

**Engine Control Workstation Using Simulink / DSP  
Platform  
(COOLECW)**

**Functional Description and System Block Diagram**

By:

Mark Bright

and

Mike Donaldson

Project Advisor:

Dr. Gary Dempsey

Date:

Dec 12, 2008

## Introduction

Cooling control systems used in vehicles have become more sophisticated in recent years due to demands for lower fuel or energy consumption. A thermostat has been the common control element but better accuracy, system speed, and minimization of fan and pump energy can be achieved with a proportional actuator. Multiple temperature sensors can also improve system performance.

An Engine Control Workstation will be designed to simulate the thermal environments found in liquid-based cooling systems. The workstation will allow users to design, test and implement controllers to more precisely regulate the thermal dissipation of a motor-generator system with the goal of reducing energy use. A user friendly GUI for temperature and engine control will be designed using MATLAB and Simulink software. Workstation controller and monitoring application software will be auto-code generated within the same program with the processing of the data being done on a DSP board.

## Project Goals

Engine Control:

- (1) Learn software packages for auto-code generation and real-time control via Simulink/DSP interface. A major focus will be minimization of C-code and execution time.
- (2) Design DSP/cooling system hardware interface
- (3) Obtain a transfer function of the cooling system at different operating temperatures.
- (4) Design DSP/motor hardware interface – will build off 2008 Mini Project
- (5) Design software for PWM generation and velocity calculation from rotary encoder.
- (6) Design active thermal load for DC generator.
- (7) Design closed-loop controllers for velocity and acceleration control.
- (8) Determine the limitations of the Simulink/TMS320F2812 DSP interface in terms of real-time execution and program memory.
- (9) Evaluate controller performance based on system accuracy, speed, and energy use.
- (10) Design Simulink/MATLAB GUI for controller parameter modification. This will include ability to graph critical output data for controller design evaluation.

### Thermal Control:

- (1) Design closed-loop controller for temperature regulation of cooling system.
- (2) Design Simulink/MATLAB GUI for controller parameter modification. This will include ability to graph critical output data for controller design evaluation.
- (3) Design energy management control system in Simulink to regulate voltage/current to each subsystem based on its energy usage.
- (4) Evaluate controller performance based on system accuracy, speed, and energy use.
- (5) Determine the limitations of the Simulink/TMS320F2812 DSP interface in terms of real-time execution and program memory.

### Functional Description

The proposed engine control workstation will consist of a total of 5 subsystems: (1) motor-generator system to mimic an engine, (2) an active thermal load for the engine, (3) a liquid cooling system consisting of a pump, tank, radiator, fan, cooling block, flow meter, and multiple temperature sensors, (4) a fixed-point DSP evaluation board and software for controller implementation, and (5) interfaced electronics.

The system will interface with the user via a GUI developed using Simulink and MATLAB. There will be several parameters that the user will be able to input and simulate, with the hopes of designing an engine due to the user's desired specifications. These inputs are, but not limited to: command velocity, controller parameters, controller types – proportional, integral, derivative, (P, PI, PID) and their perspective values for  $K_p$ ,  $K_i$ , and  $K_d$ , and load changes to the system.

Based upon these inputs, system identification can be performed to obtain a mathematical model of the system. The developed closed-loop controller for velocity and acceleration will be active for load and no load conditions on the system. This will be implemented using auto-code generation and real-time control in Simulink interfaced with a fixed-point 32-bit TMS320F2812 DSP Evaluation board. The outputs of the system are, but not limited to: output velocity, motor current, SS error, transient response – analyzing down to overshoot percentage, settling time, and time to first peak, PWM percentage, and controller signal. These outputs will be organized on a GUI that group these outputs in order to analyze desired specifications or continue design iterations.

From a hardware level, the system will also require a DSP to motor interface utilizing similar design techniques as in the 2008 Mini Project. Also, a deliverable of this project is to explore all areas of Simulink and DSP Evaluation board interfacing and document controller limitations, performance, and

enhancements needed. From this another goal was created, to minimize both C-code and its execution time – down to an assembly level to ensure there are no possible real-time execution constraints created by the Simulink auto-code generation.

The Thermo control portion of the project will focus on developing controllers for the pump and the fan of the system. Multiple types of controllers are to be developed and implemented. These controllers will then be used to control the temperature of the coolant and/or the plant to a specific value.

In order to develop controllers to control temperature it is crucial to develop a transfer function and model of the thermo portion of the project by system identification.

### Plant/Engine Control GUI parameters

<b>Input</b>	<b>Output</b>
Command Velocity	Velocity
Controller Parameters	Motor Current
Load	Steady State Error
	Transient Response
	PWM %

### Thermo Control GUI parameters

<b>Input</b>	<b>Output</b>
Set Point (temp)	Flow Rate
Pump Velocity	Radiator Out/Inlet Temp
Fan Velocity	Engine/Plant Temp
	Motor Current (Pump/Fan)
	PWM % (Pump/Fan)

## System Block Diagrams

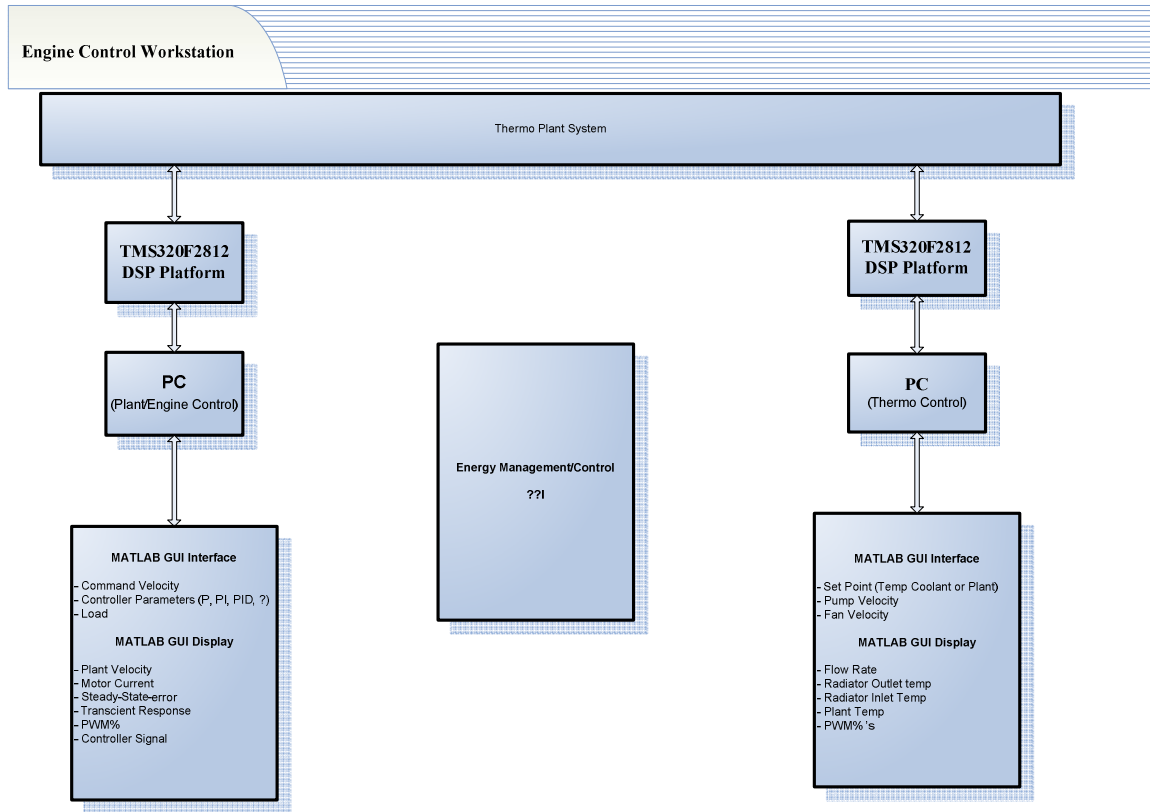


Figure 5-1 Engine Control Block Diagram

This overall block diagram is a look at the users input and output parameters. Input will go into the Simulink GUI, then sent through the Simulink/DSP board interface, to the DSP board for computation, then back through the same route in order to be displayed on the output GUI.

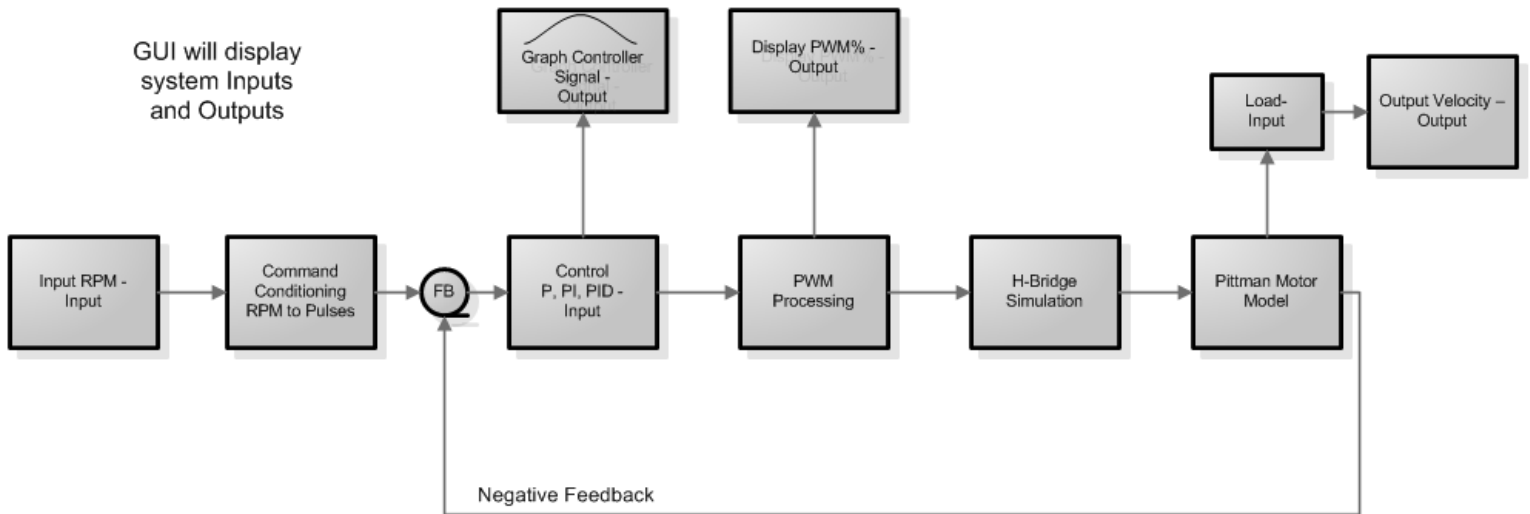


Figure 6-1 Engine Control Software Flowchart

As per the Functional Description, the software portion will be created using both Simulink auto-code generation feature and C-code. Here, the block diagram is very similar to a model that would be setup in Simulink containing all crucial system inputs and outputs. As stated, these will be calculated on the DSP board and then displayed/graphed on the GUI. The internal signals of these blocks, including their parameters and transfer functions will be derived through system identification.

# Thermo Plant System

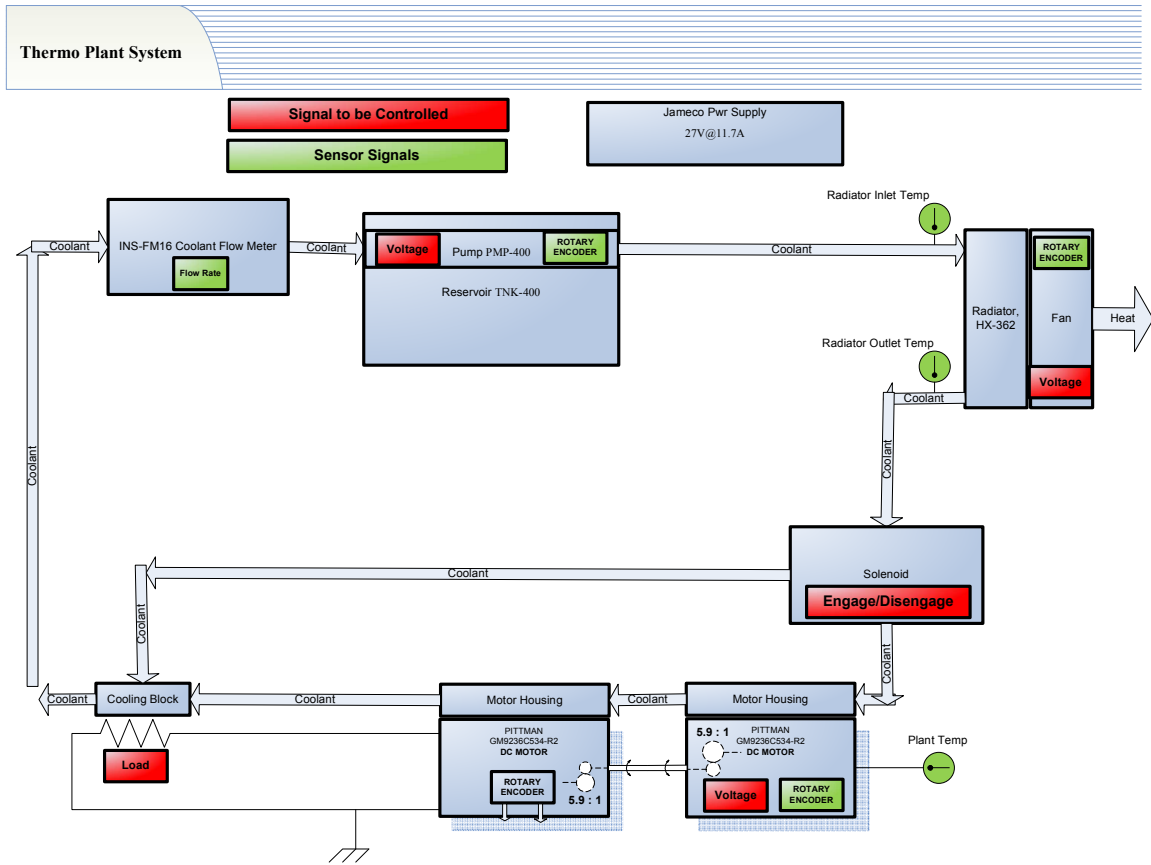
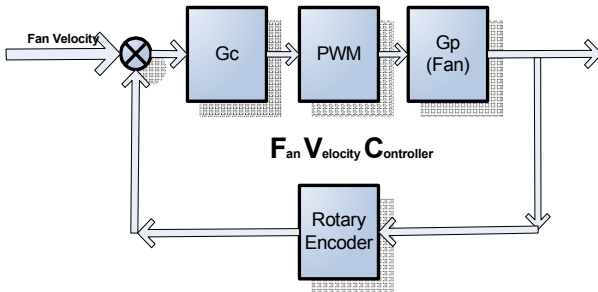


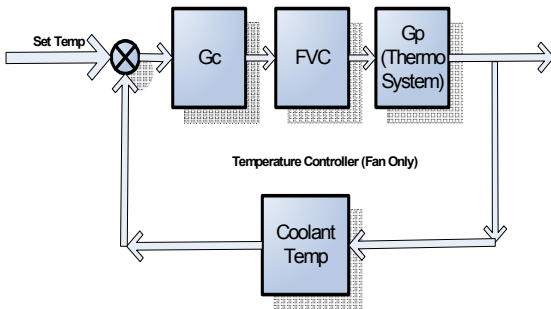
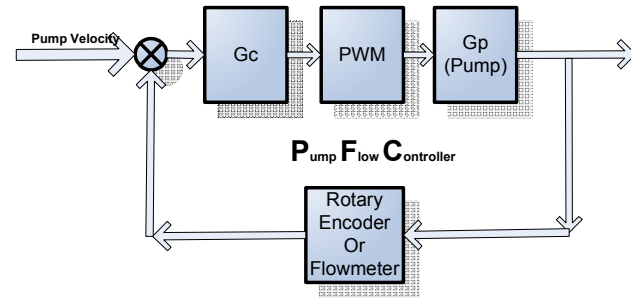
Figure 7-1 Cooling System

Below are the descriptions of the individual blocks within this flowchart. They include: the fan velocity controller, the pump flow velocity controller, and both temperature controllers.



The Fan velocity controller will be a negative feedback closed-loop. The rotary encoder will be used as the feedback signal. Various controllers ( $G_c$ ) will be developed.

The Pump flow velocity controller will be a negative feedback closed-loop. The rotary encoder or flowmeter will be used as the feedback signal. Various controllers ( $G_c$ ) will be developed.



One controller for temperature will use the fan to pull heat from the radiator. This controller will use the Fan Velocity controller within the negative feedback loop.

The second controller for temperature will use the pump to circulate coolant through the system. This controller will use the Pump Flow controller within the negative feedback loop.

