Energy Management System for Electric Engines Using Collaborative DSP Controllers (EMSEECDC)

Senior Project Report on Engine Control

Authors
Jacob G. Teague

Project Advisor
Dr. Gary Dempsey

Date
May 13, 2010

Table of Contents

Abstract

Engine, temperature, and energy management control system designs were applied to a small scale electric engine cooling system. Precision control of temperature and fast response to engine power dissipation changes were achieved by observer design and neural networks. Temperature data is exchanged via a Controller Area Network (CAN bus) interface between the engine DSP controller and the thermal DSP controller. These advanced controller methods and the inclusion of a communication data channel allow for better energy management.

Introduction

Project Goals

The objective of this project was to develop an energy management system for an electric engine and cooling system [1]. Last year, Mark Bright and Mike Donaldson developed a small scale engine control workstation for their senior project (COOLECW [2]). The task for this year's senior project (EMSEECDC) was to utilize this new workstation to develop an engine controller. An engine controller that not only generates the control signal (PWM) for the motor, but calculates power loss, communicates through a CAN bus, and prevents motor damage. (Note: All MATLAB SIMULINK models are located in Appendix A)

Specifications

The following specifications were set prior to developing the engine controller. A fast response to the desired RPM should have a settling time 30ms for no load, 100ms for full load, and a peak at 20ms. The response must be precise within $\pm$5 RPM for steady state error and should range from 0 to the motor's maximum velocity of 834 RPM. Overshoot should be restricted to 10% to limit overcompensation by the motor. The phase margin and gain margin should be 60^{o} and 6dB respectively due to the wide range of applied loads. The time delay of the temperature change is so great that the CAN bus transmit rate is 1s and the receive rate is 500ms. Below is a summary of these specifications (Fig 1.1).

System	Goal
Steady State Error	$\pm$5 RPM
Ts	30ms
Tp	20ms
% overshoot	10%
Phase Margin	60^{o}
Gain Margin	6dB
RPM Range	0 to 834 RPM
Data Rate to Thermal System	500ms
Data Rate to Engine system	1s

Figure 1.1 – Summary of Engine Control Specifications

System Overview

High-Level Diagrams

The energy management system (Fig 2.1) consists of five subsystems: a motor-generator system, a liquid cooling system, two fixed-point 32-bit TMS320F2812 DSP control boards, and interface electronics (shown in Figure 3.2). One DSP control board will be used for engine speed control, while the other DSP control board will be used to regulate engine temperature. A CAN bus interface will be designed to allow communication between the two DSP control boards. This feature will allow for the design of an overall control system that uses minimal energy. In addition, the data exchanged will provide precise temperature regulation with a fast response to system changes such as engine speed and external load.

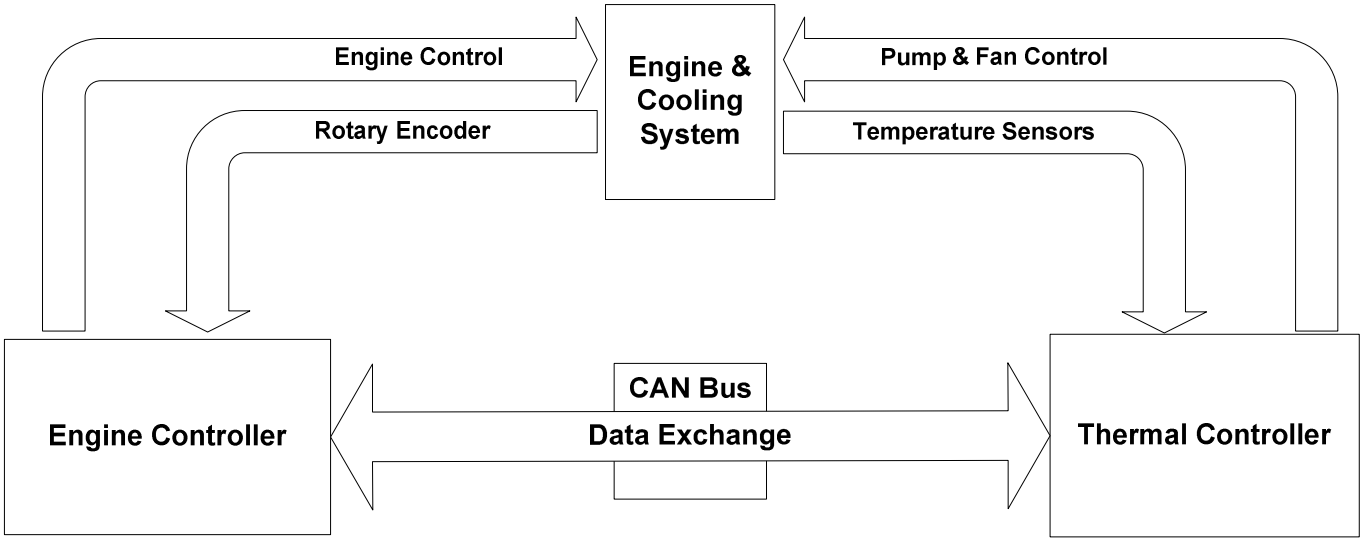


Figure 2.1 - Overall System Diagram

Components

The system contains a Pittman DC motor and a Koolance PC cooling system (pump, tank, radiator, fan, cooling block, flow meter, and multiple temperature sensors). These components simulate the electric engine and cooling system of an electric car (shown in Fig 3.1 & 3.2).

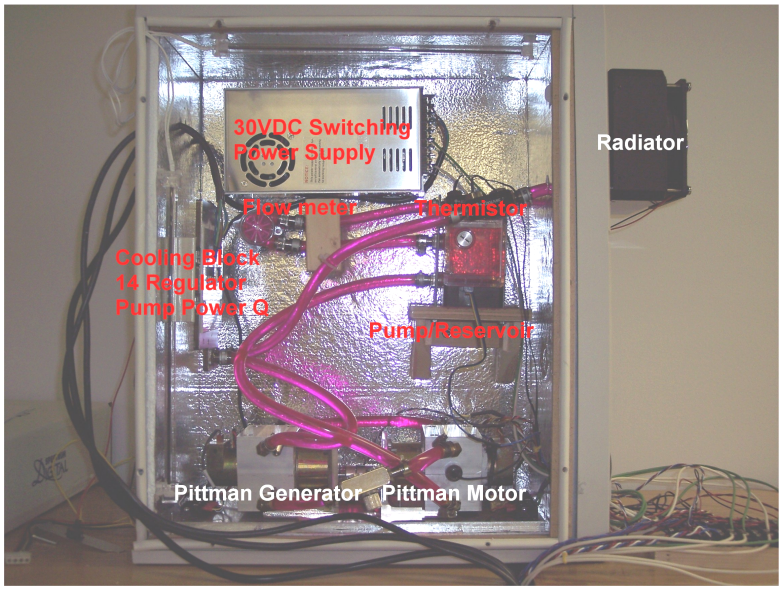


Figure 3.1 - Engine & Cooling System [2]

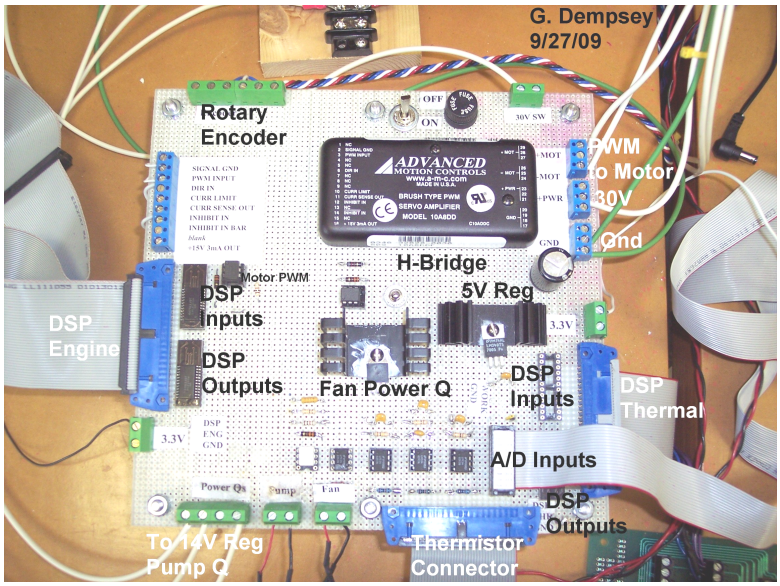


Figure 3.2 - Thermal & Engine Controller [2]

Thermal Controller & Engine Controller
The engine controller (Fig 4.1) will receive data from the rotary encoder and the CAN Bus to control the engine. The thermal controller (Fig 4.2) will receive data from temperature sensors and the CAN Bus to regulate temperature via the cooling system. The CAN Bus is crucial to the energy management of the motor. Power data and Thermal data will be passed between the two DSP boards.

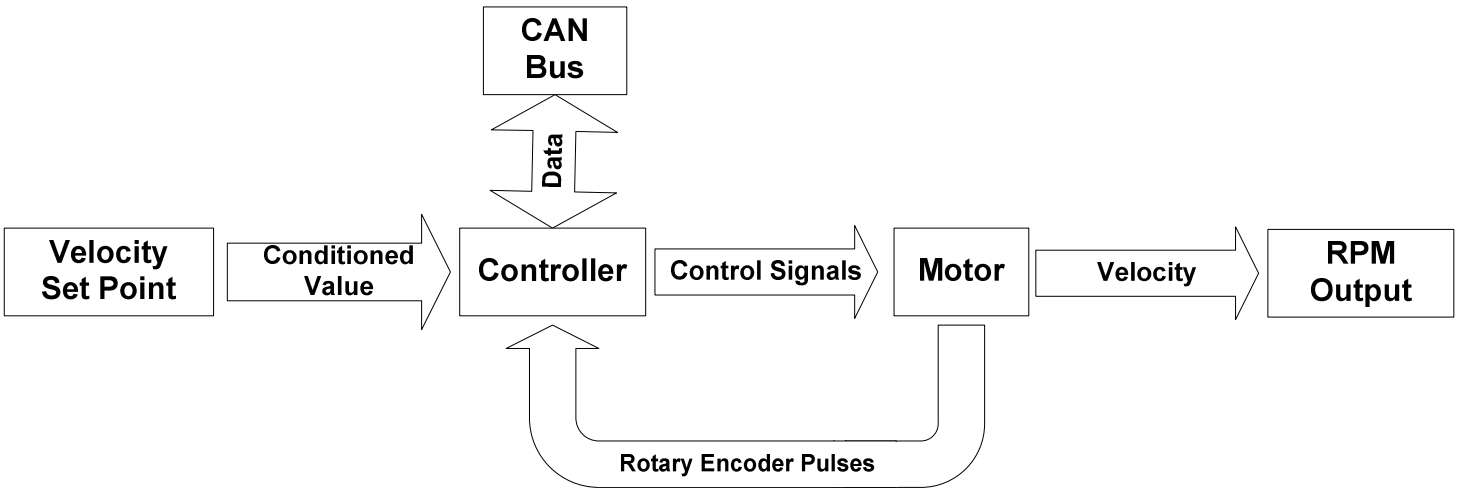


Figure 4.1 - Engine Controller

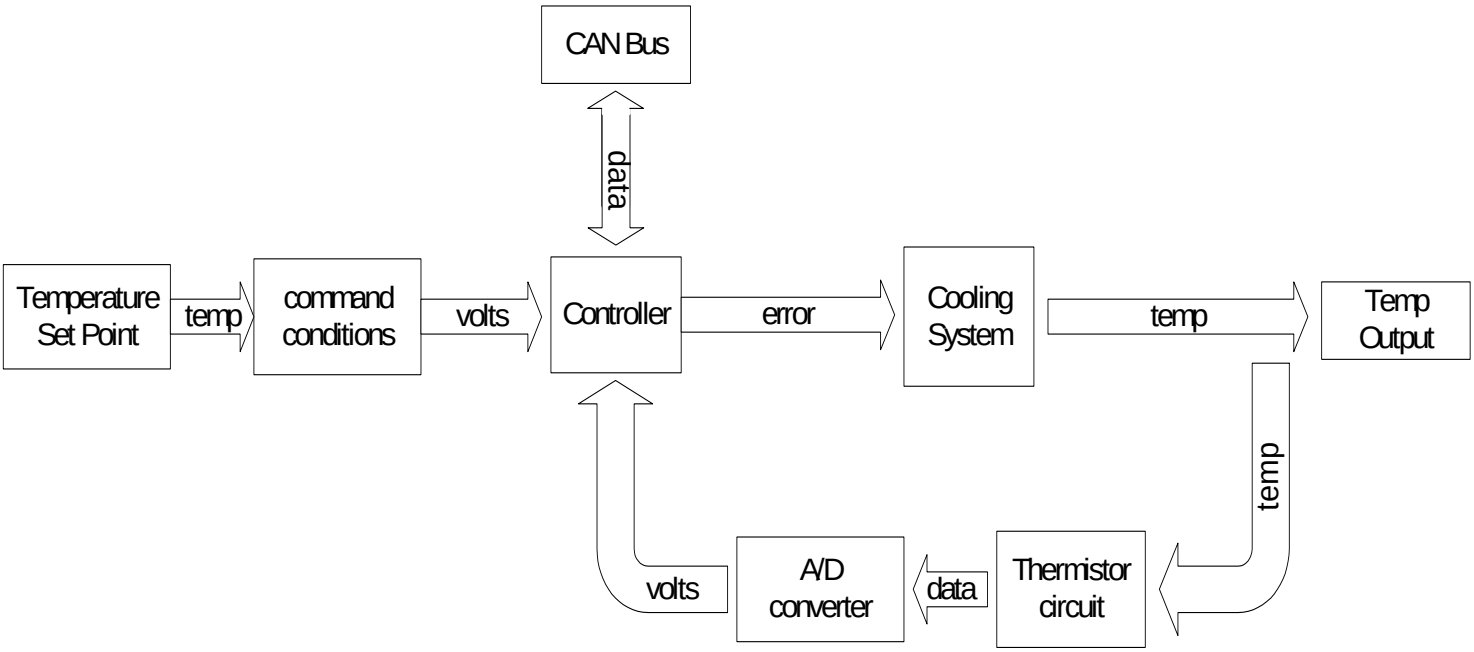


Figure 4.2 - Thermal Controller

Design & Modeling

Controller

Engine control was implemented with four control designs: Proportional (P), Integral (I), Lead Network (LN), and Feedforward (FF). P control did not meet any of the project specifications indicating the need for a sophisticated model. P & I control had great precision but a very poor response time. A small LN was implemented with the P & I control which yielded a much faster result that was within specifications. However, the final addition of FF control allowed the controller to further increase its speed. Therefore, the design that most effectively met specifications was the Proportional Integral & Lead Network (Fig 5.1).

The Bode Diagrams for the uncontrolled motor and the controlled motor are shown in Fig 6.1 & 6.2. Notice the improvements in all the desired specifications. Phase margin was increased by 70^{o}, gain margin by 16.51dB, and the crossover frequency decreased by a factor of 4. Although the motor was faster according to the last specification, there is a tradeoff of speed for stability.

Now that a controller has been determined, we need the sampling frequency for the controller. Since PWM converters are inexpensive in the range of 20 to 40 (20 being the least expensive and outputs an acceptably smooth PWM signal), I chose to calculate the sampling frequency using a PWM cost of 20 (Fig 7.1). As shown below (Fig 7.1), the appropriate sampling frequency is around 1kHz.

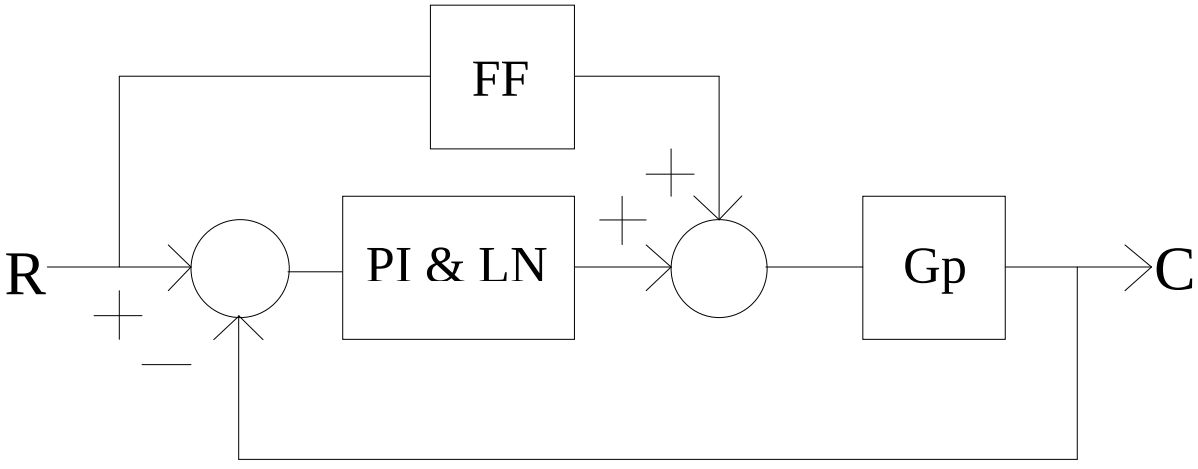


Figure 5.1 – Controller Design

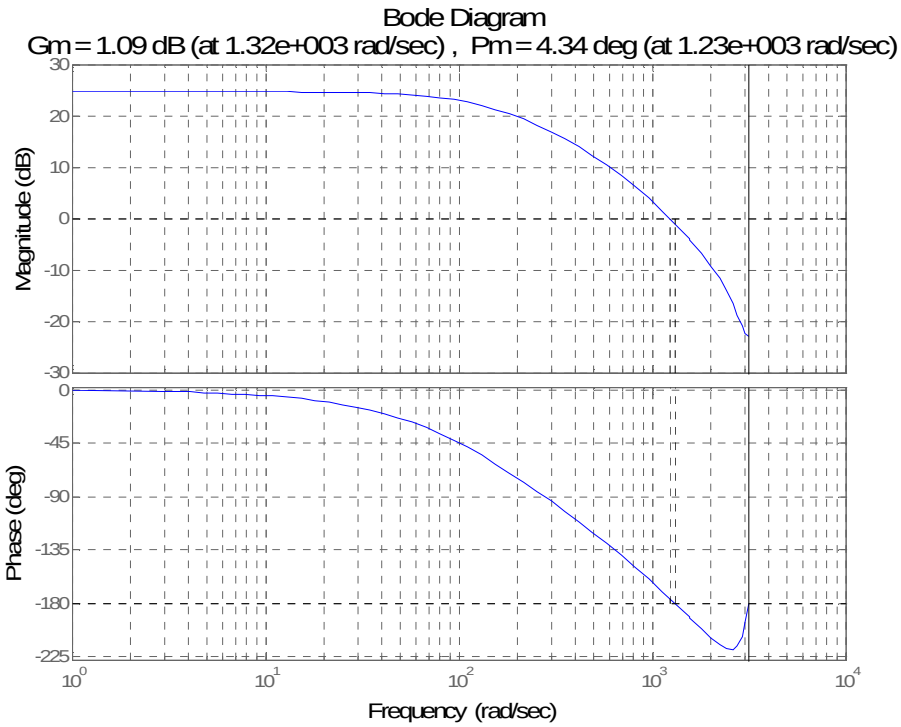


Figure 6.1 – Motor (Gp) Bode Diagram

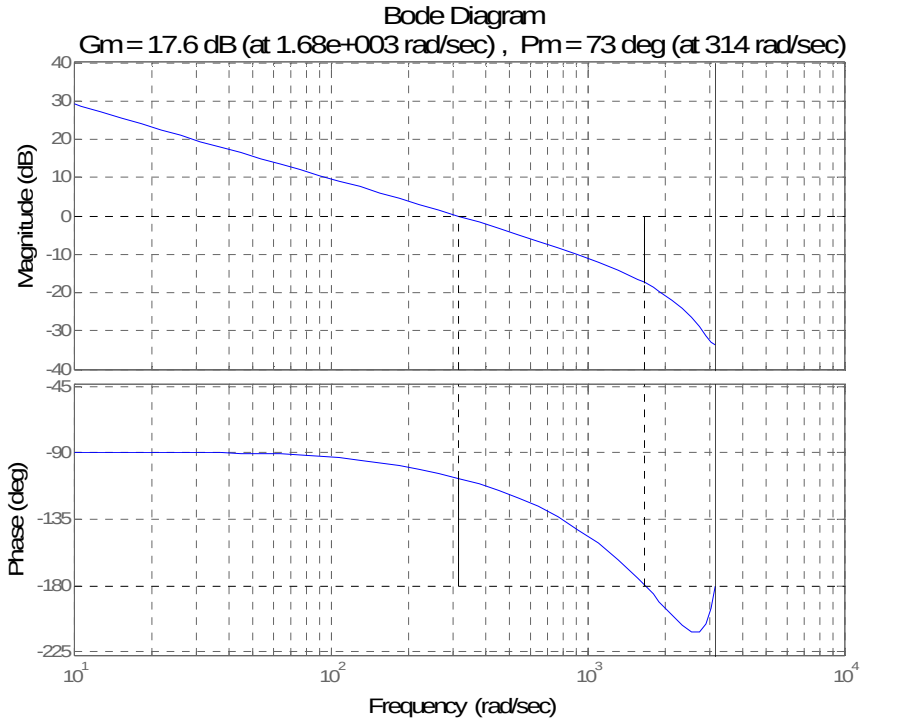


Figure 6.2 – Proportional Integral, Feedforward, & Lead Network Bode Diagram

$$PWM_{cost} = w_{samp} / w_{cross}$$

Figure 7.1 – PWM Convertor Cost

$$20 = w_{samp} /(314 rad / \sec)$$

$$w_{samp} = 6600\, rad / \sec$$

$$f_{samp} \approx 1\, kHz$$

Figure 7.1 – Sampling Frequency

Observer

An Observer Design (OD) uses the ideal model of the motor and the error in its output velocity to the actual motor as feedback to converge the ideal model to the actual model. There are two energy storage elements in the Pittman motor (Fig 8.1) that may be modeled: Inductor (L) and Inertia (J). After creating a block diagram to represent the motor's internal poles (Fig 8.2), factoring the poles into two integrators (Fig 8.2) yields the appropriate model for OD calculations (Fig 8.3).

The motor states are then represented by two state equations (Fig 9.1), one for each energy storage element. The input to the motor is a voltage (through PWM) that is integrated by the inductor to become motor current (state 2) which in turn is integrated to the motor velocity (state 1). An observability test must be performed to verify that the plant may use an OD (Fig 9.2). The reduced row echelon form of the observability matrix (θ) has a rank of 2 meaning both of the motor states are observable.

For each state the velocity error is fed back and multiplied by an L gain. Modifying the state equations for the motor plant to include the L gains L1 and L2 (Fig 9.3) and setting the determinant of those equations equal to the desired motor poles (usually 4 to 6 times motor poles due to sampling theory-the sampling frequency should be at least 2 times the system's operating frequency) yields the L1 and L2 feedback gains. The OD response to a 30.3V (834 RPM) input is shown in Fig 9.4.

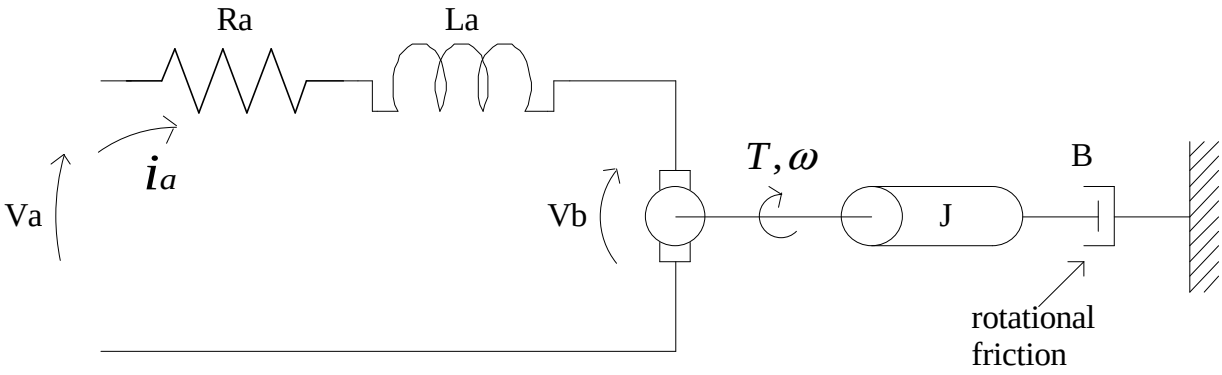


Figure 8.1 – Motor Schematic

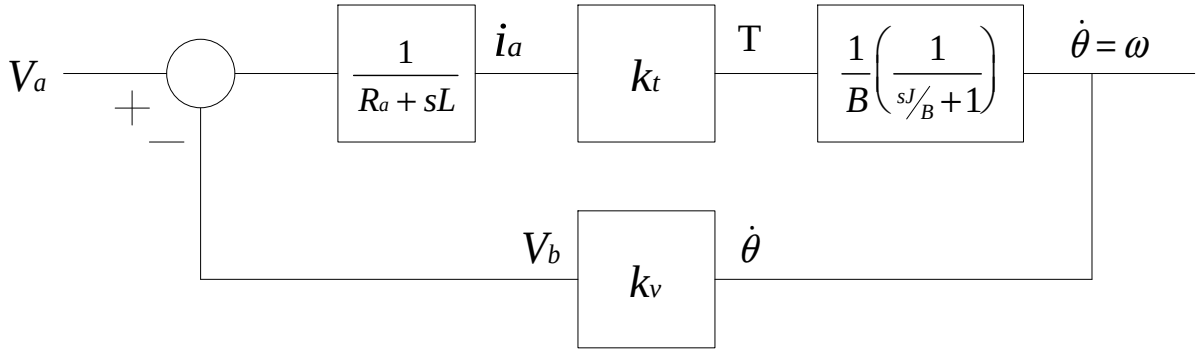


Figure 8.2 – Motor Block Diagram

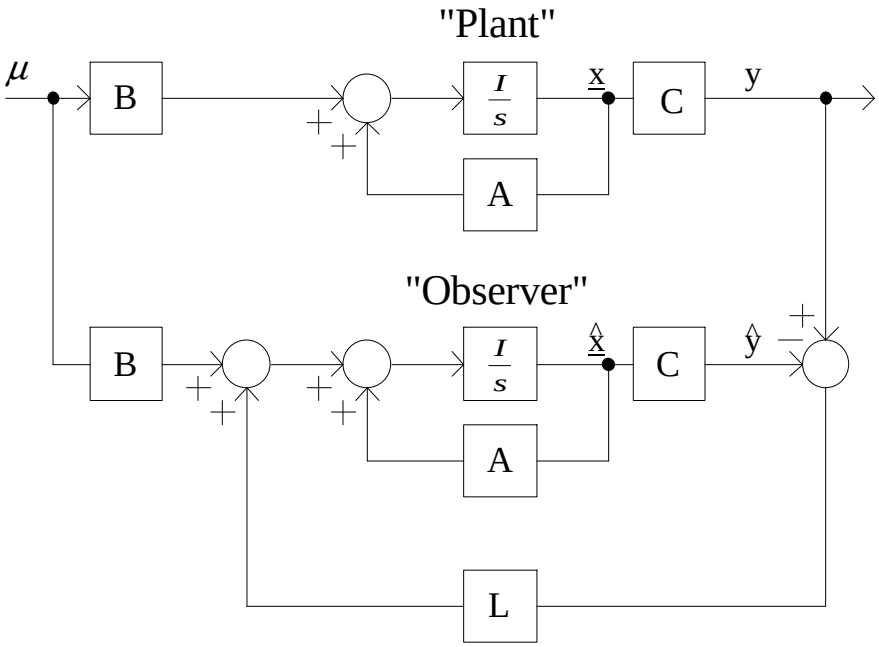


Figure 8.3 – Observer Design

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}-B/J & Kt/J\\ -Kv/La & -Ra/La\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}0\\ 1/La\end{bmatrix}u$$

$$y=\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+[0]u$$

Figure 9.1 – Plant State Equations

$$\theta=\begin{bmatrix}C\\ CA\end{bmatrix}=\begin{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix} & \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}-B/J & Kt/J\\ -Kv/La & -Ra/La\end{bmatrix}\end{bmatrix}^T$$

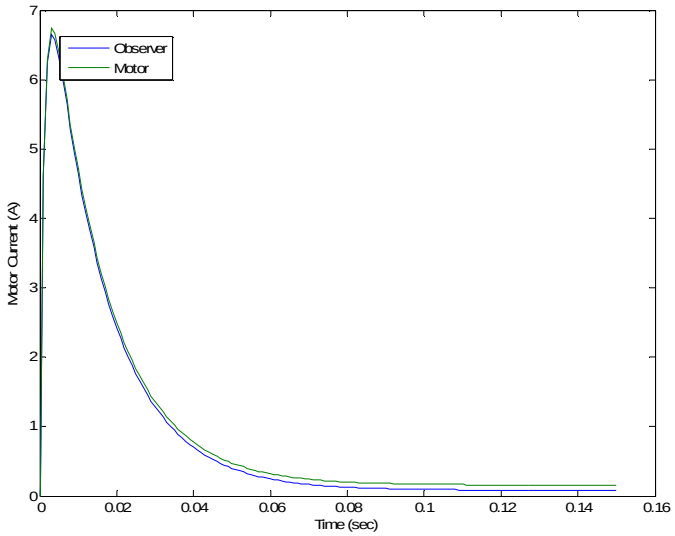
$$reduced\_row\_echelon\_form(\theta)=\begin{bmatrix}1 & 0\\ -0.5014 & 8243.63\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

Figure 9.2 – Observability Equations

$$\begin{bmatrix}\dot{\hat{x}}_1\\ \dot{\hat{x}}_2\end{bmatrix}=\left(\begin{bmatrix}-B/J & Kt/J\\ -Kv/La & -Ra/La\end{bmatrix}-\begin{bmatrix}L1\\ L2\end{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix}\right)\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\end{bmatrix}+\begin{bmatrix}0\\ 1/La\end{bmatrix}u+\begin{bmatrix}L1\\ L2\end{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

$$\begin{bmatrix}\dot{\hat{x}}_1\\ \dot{\hat{x}}_2\end{bmatrix}=\left(\begin{bmatrix}-B/J-L1 & Kt/J\\ -Kv/La-L2 & -Ra/La\end{bmatrix}\right)\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\end{bmatrix}+\begin{bmatrix}0\\ 1/La\end{bmatrix}u+\begin{bmatrix}L1 & 0\\ L2 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

Figure 9.3 – Observer State Equations



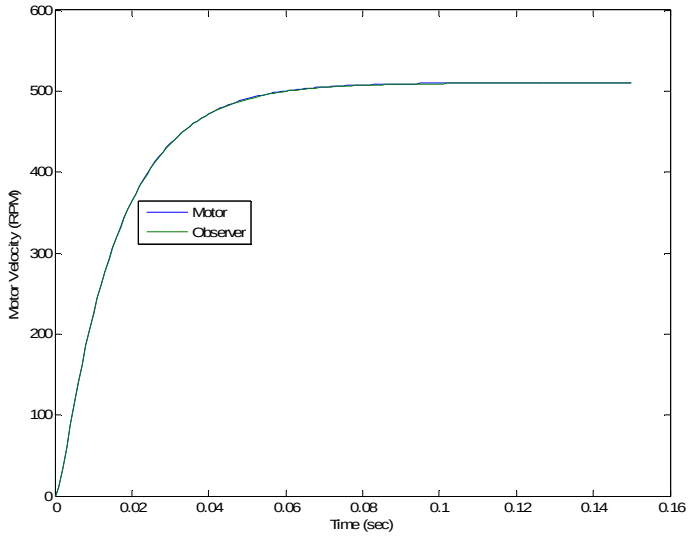
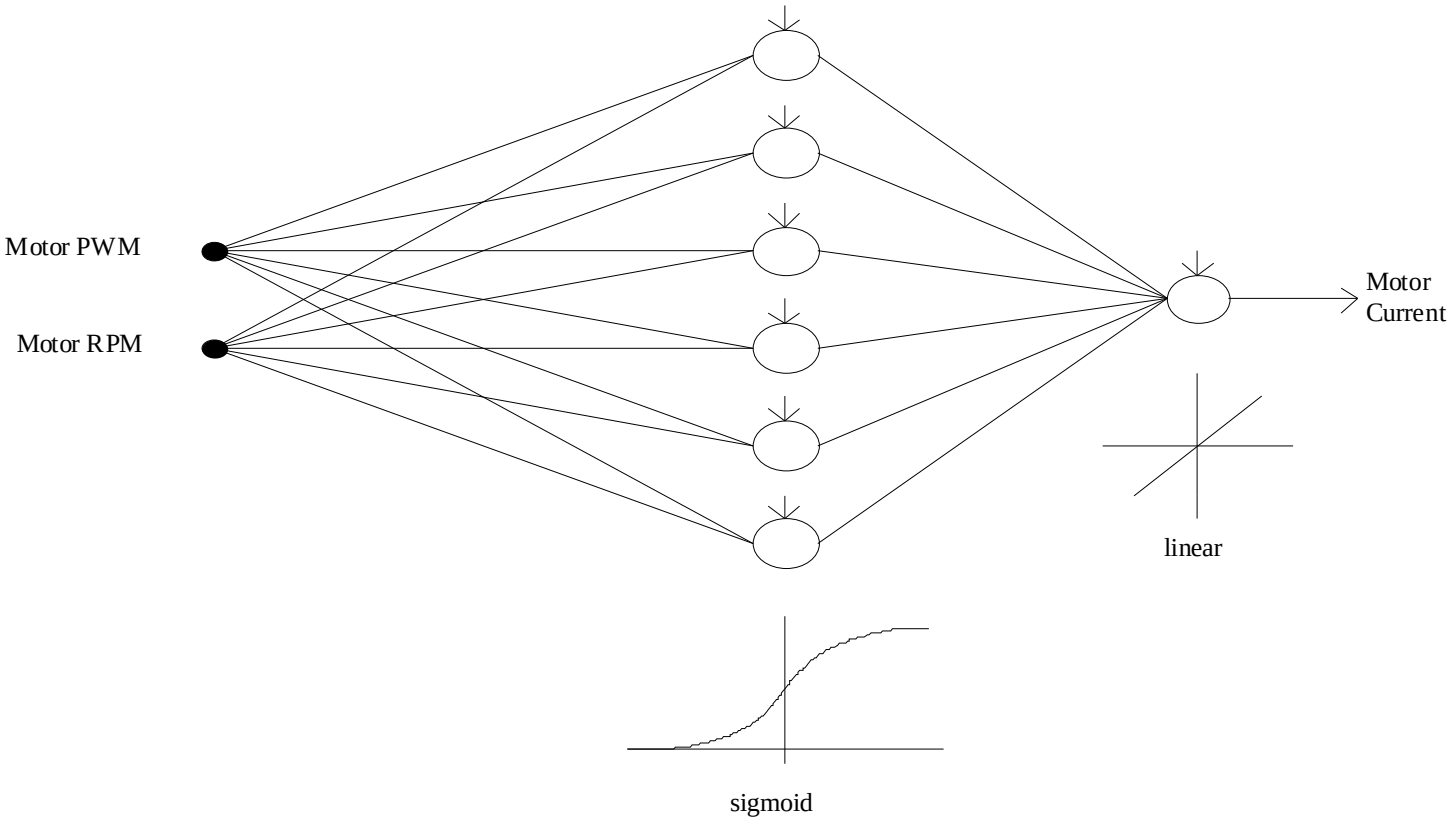


Figure 9.4 – 30.3V (834 RPM) Observer Response

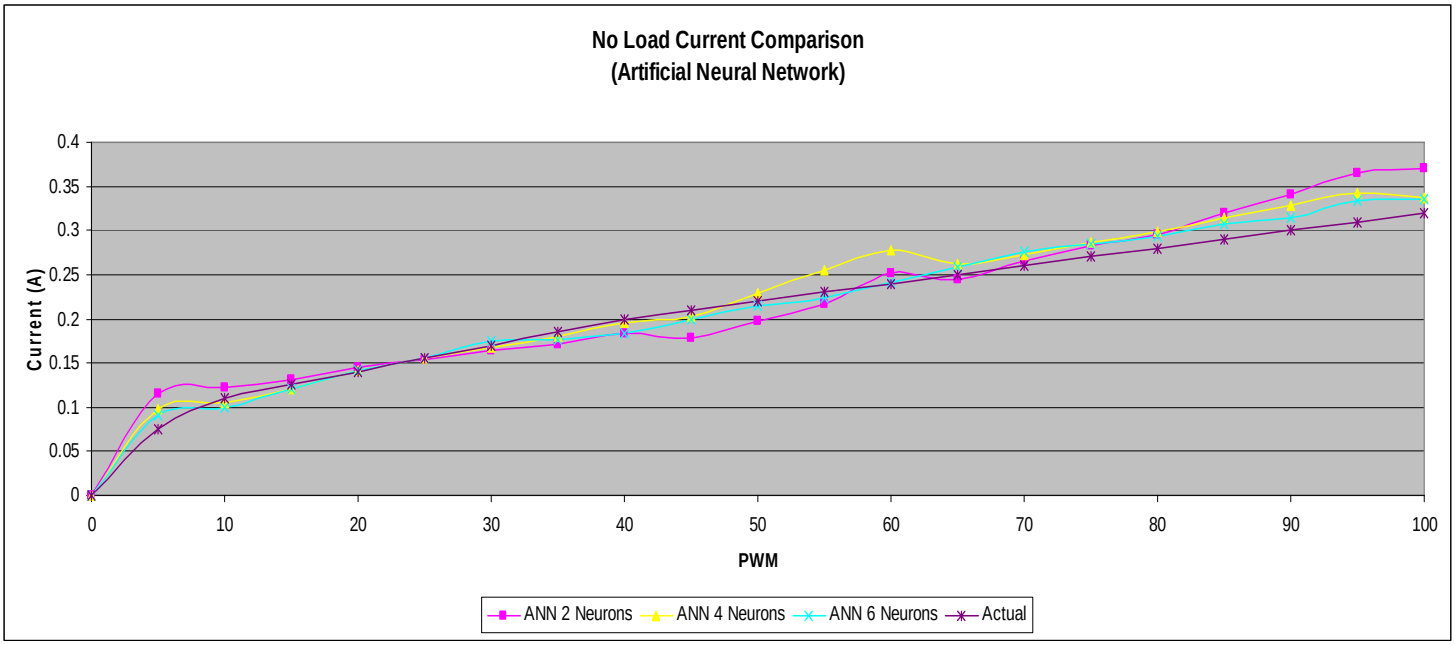
Artificial Neural Network

Although the Observer Design yielded acceptable results, an Artificial Neural Network (ANN) was created for comparison. The motor has a single input (PWM) and a single output (RPM). Since I want to estimate the Motor Current, I collected Motor Current data at different Motor PWMs (0-100) and Motor RPMs (0-834) for three different loads (0 amp load, 0.5 amp load, and a 1 amp load). The training data would be the data collected at every 10 PWM and the test data was the data collected every 5 PWM.

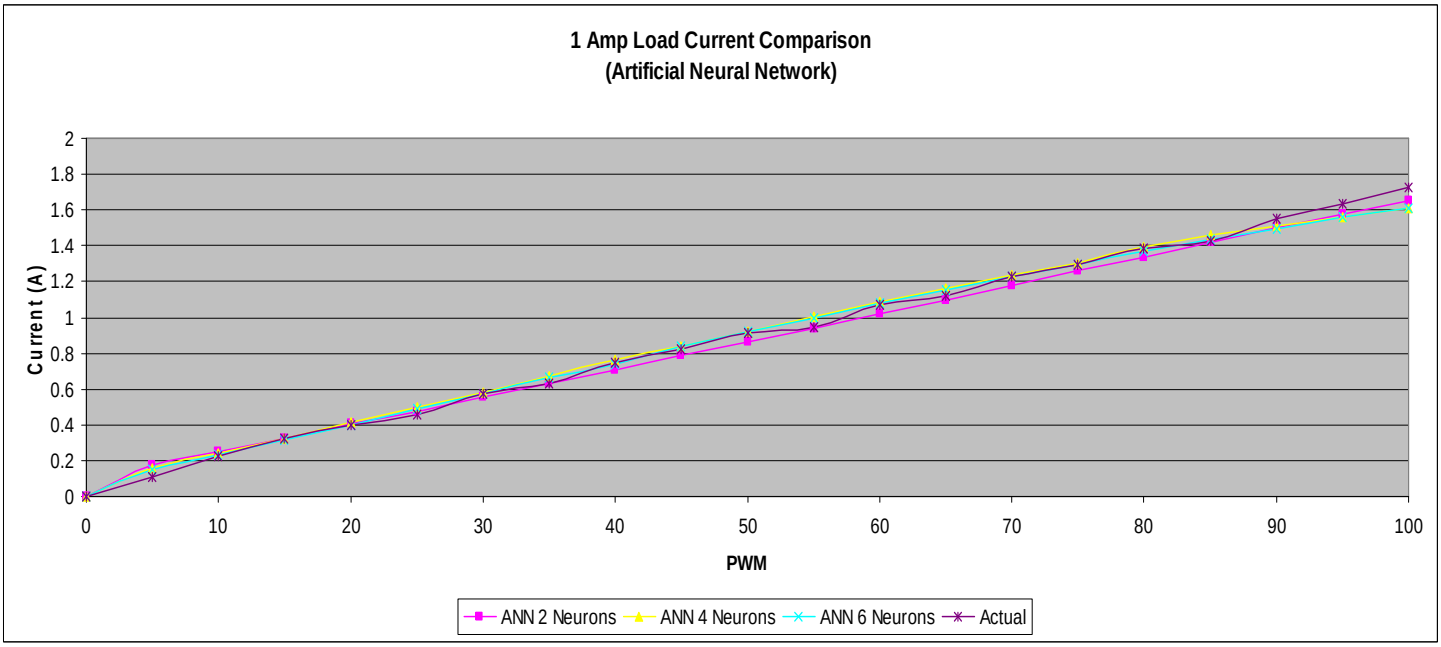
Initially the ANN was modeled as a single 'purelin' neuron (Adaline) but the model did not yield acceptable results. Hence, Multilayer Feedforward Back Propagation Trained networks were introduced with a hidden layer of 2, 4, and 6 neurons. The ANN performed a Mapping Process, where the output is mapped to the input values, which required a hidden layer of 6 'logsig' neurons and 1 'purelin' neuron (Fig 10.1). The 6 neuron ANN had the least error (Fig 11.1 & 11.2).



10.1 – Artificial Neural Network Design



11.1 – Artificial Neural Network No Load Comparison



11.2 – Artificial Neural Network 1 Amp Load Comparison

Energy Management Equations

The motor schematic (Fig 8.1) for the Pittman Motor was used to calculate the power losses from the armature, linear friction, Coulomb friction, and the load (Fig 12.1 to 12.5). Summation of these power losses produces an estimation for the total power losses in the motor. Total power loss combined with the ambient temperature from the CAN bus (Thermal controller) allows for the estimation of the winding temperature, motor efficiency, and continuous torque (Fig 12.6 to 12.10). Observations of the motor efficiency and winding temperature are shown in Fig 13.1 and 13.2 respectively.

$$P_{armature} = i_a^2 * Ra$$

Figure 12.1 – Armature Power

$$P_{Colfric} = w * T_{Coulomb}$$

Figure 12.2 – Coulomb Friction Power

$$P_{linefric} = w^2 * B$$

Figure 12.3 – Linear Friction Power

$$P_{load} = K_t * (i_a - i_{noload})$$

Figure 12.4 – Applied Load Power

$$P_{loss} = P_{armature} + P_{linefric} + P_{Colfric} + P_{load}$$

Figure 12.5 – Power Loss

$$P_{in} = i_a * va$$

Figure 12.6 – Power Input

$$P_{out} = P_{in} - P_{loss}$$

Figure 12.7 – Power Output

$$Efficiency_{Mot} = P_{out} / P_{in}$$

Figure 12.8 – Motor Efficiency

$$Temperature_{MotWind} = P_{loss} * R_{thermal} + Temp_{amb}$$

Figure 12.9 – Motor Winding Temperature

$$Tcont = \sqrt{\frac{Temp_{MotWind} - Temp_{amb}}{R_{th}} - \frac{T_f \cdot S}{C}} \cdot K \cdot K_m - T_f$$

Figure 12.10 – Motor Winding Temperature

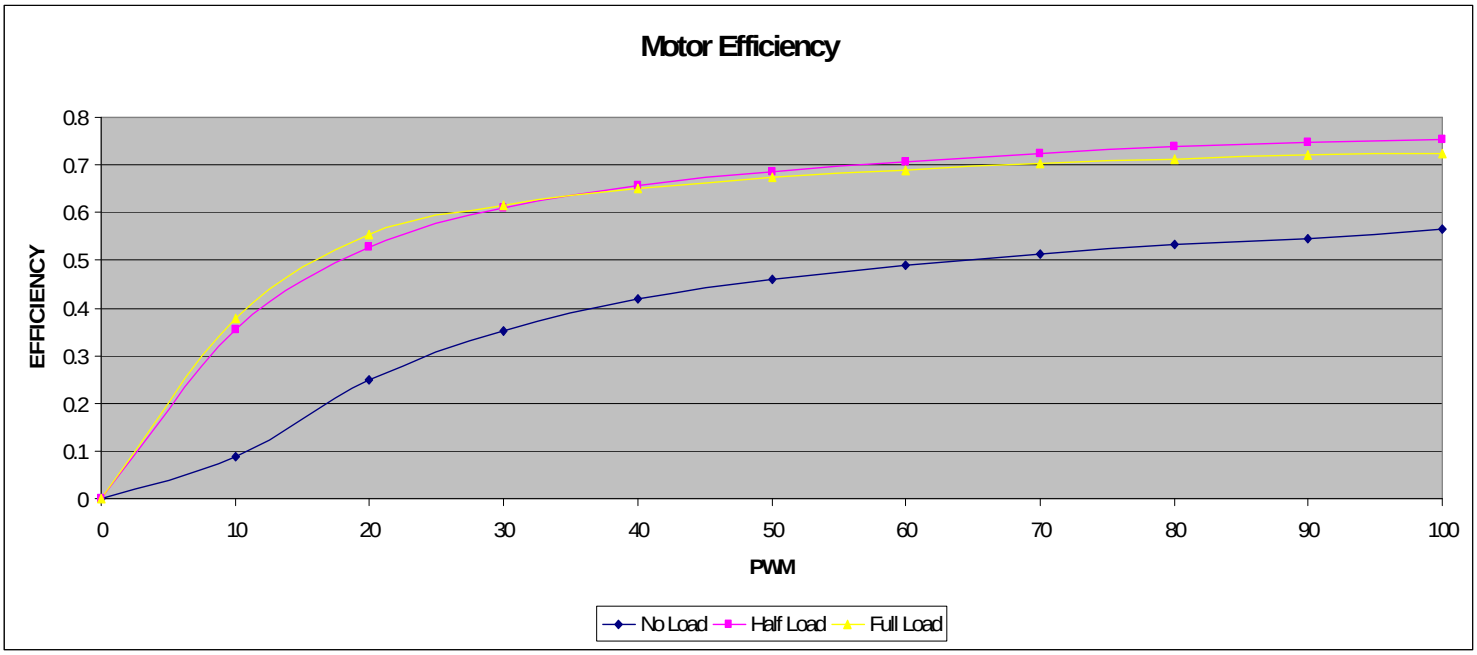


Figure 13.1 – Motor Efficiency

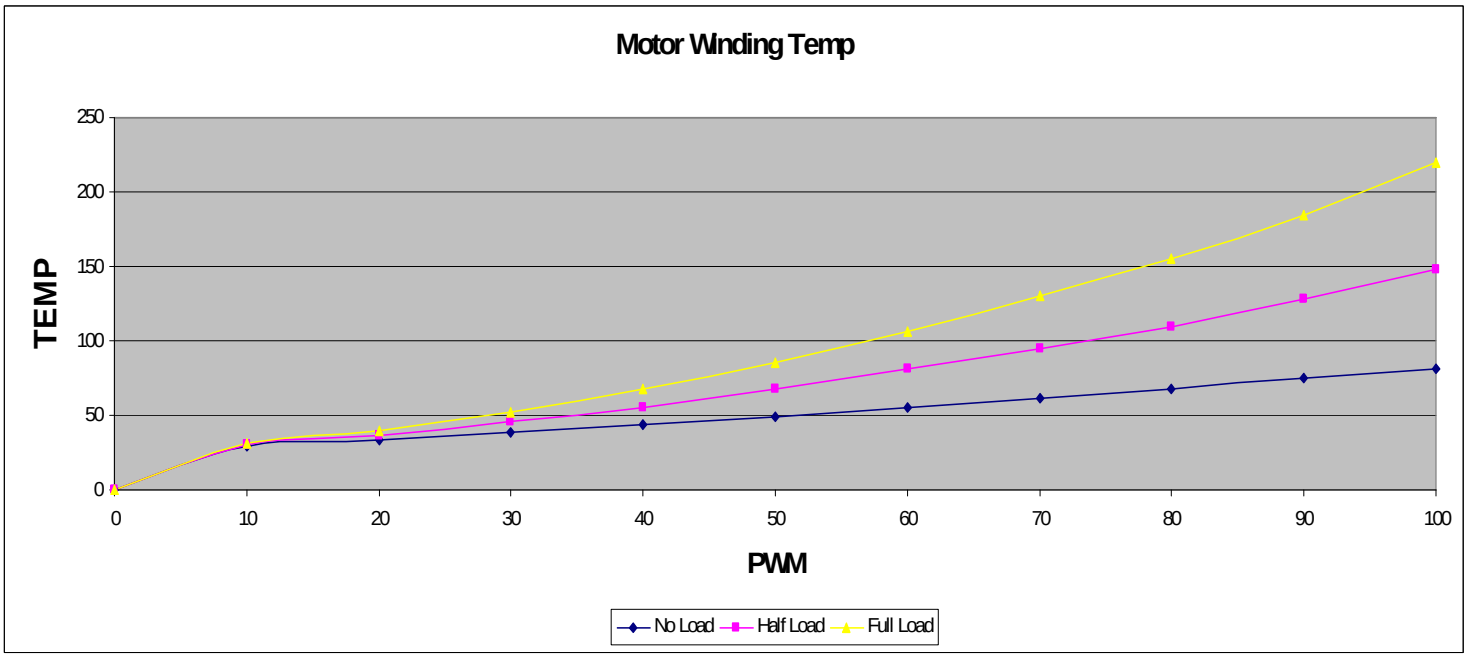


Figure 13.2 – Motor Winding Temperature

Analysis of Results

Controller Designs

Engine control was implemented with four control designs: Proportional (P), Integral (I), Lead Network (LN), and Feedforward (FF). Below I have summarized the results (Fig 14.1 & 14.2).

System (Best, Average, Worst)	P	PI	PI & LN	PI, LN, & FF	Goal
Steady state error	-318	0	0	0	±5 RPM
Ts	11ms	38ms	8ms	5ms	30ms
Tp	9.5ms	18.5ms	n/a	6ms	20ms
% overshoot	10%	19.1%	0%	10%	10%
Phase Margin	7.4°	55°	69.7°	69.7°	60°
Gain Margin	n/a	23.3dB	16.1dB	16.1dB	6dB
Maximum Velocity (RPM)	516	834	834	834	834

Figure 14.1 – Control Comparison

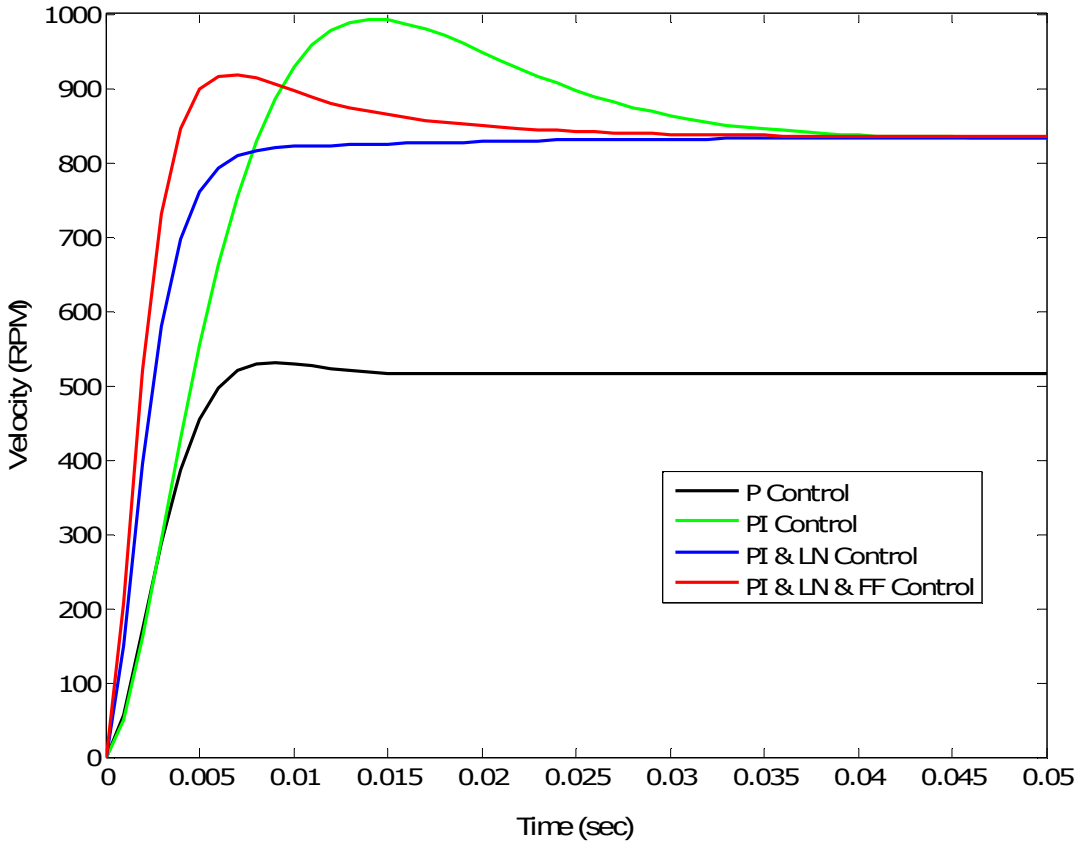


Figure 14.2 – 834 RPM Step Response of Control Designs

Observer Design vs. Artificial Neural Network

Estimation of motor current was implemented in two separate designs: Observer Design (OD) and Artificial Neural Network (ANN). Below I have summarized the results (Fig 15.1, 15.2, & 15.3).

Observer Design vs. Artificial Neural Network Trade-Offs				
	Memory (Bytes)	**Speed (MATLAB)**	**Overshoot (MATLAB)**	**Future-Proof**
Observer Design	13x4=52	98ms	700%	Converges to Motor
Artificial Neural Network (6 Neuron)	34x4=136	90ms	14.6%	Converges to Generalized Value

Figure 15.1 – No Load Current ANN vs. OD Comparison

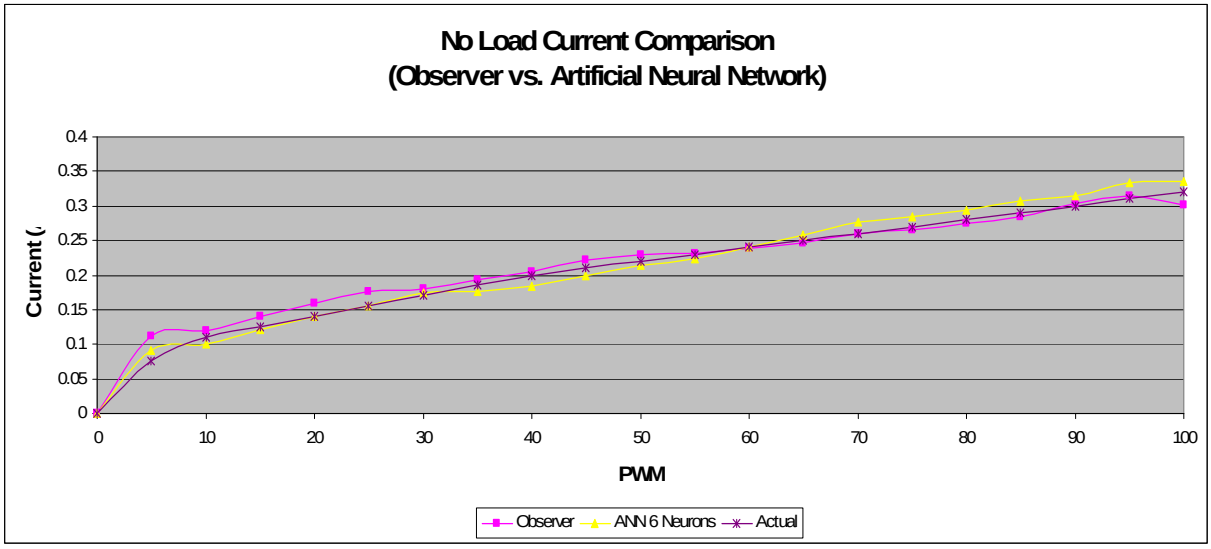


Figure 15.2 – No Load Current ANN vs. OD Comparison

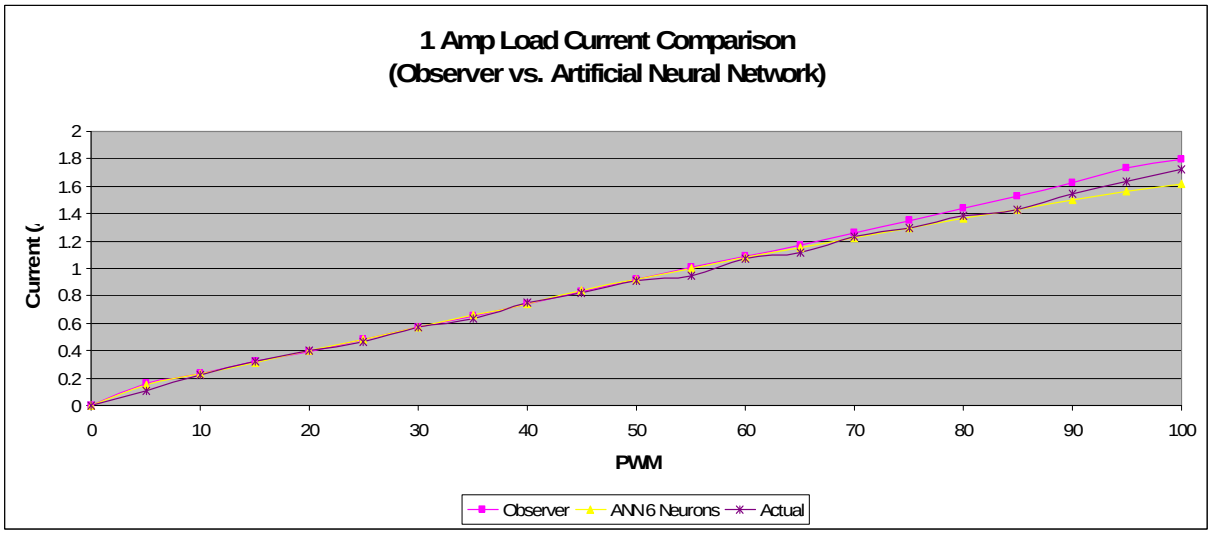


Figure 15.3 – 1 Amp Load Current ANN vs. OD Comparison

CAN Bus

Temperature data is exchanged via a CAN 2.0B bus interface. This interface does not use physical memory addresses to send information. Every message sent by the controller is instead tagged with a specific identifier. These identifiers, in a similar fashion to interrupts, have different priorities by which they are read by the other controllers.

Temperature and Torque Governor

Temperature data from the CAN bus is combined with the continuous torque calculation to limit the motor. Limitations were set according to the Pittman motor data sheet [3]. The temperature of the motor winding is limited to 155° C, since there is a time delay between the winding and the case, a safe motor operation temperature was set to 115° C. Continuous torque was arbitrarily limited to 0.01 N·m below its maximum rating of 0.067 N·m.

Conclusion

The Energy Management System for Electric Engines using Collaborative Controllers met its desired specifications. The CAN bus between the engine controller and thermal controller was crucial to managing both the engine and the cooling system. The exchanged temperature data allowed the engine controller to keep the motor below or at its rated limits while still applying maximum torque to the load. In this manner the motor was expending the appropriate amount of energy regardless of the applied load.

References

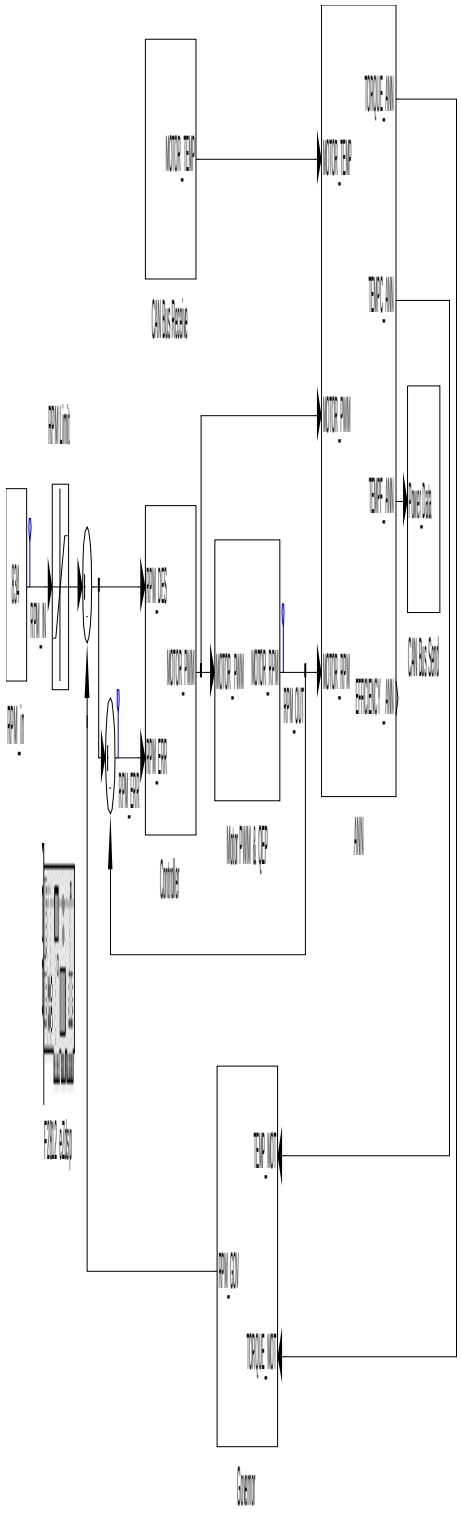
[1] Dr. Gary Dempsey. "Energy Management System for Senior Capstone Project 2009/2010," Bradley University, 2009.

[2] Mark Bright, Mike Donaldson, and Dr. Gary Dempsey. "Engine Control Workstation Using Simulink / DSP Platform: Functional Description and System Block Diagram," Bradley University, 2009.

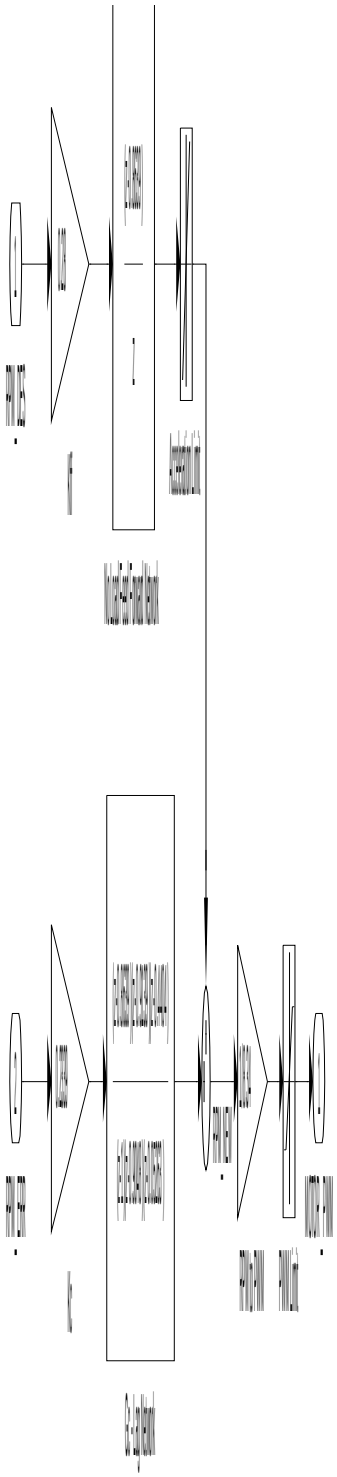
[3] Pittman. "LO-COG DC Motors," Pittman, 2009.

Appendix

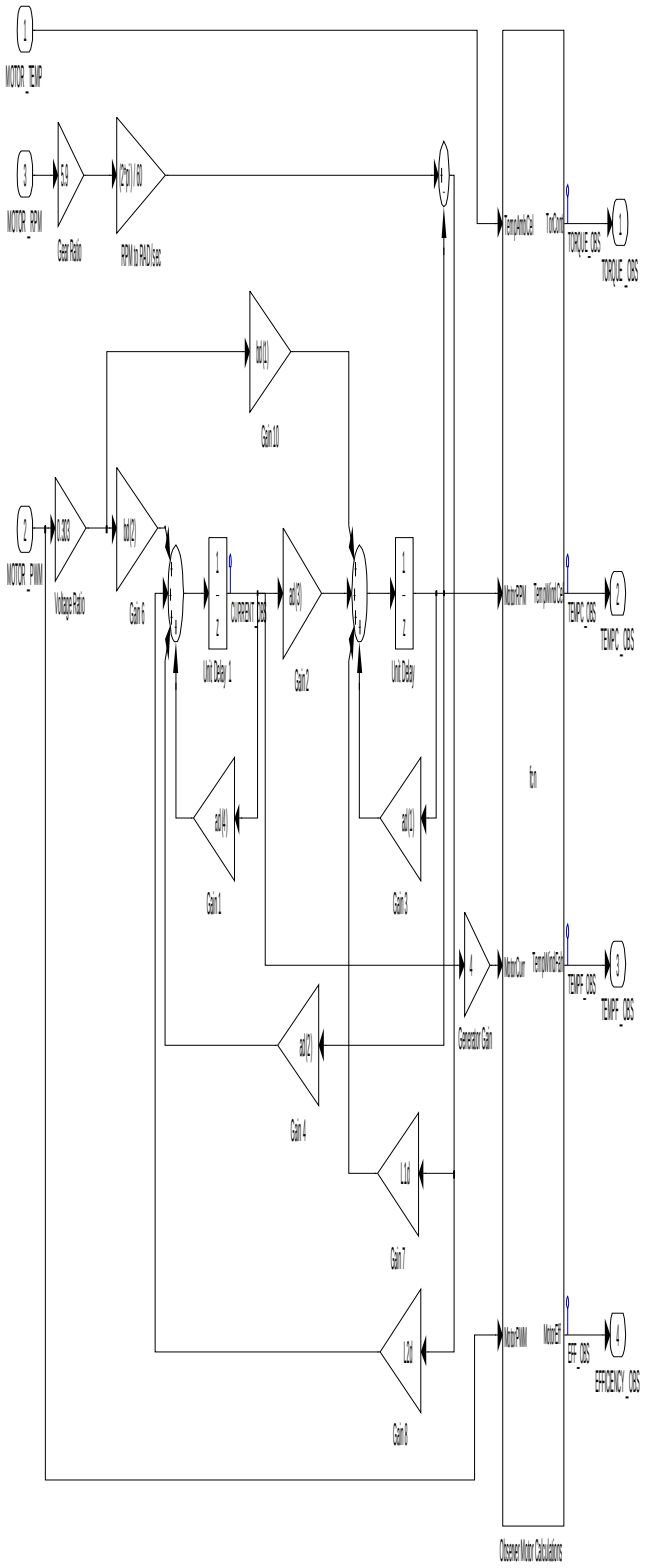
Overall System



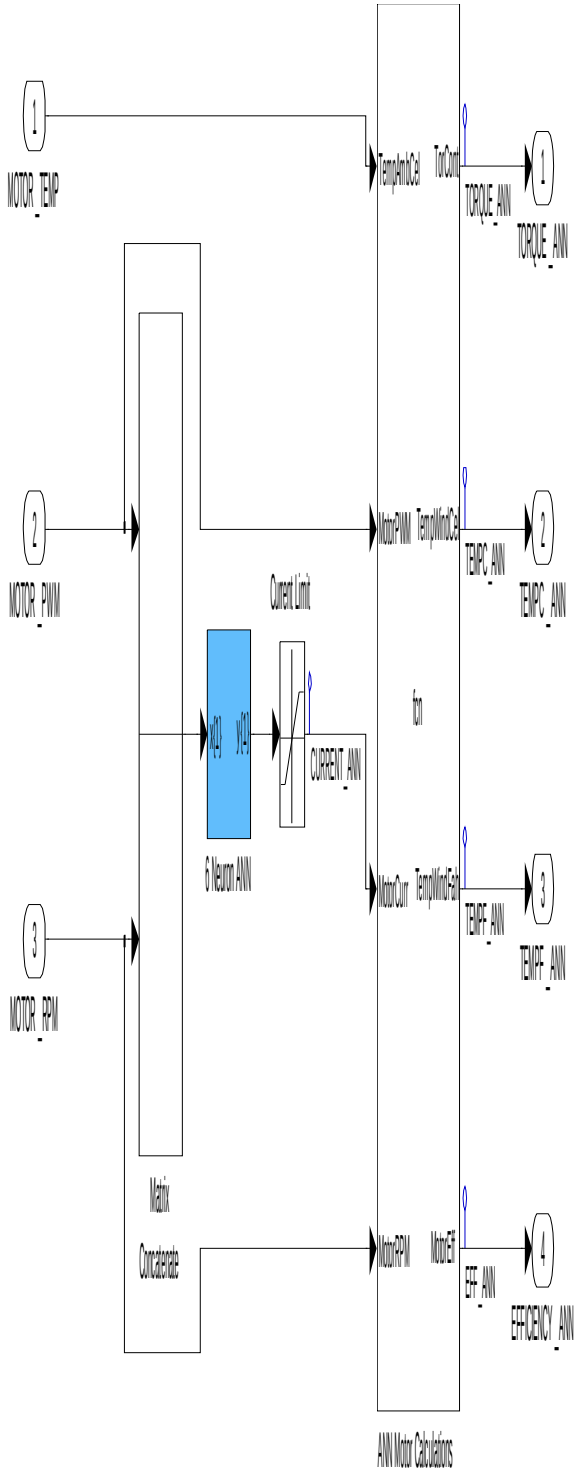
Controller



Observer Design



Artificial Neural Network



Governor