Observer-based Engine Cooling Control System (OBCOOL)

Final Report

Students:
Andrew Fouts & Kurtis Liggett

Advisor:
Dr. Gary Dempsey

Date:
May 3, 2011

Abstract

The primary goal of our project was to research and implement an observer-based control system proposed by George Ellis in his book, Observers in Control Systems. Ellis's observer is unique in that it can be tuned with a classical PID controller. The plant for evaluating the controller designs was a small scale engine cooling system. The controllers were implemented on two Texas Instruments TMS320F2812 DSP Evaluation boards. One DSP was used for engine control and the other DSP for temperature regulation of the cooling system.

The performance evaluations of the observer-based control systems were compared with conventional controllers designed with root locus and frequency domain methods. Additional topics addressed in the project included plant system identification, sensor linearization, anti- windup control, feed-forward control, engine power governor, energy management and safety features for cooling system, and auto-code generation using Simulink, Real-Time Workshop, and Code Composer software.

Table of Contents

Introduction

Control systems exist in many applications today, from home thermostats and vehicle cruise controls to engine temperature regulation and missile-guidance systems. Many control system designs exist, and one of the newer, more sophisticated concepts in modern control systems is the concept of observers. Observers are algorithms used to predict a system's internal response. While complex, observers are a powerful addition to a control system and greatly improve the system's performance [1].

Our project will consist of researching observer-based control systems and applying this knowledge to design closed-loop controllers for velocity control and temperature regulation of an engine cooling system. The controllers will be implemented using DSP boards with Simulink auto-code generation. In addition, energy management and controller performance will be evaluated.

System Description

The engine control workstation consists of the following subcomponents:

- Engine model simulated by a motor-generator system
- Variable load for engine
- Cooling system consisting of a fan, radiator, cooling block, reservoir, pump, flow meter, and three temperature sensors
- Two eZdsp F2812 DSP boards
- PC software GUI (MATLAB/Simulink and Code Composer)

A closed-loop control system will be implemented for both the engine system and the thermal system. While the initial control systems will be developed using EE 431 ("classical") control methods, the final system will incorporate observers to improve the systems' responses.

The overall system functions as follows:

1. Using the PC GUI, the user sets the system inputs (engine RPM, etc.) for the engine.
2. The PC sends data to the DSP boards through Code Composer.
3. The engine control DSP board sets the engine RPM to the desired value using the implemented control algorithm and PWM signals.
4. The thermal control DSP board adjusts the temperature of the engine by changing the pump & fan motor speeds using the implemented control algorithm and PWM signals.
5. The engine control output information and the thermal control output information are sent back to the PC and are displayed in the GUI.

A table of inputs and outputs can be found in Appendix A. A list of equipment used can be found in Appendix B.

System Block Diagram

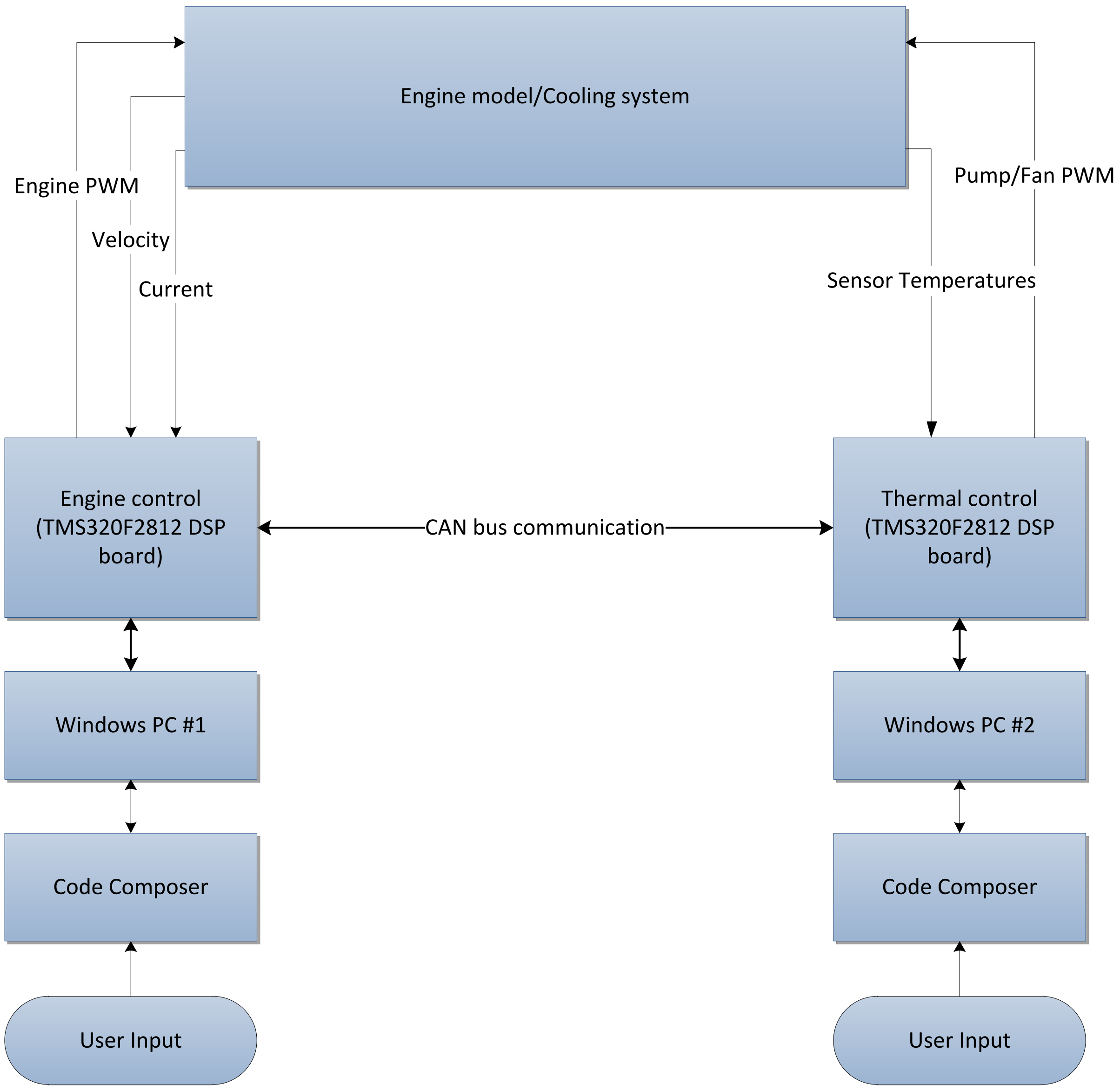


Fig. 5-1
Overall Block Diagram

The overall system consists of the plant (engine/cooling system), the engine & thermal controls (DSP boards), and two Windows PCs with Code Composer interfaces. The user’s input will be sent to the DSP boards for processing. After the boards have executed the user’s commands, the resulting output will be sent back to the Code Composer interface and displayed.

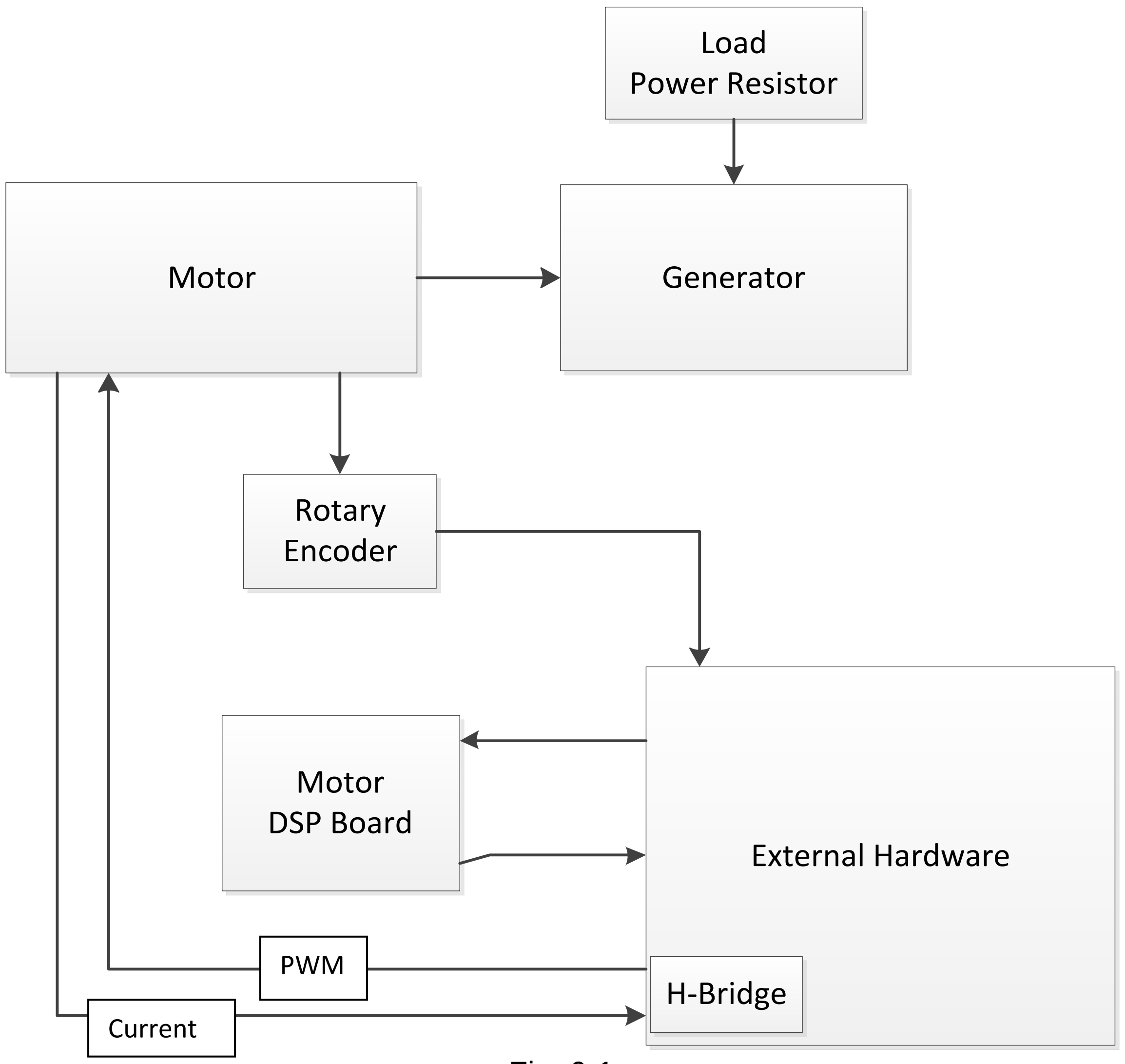


Fig. 6-1
Engine Block Diagram

The engine subsystem includes the motor, generator, load, rotary encoder, external hardware, and one DSP board. [2]

- The load is simulated using a power resistor. This will load the generator, which will load the motor. The system will be designed to accommodate varying loads.
- The rotary encoder is used to detect the speed of the motor, which will be used in the observer calculations
- The H-bridge provides a means to control the motor using a PWM signal from the DSP board.
- The DSP board allows computations to be done quickly. The observers will be implemented in software on the DSP board.

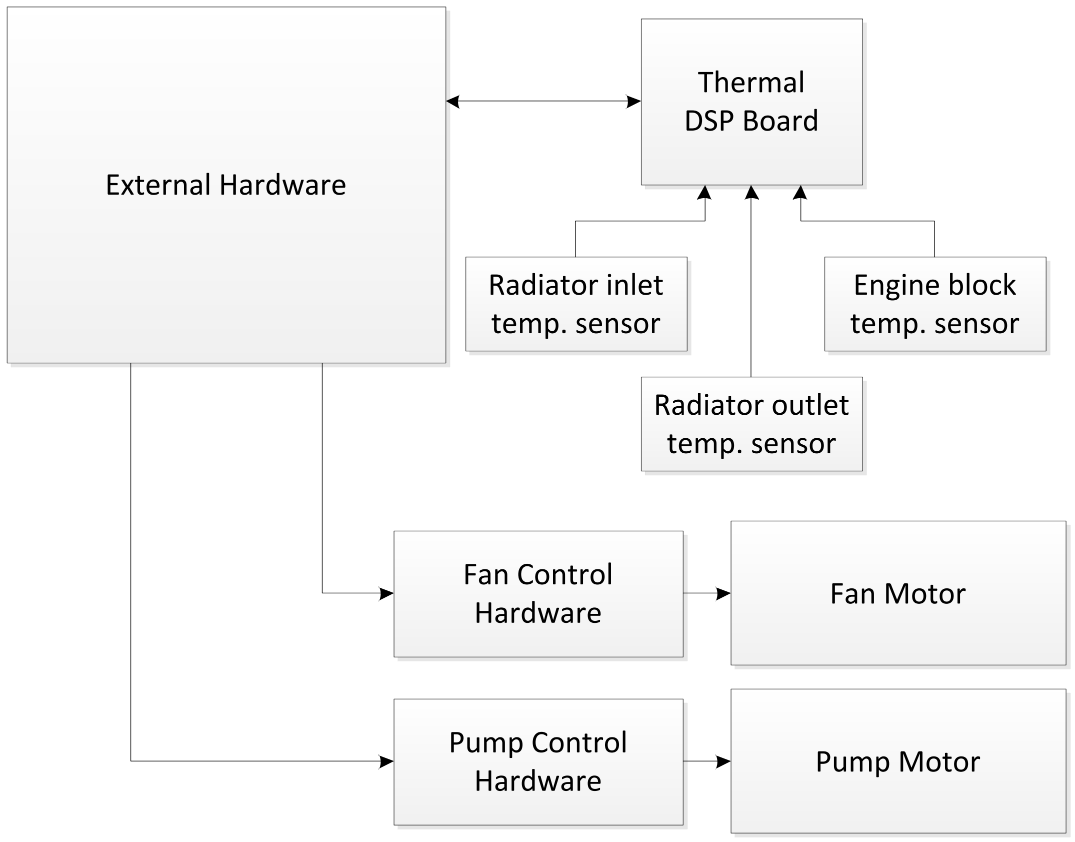


Fig. 7-1
Thermal Control Block Diagram

The thermal subsystem includes the fan & pump motors, hardware for controlling each motor, three temperature sensors, and one DSP board. [2]

- The temperature sensors each contain one thermistor for measuring the temperature. The thermistor's resistance varies with temperature, causing the voltage output of each sensor to change.
- The DSP board converts the voltage levels from the temperature sensors into digital values and calculates the required fan/pump motor speeds required to cool the system.
- The DSP board outputs a PWM signal (through the external hardware) to the fan/pump motors and adjusts their speed.
- The DSP board allows computations to be done quickly. The observers will be implemented in software on the DSP board.

Functional Requirements

Engine control system

The engine control system will go through multiple designs. A basic proportional controller will be implemented first, followed by PI & PID controllers. The final controller will be observer-based.

The final engine control system shall meet the following specifications using a step input:

- Steady-state error = ± 5 RPM
- Percent overshoot ≤ 10%
- Rise time ≤ 30 ms
- Settling time ≤ 100 ms
- Phase margin = 45°

The data for these specifications will be collected for each method of control. This data will then be compared to make conclusions on the advantages and disadvantages for each control method. Each control method will be implemented in the engine control system. Both theoretical and experimental data will be collected. The control method command input range will vary based on the method used.

Thermal control system

The thermal control system will go through several design iterations. A basic proportional controller will be implemented first, followed by PI & PID controllers. The final controller will be observer-based.

The final thermal system shall meet the following specifications using a step input:

- Steady-state error = ±2° Celsius
- Percent overshoot ≤ 25%
- Rise time ≤ 2 seconds
- Settling time ≤ 10 seconds
- Phase margin = 45°

During system operation, the thermal control system shall ensure that the engine temperature remains below 40° C (104° F). The power consumed by the thermal control system shall remain at a minimum level. Each controller method listed above will be tested against the defined requirements. The method that best meets these requirements will be used in the final thermal control system.

System flowcharts can be found in Appendix C.

Analysis of Engine Results

The engine subsystem controller design went through several stages including proportional control, proportional integral control, feedforward control, and observer control. Each stage was analyzed and compared to determine in which cases any of the controllers would show advantages over the others.

The first controller that was designed was proportional control. Design methods from EE 432 were used to find the proportional gain for an overshoot of 10% because higher overshoot may damage the motor/generator coupling due to excessive vibration. The system was very fast, but the steady-state error was unacceptable.

Next, a PI controller was designed. This controller corrected the steady-state error, but it also slowed down the system. The system still met the settling time specification of 100ms.

A feedforward design method was used to speed up the system in the presence of a disturbance or small change. Under normal operating conditions, this controller resulted in the same output as the PI controller. This was expected because it reacts to system changes. With very small changes to the system, the feedforward controller does indeed speed up the system’s response times.

Finally, the observer controller was designed and implemented. This controller showed significant improvements only at a few velocities; however, with this model, a much cleaner feedback signal is expected. Also, current and various other engine states from the observer model that would otherwise be difficult or impossible to obtain can be found.

A table with rise time and settling time of each controller can be found in Table D1-2 of appendix D. The observer shows no clear speed improvements to the system. The advantage is obtaining the observed current signal to regulate the motor power.

Analysis of Thermal Results

Several different controller types were tested and evaluated using the thermal cooling system; these included proportional, proportional-integral, optimum phase margin, and observer-based methods. The performance of each controller was measured using both the models of the controllers and the actual systems. The pros and cons of each are discussed below.

The first controller implemented was a basic proportional controller. It was designed using tuning to find a working controller and was implemented before system identification. This was by far the simplest controller and the quickest to implement. The resulting controller was fairly poor, however; it operated in a “bang-bang” method of rapidly turning on and off.

After system identification, the proportional-integral controller was designed using the resulting model. The system model allowed for much quicker tuning before implementation into the actual system. The resulting controller was easy to implement but only produced moderate results.

The next controller was implemented using optimum phase margin (OPM) design. A lesser known design method, optimum phase margin controllers are designed for a target phase margin & frequency response shape. The goal of designing an optimum phase margin controller is typically to improve stability; however, the significant time delay greatly affected the design technique’s effectiveness. The resulting controller was actually much less stable than the other controller types, having a much higher percent overshoot in the step response. However, this controller was also much faster than the others.

The last control method implemented was an observer-based controller. Using the model found from system identification, an observer was constructed and implemented. As expected, the observer-based controller was the most stable of all of the controllers. In addition, the speed of the controller was acceptable as well; while not as fast as the optimum phase margin controller, the observer-based controller was about the same speed as the PI controller.

A model of the observer-based controller can be found in Figure E1-1, and a graph comparing the results of the PI, OPM, and observer-based controllers can be found in Figure E1-2.

Appendix A – System Tables

Inputs / Outputs	
Engine	
Inputs	**Outputs**
RPM	RPM
Power Limiter	Current
	Power dissipated
	Output power
	Observer RPM
	Observer current
Thermal	
Inputs	**Outputs**
Temperatures	Temperature of radiator inlet
	Temperature of radiator outlet
	Engine block temperature
	Power of each subsystem
	Observer temperatures

Table A1-1
System Inputs/Outputs

Appendix B – Equipment

The equipment used and costs for the project are listed below in Table 19-1. However, all of the equipment had already been purchased from previous projects. The only additional cost for this year's work on the Engine Cooling System is the cost of two copies of Observers in Control Systems by George Ellis ($118 each).

Parts Description	Cost
Fan	$11
Radiator	$29
Cooling Block	$55
Reservoir and Pump	$117
Flow meter	$20
Coolant	$15
Code Cathode	$11
Temp Sensors (4)	$40
Pittman Motors (2)	$160
Motor Heat Sinks	$20
Tubing, hose clamps	$10
30Volt, 315 Watt Switching Power Supply	$75
Advanced Motion Controls H-Bridge (6A) (donated)	$350
Control and Interfacing Circuitry	$30
eZdsp F2812 Texas Instruments DSP Boards (2)	$975
Sub-total	$1,918
Heat Sink Machine Shop Work 10 hours x $75/hr	$750
Cooling Station Construction 40 hours x $75/hr	$3,000
Software Installation 10 hours x $75/hr	$750
Sub-total	$4,500
Total	$6,418

Table B11-1
Engine Cooling System Equipment & Costs

Appendix C – System Flowcharts

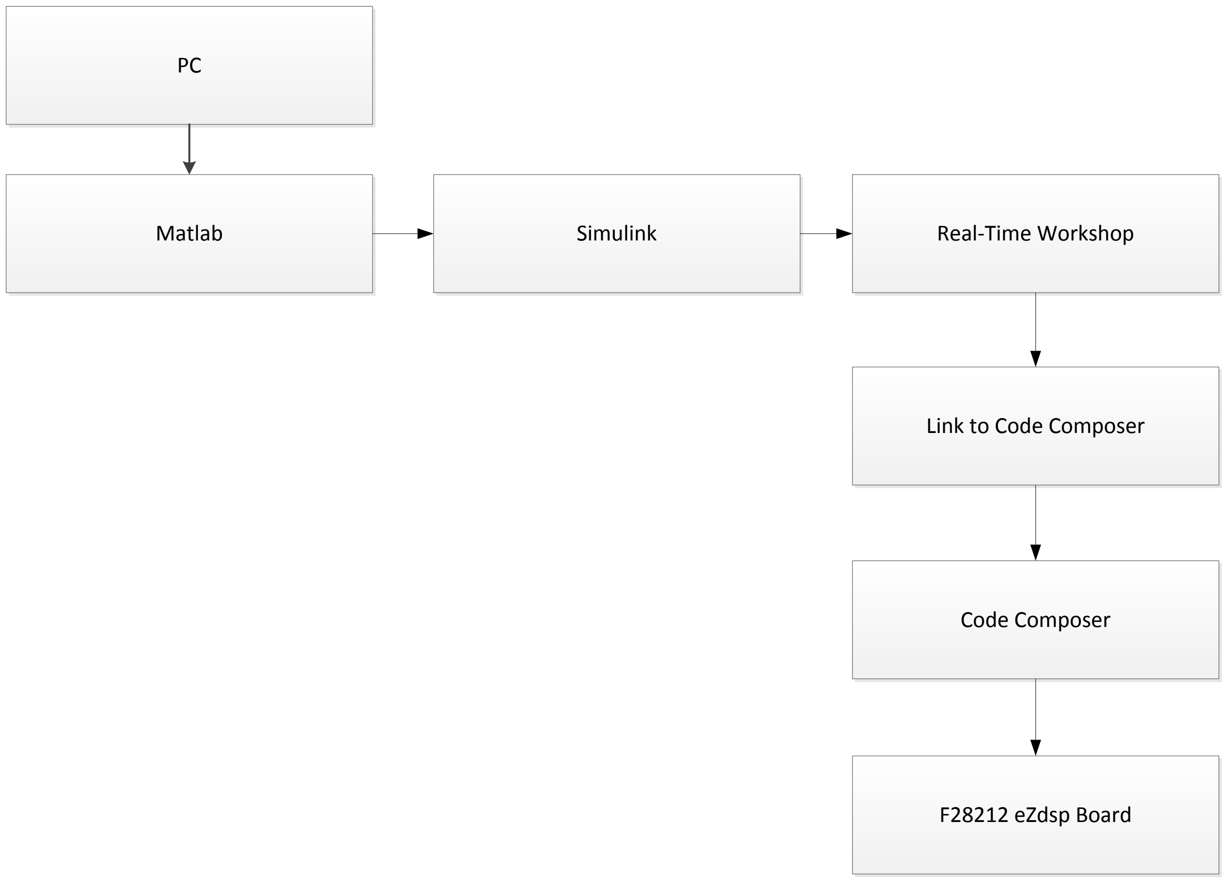


Fig. C1-1
General Software Flow Chart [2]

The eZdsp F2812 DSP board was used in both the motor control subsystem and the cooling control subsystem. The design and implementation of these control systems is done through the PC. The software packages that were used are MATLAB, Simulink and Realtime Workshop, and Code Composer 3.1.

- MATLAB is the main program associated with the project. It will be the host to other software, such as Simulink.
- Simulink is used to build the models of the subsystems.
- The Realtime Workshop is used to convert the Simulink model into C code using Code Composer.
- Link to Code Composer is used to link the Real-time Workshop to Code Composer

Appendix D – Engine Results

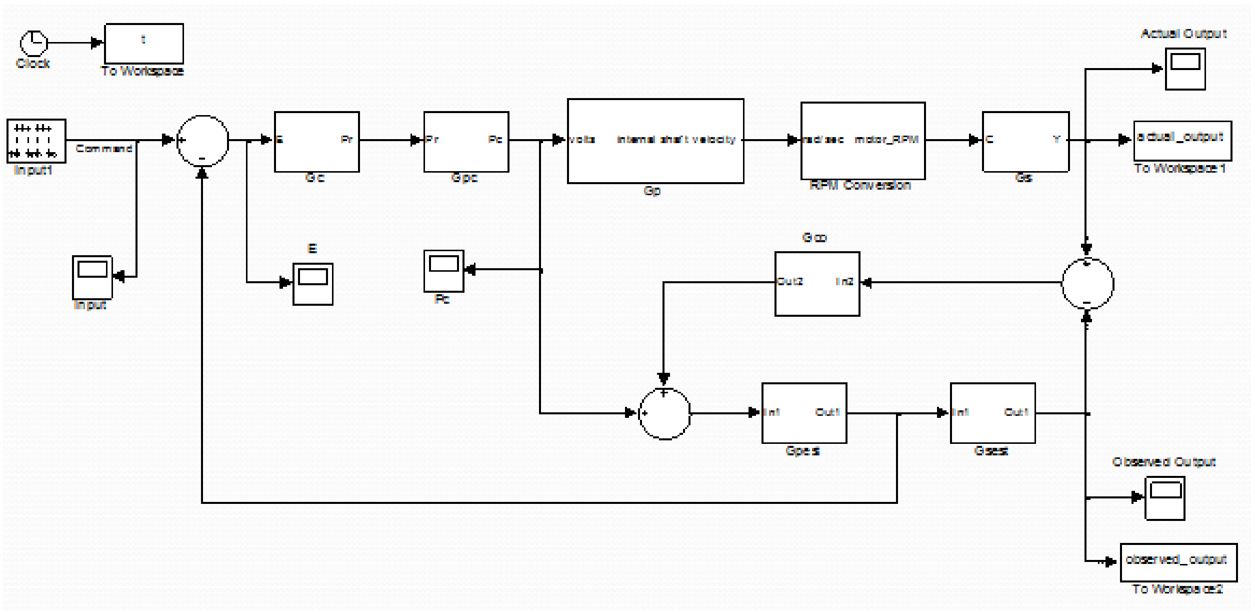
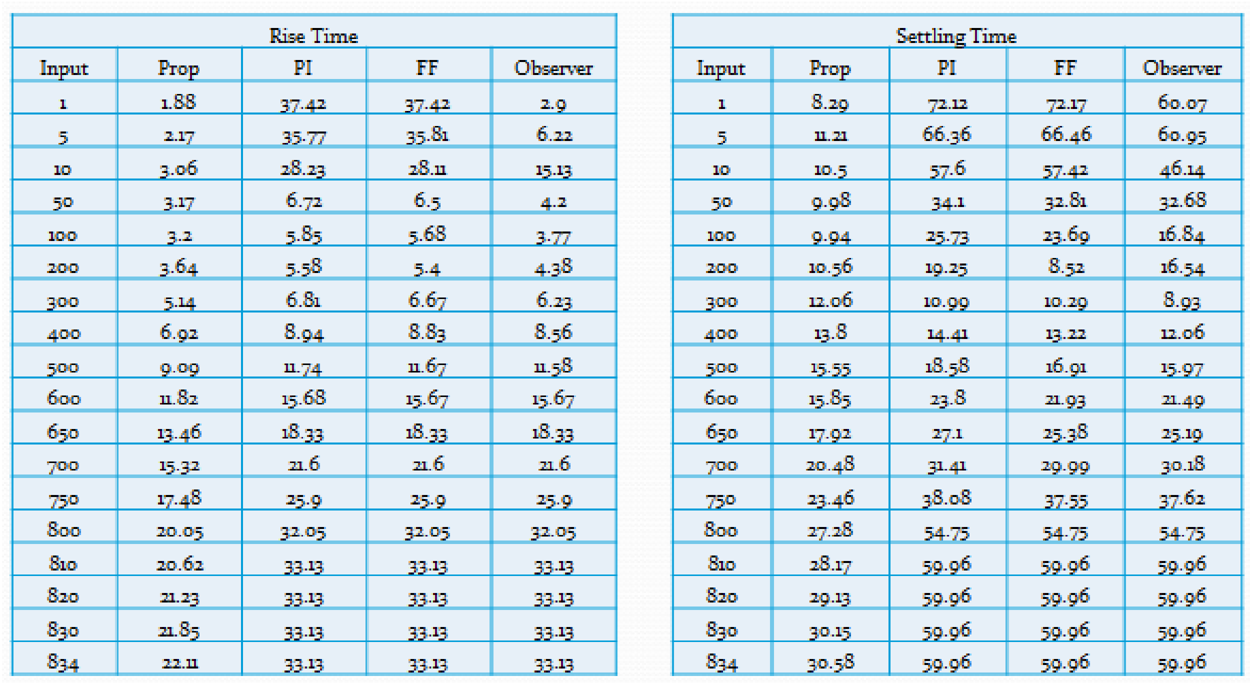


Figure D1-1
Engine Observer Control



Rise Time				
Input	Prop	PI	FF	Observer
1	1.88	37.42	37.42	2.9
5	2.17	35.77	35.81	6.22
10	3.06	28.23	28.11	15.13
50	3.17	6.72	6.5	4.2
100	3.2	5.85	5.68	3.77
200	3.64	5.58	5.4	4.38
300	5.14	6.81	6.67	6.23
400	6.92	8.94	8.83	8.56
500	9.09	11.74	11.67	11.58
600	11.82	15.68	15.67	15.67
650	13.46	18.33	18.33	18.33
700	15.32	21.6	21.6	21.6
750	17.48	25.9	25.9	25.9
800	20.05	32.05	32.05	32.05
810	20.62	33.13	33.13	33.13
820	21.23	33.13	33.13	33.13
830	21.85	33.13	33.13	33.13
834	22.11	33.13	33.13	33.13

Settling Time				
Input	Prop	PI	FF	Observer
1	8.29	72.12	72.17	60.07
5	11.21	66.36	66.46	60.95
10	10.5	57.6	57.42	46.14
50	9.98	34.1	32.81	32.68
100	9.94	25.73	23.69	16.84
200	10.56	19.25	8.52	16.54
300	12.06	10.99	10.29	8.93
400	13.8	14.41	13.22	12.06
500	15.55	18.58	16.91	15.97
600	15.85	23.8	21.93	21.49
650	17.92	27.1	25.38	25.19
700	20.48	31.41	29.99	30.18
750	23.46	38.08	37.55	37.62
800	27.28	54.75	54.75	54.75
810	28.17	59.96	59.96	59.96
820	29.13	59.96	59.96	59.96
830	30.15	59.96	59.96	59.96
834	30.58	59.96	59.96	59.96

Figure D1-2
Results of All Engine Controllers

Appendix E – Thermal Results

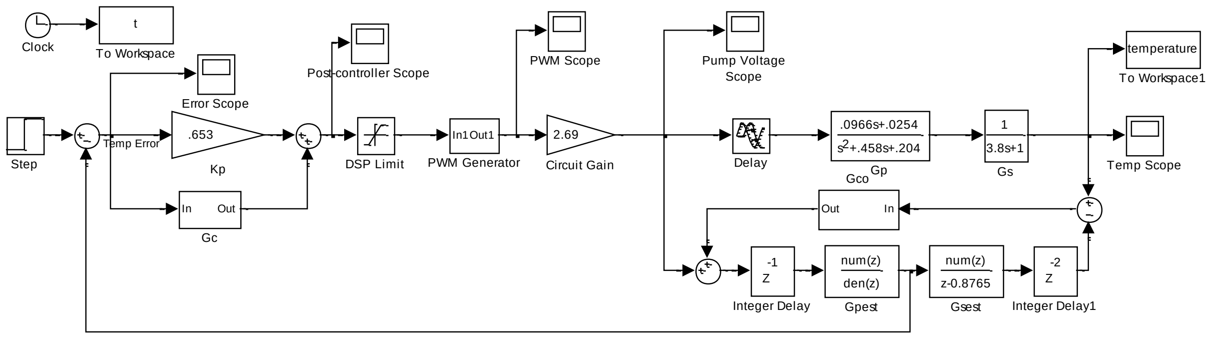


Figure E1-1
Thermal Observer Controller

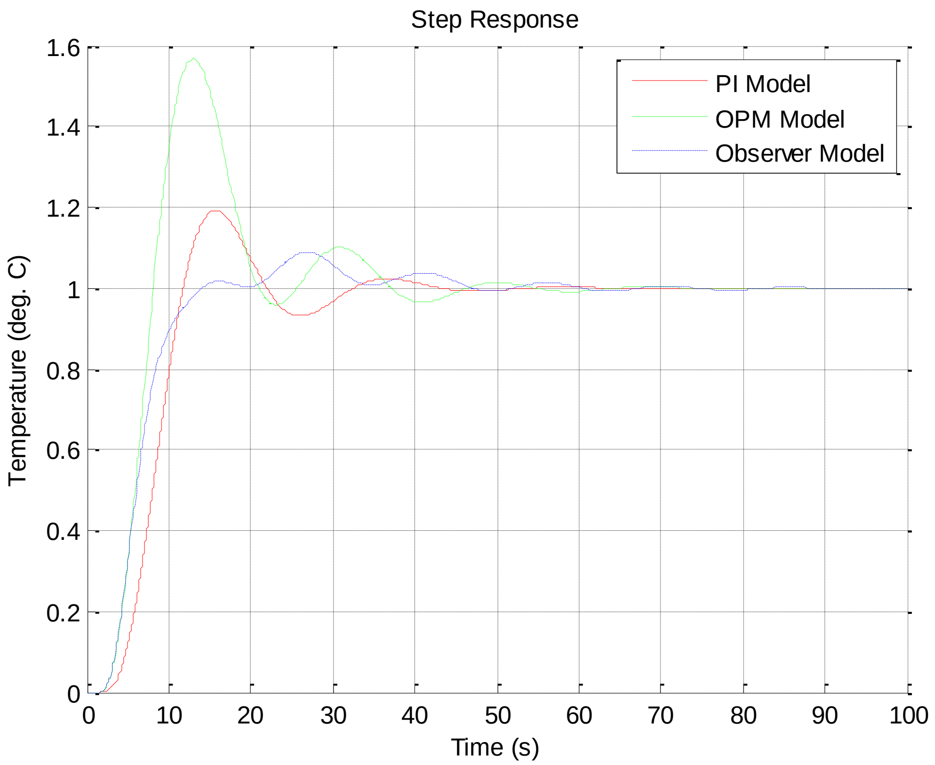


Figure E1-2
Comparison of controller step responses

Appendix F – References

[1] George Ellis. “Observers in Control Systems”, Academic Press, 2002.
[2] Gary Dempsey. “Engine Control Workstation System Overview”, Electrical and Computer Engineering Department, Bradley University, September 2010
[3] Gary Dempsey. “Observer-based Engine/Cooling Control System”, Electrical and Computer Engineering Department, Bradley University, August 2010