rail may be controlled in three different ways, two of which are supported by the current control software. In the first method, one lobe of the pump is disabled by a solenoid valve, reducing the pumping capacity by 1/3. This is known as 'Binary VCV' (Volume Control Valve) control and is used by the OM660. This method is not used on modern pumps, which often have only a single lobe for small engines. In the second method, the flow into the high pressure pump is restricted to reduce the volume pumped. This method is known as inlet metering and is achieved using a Linear VCV valve, also known as a Metering Unit (MeUn). This method is not present on the OM660 and not implemented in this control. Both of these methods reduce the work done by the high pressure pump, and thus the mechanical power draw on the engine. The third method relies on a rail pressure control valve (PCV) located on the fuel rail, which bleeds fuel to the return line to maintain the specified pressure. The PCV valve is the primary method of rail pressure control for the OM660.

6.4. Pressure Control Valve (PCV)

The Rail Pressure Control Valve is physically a variable-force solenoid, and thus the current through the solenoid should be proportional to the target rail pressure. However, the Woodward module does not support constant-current outputs, so the solenoid is PWM controlled to a target voltage. The target voltage is the sum of the open-loop voltage and the closed loop voltage. The resulting voltage is converted into a PWM duty cycle using the battery voltage.

The open loop voltage is mapped by a curve from the target rail pressure. The closed loop voltage implements a PI type controller, with special handling for the integrator to avoid wind-up. Closed loop control is disabled when the engine is not running or when there are circuit faults for the rail pressure sensor. Previously, closed loop control was additionally disabled when the engine is in crank mode, but this requirement was removed for 2018 software due to irregularities with open loop rail pressure at low fuel temperatures.

In addition to controlling the PCV valve, the PCV subsystem is responsible for solenoid diagnostics. Motohawk provides short and open circuit diagnostics, and these are passed to the Motohawk fault manager.

6.5. Volume Control Valve (VCV)

As the OM660 engine contains a CP1 type common rail pump (instead of the more recent CP4 pumps), the pump does not contain an inlet metering valve. Instead, it contains a binary solenoid which disables one lobe of the three-lobe pump, reducing the effective displacement. The solenoid may be activated when the engine is in run mode, the solenoid is enabled, and the coolant temperature has reached the activation threshold. The solenoid is controlled by two curves which define the switch-on and switch-off torque quantity versus engine speed, allowing the calibrator to map a hysteresis as appropriate. Like the PCV, the VCV passes hardware diagnostic information to the Motohawk fault manager.

6.6. Lift Pump

As the CP1 pump on the OM660 engine does not contain a dedicated suction pump and cannot suck fuel through fuel system restrictions, fuel must be supplied under pressure to the pump inlet. The pressure does not need to be substantial, but must be higher than the return line backpressure to prevent the pump's internal shutoff valve from closing. The lift pressure is supplied by the fuel lift pump, and electric pump located in the fuel tank. The lift pump is turned on whenever the lift pump is enabled in calibration, the kill switch is enabled, the key switch is enabled, and either the engine is not stalled, the key on timer is less than 5 seconds (to prime the line at key-on), or the pedal is currently pressed (to prime the line for service). Like the other actuators, fuel pump circuit faults are diagnosed. For 2017, a relay driver was used to drive a fuel pump relay, and diagnostics were not provided. For 2018, the fuel pump is driven directly from the ECU, and diagnostics are provided.

7. Thermal Manager (TM)

7.1. Glow Plug Control

Glow Plugs are used on Diesel engines to aid in cold starting. As Diesel combustion relies on high in-cylinder temperature and pressure to initiate combustion, assistance is often required when the engine is cold. Glow plugs are small heating elements located in the combustion chamber. Glow plugs are turned on via the glow plug relay. If the engine is not running, the glow plugs are turned on for a calibrated amount of time after key on, as a function of engine coolant temperature. If the engine is cranking, the glow plugs are turned on if the coolant temperature is below a calibrated threshold. If the engine is running, the glow plugs are turned on for a calibrated amount of time after the engine reaches run mode, as a function of coolant temperature.

7.2. Radiator Fan Control

Most snowmobiles are cooled using a track mounted heat exchanger, which is located on the underside of the track tunnel and cooled by snow thrown up by the moving track. As this particular snowmobile is utility focused, it also includes an air cooled radiator to allow for extended idle. To cool the radiator when the sled is not moving, it is equipped with a fan. The fan is turned on when either the air charge or coolant temperature is above a threshold, and turned off when both the air charge and coolant temperature are below another threshold.

7.3. Planned Enhancements

The cooling system is being mechanically redesigned for CSC 2018 and will include an electric water pump. Software changes are proposed to rewrite the thermal manager to coordinate engine warmup and cooling strategies with the electric water pump.

8. Injector Driver Board

8.1. Board Selection

The injector driver board is a NXP (Freescale) 33816FRDMEVM, based on the MC33816 precision solenoid driver IC. The IC contains 5 high-side MOSFET pre-drivers, 7 low-side MOSFET pre-drivers, and 4 current shunt amplifiers with current comparators, along with programmable microcores which may execute code to switch each gate based on current feedback, timer, or trigger events. While the MC33816 is extremely flexible in the driver topology which it may control, it must be programmed in a unique assembly language designed for low latency control.

The development kit included an NXP 'Freedom Board', with an MKL25Z family ARM processor, designed to mount directly to the MC33816 board. This microcontroller board is used in the development kit to pass along commands from the host computer over USB, allowing the user to manually configure the chip and trigger it using a GUI. New code was written for the MKL25Z to initialize the MC33816 code image and register settings with a fixed configuration, allowing the MC33816 to receive triggers from the Woodward control module directly.

8.2. MC33816 Background

The MC33816 smart solenoid driver features 12 N-channel MOSFET pre-drivers (5 supporting high side configurations), 4 current sense amplifiers with programmable comparators, and two logic channels. Each logic channel contains 1024 words of program RAM, 64 words of data RAM, and two microcores. The two microcore instruction clocks are phased to avoid memory contention. The MC33816 is designed to drive a high voltage boost supply (up to 70v) to supply the solenoids during the boost phase, and supports many driver topologies. The block diagram of the chip is shown below (from the datasheet).

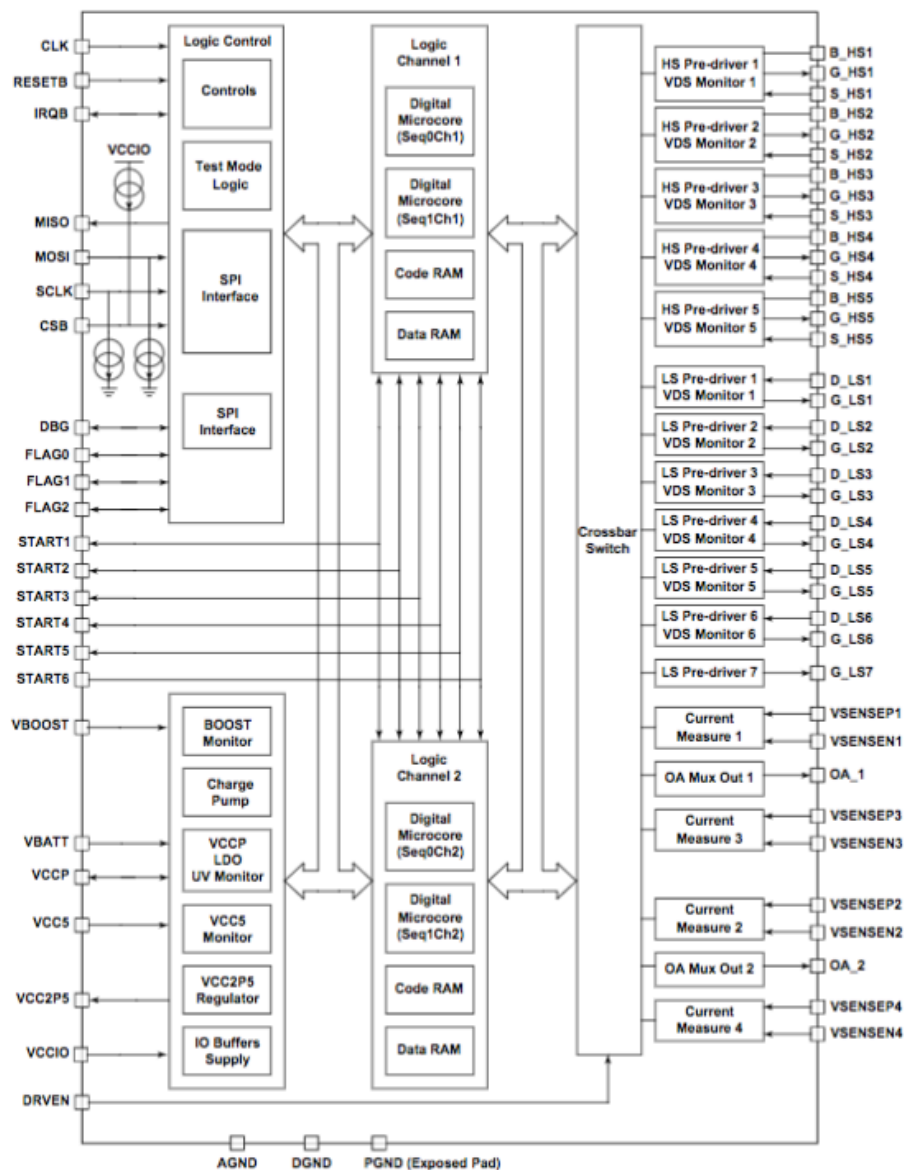
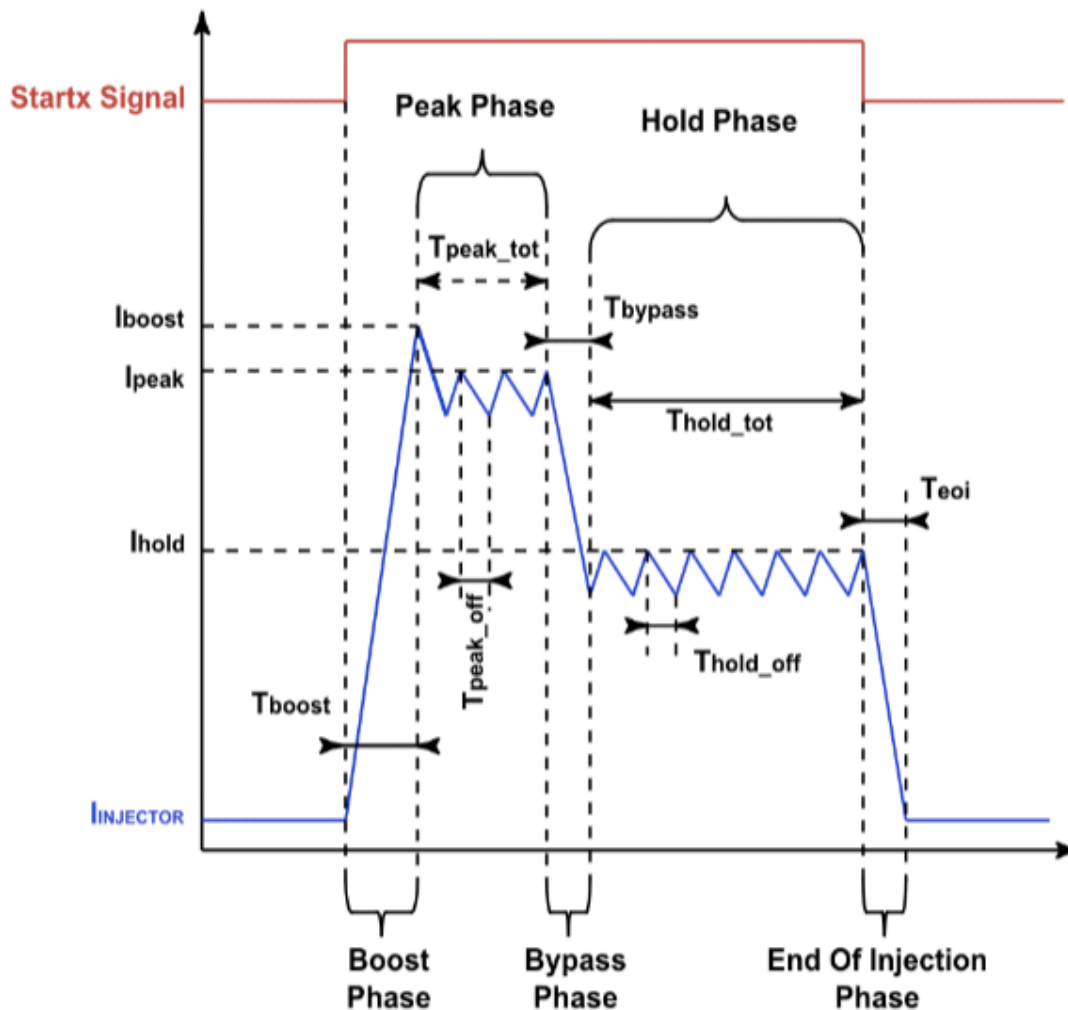


Figure 2. 33816 simplified internal block diagram

8.3. MC33816 Microcode

The microcode running on the MC33816 is based on Freescale app note AN4849 'Four Injector and Fuel Pump Drive'. As the app note was designed for a gasoline direct injection application, and was modified to suit the Diesel injector requirements. The fuel pump drive features are not used. The project was also modified to compile in MC33816 Studio, as the the original app note was designed to be compiled using a command-line assembler. As the



MC33816 provides only 1024 words of program memory and 64 words of data memory, among other limitations, it must be programmed in a special assembly language. The MC33816 CPU provides a 'wait table' system allowing hardware assisted event driven programming, where an execution thread may schedule up to 5 'events' to trigger based on hardware conditions, and specify the address to execute when the event occurs. Since the MC33816 CPU does not support interrupts, it must be programmed using short threads of execution which complete a single task and end in a wait instruction, causing the CPU to idle until an event condition occurs. The current profile is triggered from the START pin (driven by the Motohawk control module) and is split into 5 current regulation phases, as shown below.

8.4. Freedom Board MKL25Z Code

The code running on the Freedom Board MKL25Z was written in C and compiled in NXP CodeWarrior for MCU. NXP Processor Expert was used to generate boot and peripheral interface code, however the generated SPI code was found to not work reliably as it used asynchronous DMA, and it was rewritten as a synchronous routine. The MC33816 assembler generates a code image as a C source file which contains const arrays for each memory region, as well as example code to initialize the device. This code was modified to use the SPI and digital control functions for the MKL25Z. As the MC33816 requires a 1Mhz clock signal, this is generated using a PWM timer in the MKL25Z. In addition, the RGB LED on the Freedom Board is used to indicate the boot status of the MKL25Z and MC33816. Once the MC33816 is initialized and started, the MKL25Z waits indefinitely. The initialization process takes less than a second.