

**dSPACE DS1103 Control Workstation Tutorial and DC
Motor Speed Control**

Project Report

By

Annemarie Thomas

Advisor: Dr. Winfred Anakwa

May 12, 2009

Abstract

The dSPACE DS1103 software and hardware tools for embedded system development come with several volumes of documentation. Consequently, the user must spend a lot of time learning how to use the tools. The purpose of this project is to write a tutorial using DC motor speed control as the example application in order to decrease the time required by future students to learn how to use the workstation so they may complete more complex projects. This required a detailed study of the DS1103 hardware and software tools, design of a controller for the DC motor and simulation of the controller in Simulink. RealTime Workshop and ControlDesk were used to generate C code for the digital controller equation automatically and download it to the DS1103 board for execution. ControlDesk software was then used to monitor and control a variety of variables.

Table of Contents

Introduction	1
Project Summary	1
Functional Description	1
<i>ControlDesk/Speed Input</i>	2
<i>CLP1103</i>	2
<i>Motor Subsystem</i>	2
<i>Optical Encoder Subsystem</i>	3
Requirements	3
<i>Overall System Requirements</i>	3
<i>Controller Requirements</i>	4
<i>Other Requirements</i>	4
Equipment	5
<i>Workstation</i>	5
<i>Other Equipment</i>	5
Project Implementation	6
<i>Plant Model</i>	6
<i>Controller Design</i>	7
<i>MATLAB Calculations</i>	8
<i>Simulink Simulation</i>	10
<i>Actual System</i>	13
Conclusions	18
References	19
Appendix A: MATLAB Code and Results	A1
<i>MATLAB Code</i>	A1
<i>MATLAB Results</i>	A3
Appendix B: Project Completion Timeline	B1

Appendix C: Datasheet Excerpts.....	C1
<i>Transistor.....</i>	<i>C1</i>
<i>Inverter.....</i>	<i>C2</i>
<i>Encoder.....</i>	<i>C6</i>

Introduction

The dSPACE DS1102 was first used at Bradley University in the year 2000 when a user's manual and a workstation based on this board were developed. Since then, a newer dSPACE DS1103 board has become available. With grant from the Bradley College of Engineering of \$2,000 and additional funds of \$12,200 provided by Dr. Larry Kendrick and Mr. Shawn Weck at the Caterpillar Technical Center, a dSPACE DS1103 board and additional equipment required have been purchased to set up a DS1103 workstation. Initial setup has been completed by Mr. Chris Mattus. The object of the *dSPACE DS1103 Control Workstation Tutorial and DC Motor Speed Control* project is to develop a tutorial for use of the new dSPACE DS1103 Workstation by implementing a DC motor speed control system. The tutorial will minimize the time required for future students to become proficient in using the workstation, therefore increasing the time they can spend designing and implementing more complex control systems.

Project Summary

The goals of the *dSPACE DS1103 Control Workstation Tutorial and DC Motor Speed Control* project are the following:

- Determine and implement any additional setup that may be required for the dSPACE DS1103 control workstation.
- Write a tutorial for use of the workstation, paying particular attention to details related to the DS1103 board using a DC motor speed controller as the example application.
- Design a controller to control the speed of a DC motor.
- Implement the designed DC motor speed controller on the dSPACE DS1103 system using the Simulink and the dSPACE blocksets, the Matlab-to-DSP interface libraries, Real-Time Interface to Simulink, and Real-Time Workshop.

Functional Description

The design portion of this project involves designing and implementing a motor speed controller using the dSPACE DS1103 board. The controller was designed and simulated using the Simulink and the dSPACE blocksets, the Matlab-to-DSP interface libraries, Real-Time Interface to Simulink, and Real-Time Workshop on a PC. The controller was then downloaded onto the Texas Instruments' TM320F240 DSP [1] located on the DS1103 board. Inputs to the DS1103 include the desired motor speed and the output of the optical encoder connected to the motor. Outputs from the DS1103 include the PWM signal input to the motor and the display of the actual motor speed. Inputs to and outputs from the DS1103 pass through the CLP1103 [2] connector panel for the DS1103 board or through the cables connected to the computer. The motor's speed remains reasonably steady until a certain amount of load torque is applied to it by a brake. A general block diagram for the system is shown in figure 1 on the next page.

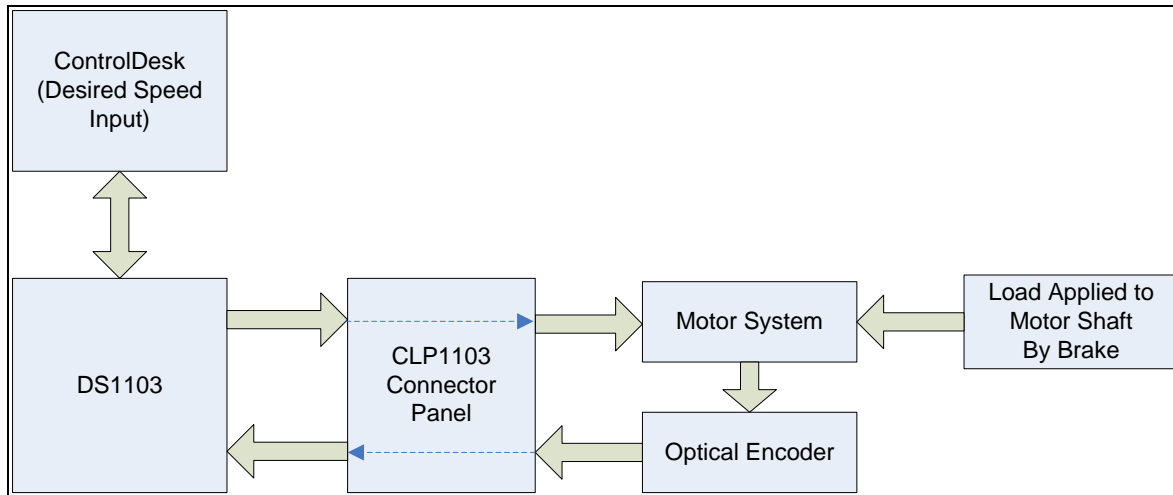


Figure 1: Motor Speed Control System Block Diagram

ControlDesk/Speed Input

“ControlDesk” in figure 1 serves multiple uses. It provides the interface for downloading controller models designed in Simulink onto the DSP. The Instrument Panel feature of ControlDesk is used to display various measurements such as the duty cycle of the PWM signal entering the motor and the actual velocity in RPM that the motor is running at and is used to control inputs such as the “Desired Speed Input” to the controller. Some inputs, such as the signal from the “Optical Encoder Subsystem”, must be sent through the “CLP1103” and “DS1103” before the measurements can be displayed in ControlDesk on the computer [3]. Initially, measurement of the system response (the actual motor speed) was completed by connecting the output of the Optical Encoder to an oscilloscope in order to verify the behavior of the outputs of the encoder blocks provided with the DS1103 software for Simulink and the results observed the ControlDesk layout.

CLP1103

The “CLP1103” shown in figure 1 serves as an interface between the “DS1103” and the external hardware portion (“Motor System”, “Optical Encoder”) of the overall system. The CLP1103 contains connectors for twenty (20) Analog-to-Digital inputs, eight (8) Digital-to-Analog outputs, several other connectors that can be used for Digital I/O, Slave/DSP I/O, Incremental Encoder Interfaces, CAN interface, and Serial Interfaces [2]. This project uses the Slave I/O Connector to output the PWM signal from the DS1103 to the Motor Subsystem and an Incremental Encoder connector to pass the Optical Encoder output to the DS1103.

Motor Subsystem

The “Motor System” identified in the block diagram in figure 1 includes the motor and additional analog components. Further information about the Motor Subsystem can be found in the Equipment section under Other Equipment section.

Optical Encoder Subsystem

The “Optical Encoder System” shown in figure 1 simply consists on the optical encoder connected to the motor and the necessary (2.7 k Ω) pull-up resistors (connected between Channel A and 5 Volts and Channel B and 5 Volts) and power/ground connections with no additional analog components being needed to convert the optical encoder output signal into a form that can be more easily handled by the DSP.

Requirements

The requirements for this project are currently as follows:

Overall System Requirements

Requirements and specifications for the overall system shown in figure 1 on page 2 are as follows:

- The operating range of the system shall be 0 to 40 degrees C.
- The system shall use a 30 VDC or smaller power supply. (Uses 24 VDC Supply.)
- The motor used shall be a Pittman DC Motor Model Part # GM9236C534-R2.
- The values for the motor model to be used for design and simulation in Simulink shall be determined from the datasheets and verified experimentally.
- A control panel shall be created in the “ControlDesk” application to monitor the inputs and the outputs of the system.
- The output from the “DS1103” to the “Motor System” shall be passed through a PWM output located on the “CLP1103 Connector Panel”.
- The output from the “DS1103” to the “Motor System” shall be a 12 kHz PWM signal.
- The input to the “DS1103” from the “Optical Encoder System” shall be passed through an Incremental Encoder input located on the “CLP1103 Connector Panel”.
- The input to the “DS1103” from the “Optical Encoder System” shall be converted to RPM for display purposes.
- The RPM measurement displayed using a control panel in “ControlDesk” shall be within ± 10 RPM of the actual value. (This requirement has not been met. It would require an encoder with higher resolution.)
- There shall be a method to input/control the desired speed of the motor.

- The desired maximum speed/velocity that shall be input is 660.594 RPM.
- Load Torque shall be applied to the motor shaft using a HB-420 Brake.
Note: The maximum load torque that can be applied has been limited to reasonable values.
- The designed controller shall be downloaded onto the Texas Instruments' TM320F240 DSP [1] located on the "DS1103" board using "ControlDesk".

Controller Requirements

Requirements and specification applicable to the controller functionality are as follows:

- The DC motor speed controller shall be designed/simulated using Simulink and dSPACE blocksets, Matlab-to-DSP interface libraries, the Real-Time Interface to Simulink, and Real-Time Workshop.
- Overshoot shall be less than or equal to 5%.
- Rise time shall be less than or equal to 110 ms.
- Steady-state error shall be minimized.

Other Requirements

Other requirements particular to this project are as follows:

- A tutorial shall be written on the use of the workstation using the DC motor speed controller as the example application.

Equipment

Equipment used in this project is as follows:

Workstation

The dSPACE DS1103 control workstation contains the following:

- The \$14,000 dSPACE DS1103 system consisting of:
 - A DS1103 Board.
 - An Expansion Box to house the DS1103.
 - A CLP1103 PPC Connector and LED Panel.
 - ControlDesk Version 3.2.2 and several other software applications provided by dSPACE.
- A PC dedicated to the workstation containing all software applications required for controller design (Matlab/Simulink Version R2008a and libraries) and implementation in the DS1103 (ControlDesk).

Other Equipment

Equipment particular to this project includes:

- A Pittman GM9236C534-R2 DC Motor.
- A Magtrol HB-420 Brake.
- A TIP120 Transistor.
- A 1N4004 Diode.
- SN7407 Hex Inverters.
- Other electronic components, power supplies, and measurement devices.

A circuit for the Motor Subsystem is shown in figure 2 below.

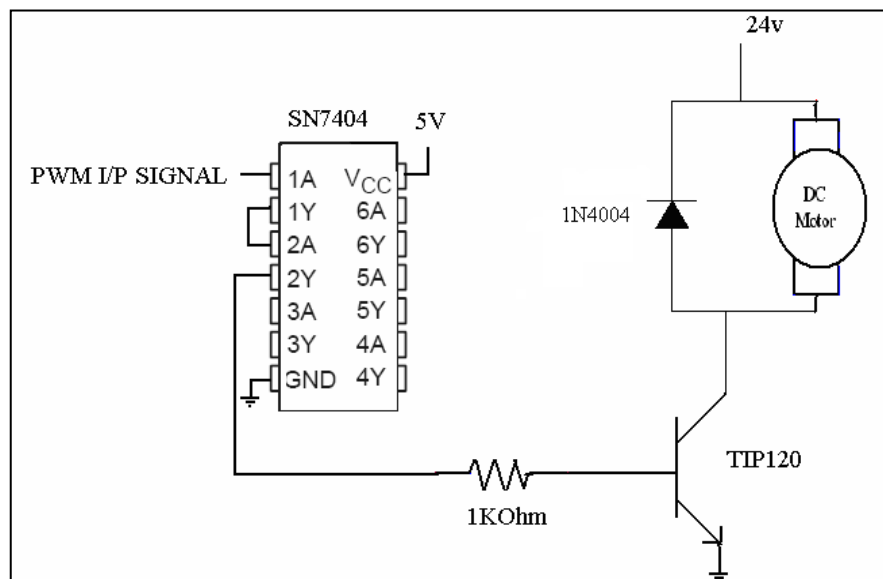


Figure 2: Motor Subsystem

Project Implementation

Plant Model

A simple motor model is shown in figure 3 below. From this model, an open loop control block diagram can be generated with a transfer function for the input voltage to the motor, V_a , in relation to the output, w , in radians per second, which can then be easily converted to other formats such as RPM (revolutions per minute). A simple transfer function that can be used to model the motor in the frequency domain, in terms of s , is:

$$\frac{w}{V_a} \equiv \frac{K_t}{(R_a + s * L_a)(B + s * J) + K_t * K_v} \quad (1)$$

where R_a and L_a are the electrical resistance and inductance of the motor, J and B are the mechanical inertia and damping/friction of the motor, and K_t and K_v are the torque constant and back emf constant of the motor.

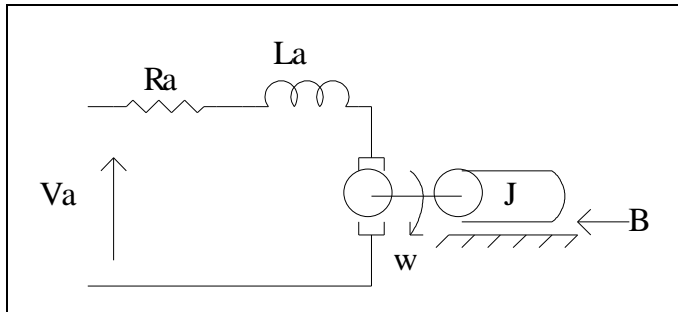


Figure 3: Simple Motor Model Schematic

One of the requirements states, “The values for the motor model to be used for design and simulation in Simulink shall be determined from the datasheets and verified experimentally.” Unfortunately, there was not enough time to verify the motor model provided through experimental measurements. The motor model used was generated in a previous project completed by Amulya Sabbisetti [6]. Though not perfect, it was found that when comparing the step response of the Simulink model for the entire system with no “BrakeTorque” applied to the step response of the actual system as recorded using ControlDesk that the rise times of the actual system and the simulation were fairly similar. These results are discussed in more depth in later sections. The plant model used is shown in figure 4 below, and a plot of its frequency response is shown in figure 5 on the next page showing its original phase margin of 84° at an ω_c of 279 radians per second. The analog transfer function for this model is found in the equation that follows:

$$G_P = \frac{502852}{(s + 14.54)s + 1776.78} \equiv \frac{19.4699}{\left(\frac{s}{14.54} + 1\right)\left(\frac{s}{1776.78} + 1\right)} \quad (2)$$

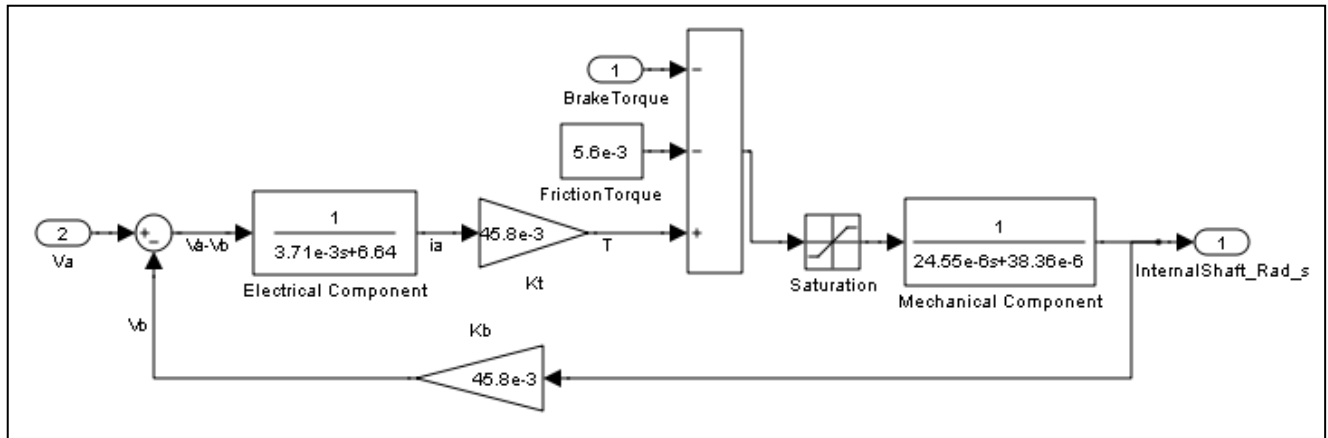


Figure 4: Motor Model

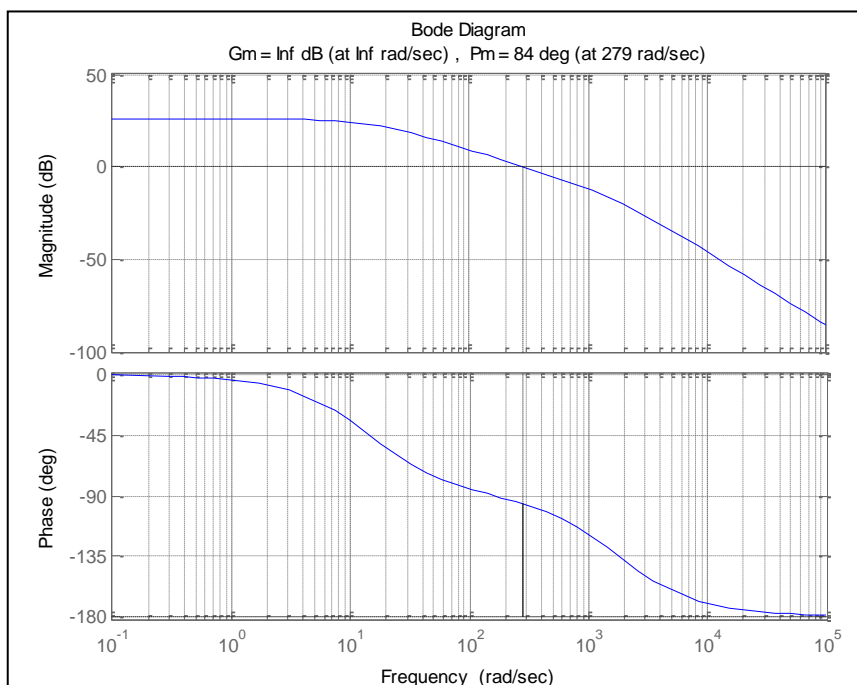


Figure 5: Frequency Response of the Motor Model.

Controller Design

The original rise time requirement was 20 ms. The controller was designed in the analog domain using Bode method for this rise time and then converted to the digital domain using MATLAB. First an integrator ($G_c = 1/s$) was added to the controller to minimize the steady state error in the system. It was determined that, for a rise time of 20 ms and overshoot of approximately five percent (5%), an ω_c of 157.08 radians per second with a phase margin of about 69.01° should be the design specifications. This leads to the following calculations:

$$\beta_c = -180^\circ - 69.01^\circ = -110.99^\circ = -1.937 \text{ radians} \quad (3)$$

$$\beta_c = -\frac{\pi}{2}(2) + \left(\frac{\pi}{2} - \frac{\omega_1}{\omega_c}\right) - \frac{\omega_c}{\omega_2} \quad (4)$$

$$\omega_c = \sqrt{\omega_1 \omega_2} \rightarrow \omega_1 = \frac{\omega_c^2}{\omega_2} \quad (5)$$

Substituting in a β_c of -1.937 radians, $\omega_c = 157.08$ radians per second, and equation 5 into equation 4 leads to $\omega_2 = 857.579$ radians per second to be used as a pole of \mathbf{G}_c . Substituting this ω_2 into equation 5 leads to $\omega_1 = 28.7718$ radians per second to be used as a zero of \mathbf{G}_c . Adding this pole and zero to \mathbf{G}_c provides an analog controller of:

$$\mathbf{G}_c = \frac{s + 28.7718}{s(s + 857.579)} \quad (6)$$

Adding gain and converting the analog controller using pre-warping and bilinear transformation for a sampling frequency of 12 kHz (the required PWM frequency) generates a digital controller as follows:

$$\begin{aligned} \mathbf{G}_{\text{Cdigital}} &= \\ &= 72.0631(0.0482282) \frac{(z + 1)(z - 0.9976)}{(z - 1)(z - 0.931)} \\ &= 72.0631(0.0482282) \frac{z^2 + 0.0024z - 0.9976}{z^2 - 1.931z + 0.931} \end{aligned} \quad (7)$$

This equation was then implemented using MATLAB calculations and a Simulink simulation and the gain was then adjusted to obtain the desired results.

MATLAB Calculations

The code and results for the MATLAB calculations and plots that were made are shown in “Appendix A”. The gain of the system was adjusted from equation 7 above to adjust ω_c to be about 157 radians per second with a phase margin of about 69.5°. See figure 6 on the next page. After the plant was converted from the analog to the digital domain using the ‘zoh’ transformation in MATLAB, the digital controller was added, and the optical encoder gain was added. The step response simulated in MATLAB is shown in figure 7 on the next page. The step response shows about six percent (6%) overshoot and a rise time of approximately 8.33 ms ([125-25samples]*[1/12000] seconds per sample).

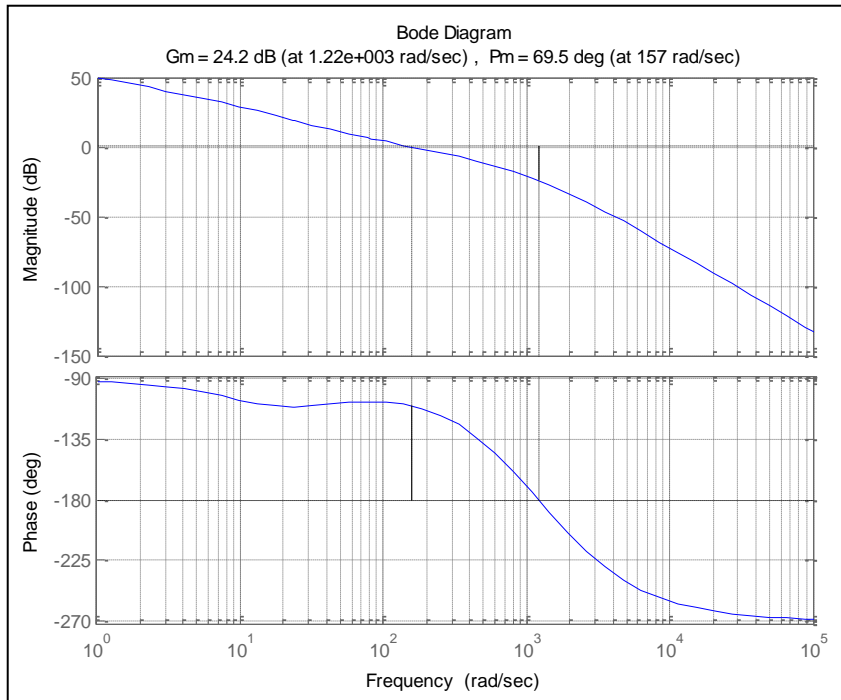


Figure 6: Frequency Response for combined digital controller and digital plant plotted in MATLAB.

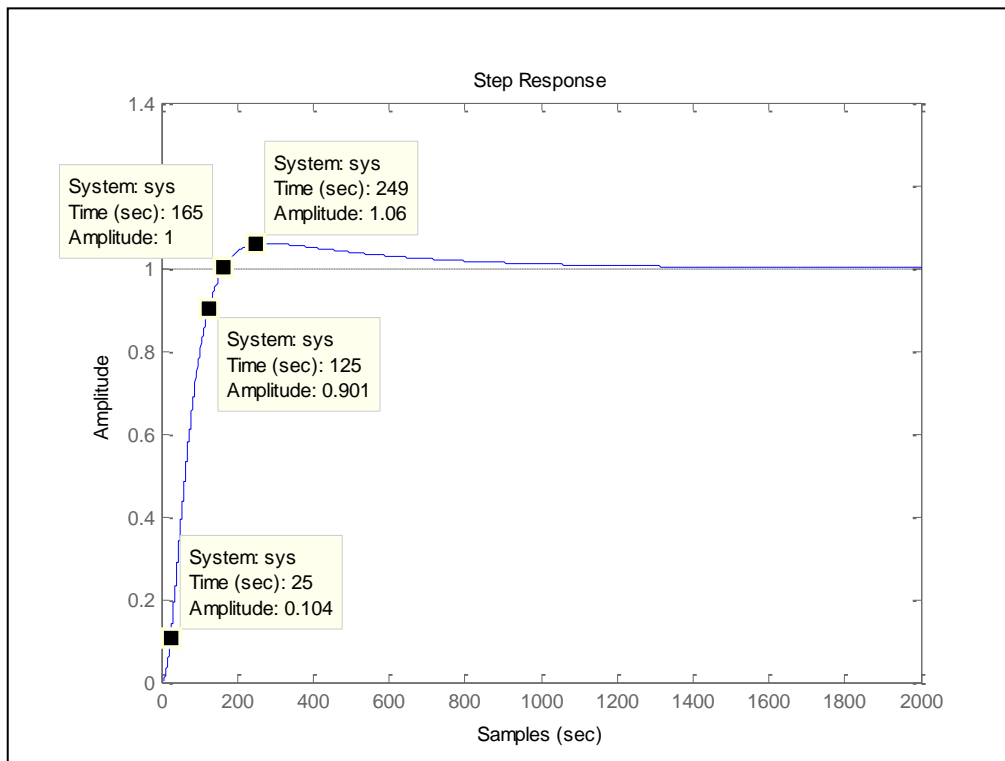


Figure 7: Step response generated by MATLAB code.

Simulink Simulation

The controller designed in MATLAB was then added to the Simulink simulation model and the closed loop system exhibited no overshoot and a significantly longer rise time. Similar rise time results were found when data was recorded with the controller downloaded to the DS1103 but overshoot or lack thereof was hard to detect using the data recorded by ControlDesk. Additional gain was added to the Simulink simulation, which was then also added to the controller downloaded to the DS1103, to obtain about five percent (5%) overshoot and a rise time of about 110 ms. The blocks used in the Simulink simulation are shown in figures 8-13 that follow. The Controller subsystem is shown in figure 10 on the next page. The “Hz_DutyCycle” gain block in figure 8 below is needed to convert the output of the “Controller”, which is a frequency value, to a duty cycle value that can then be converted to the voltage input for the “Motor” subsystem using the “PWM_to_Volts” subsystem, which models the effects of the hardware between the PWM output and the motor in the actual system.

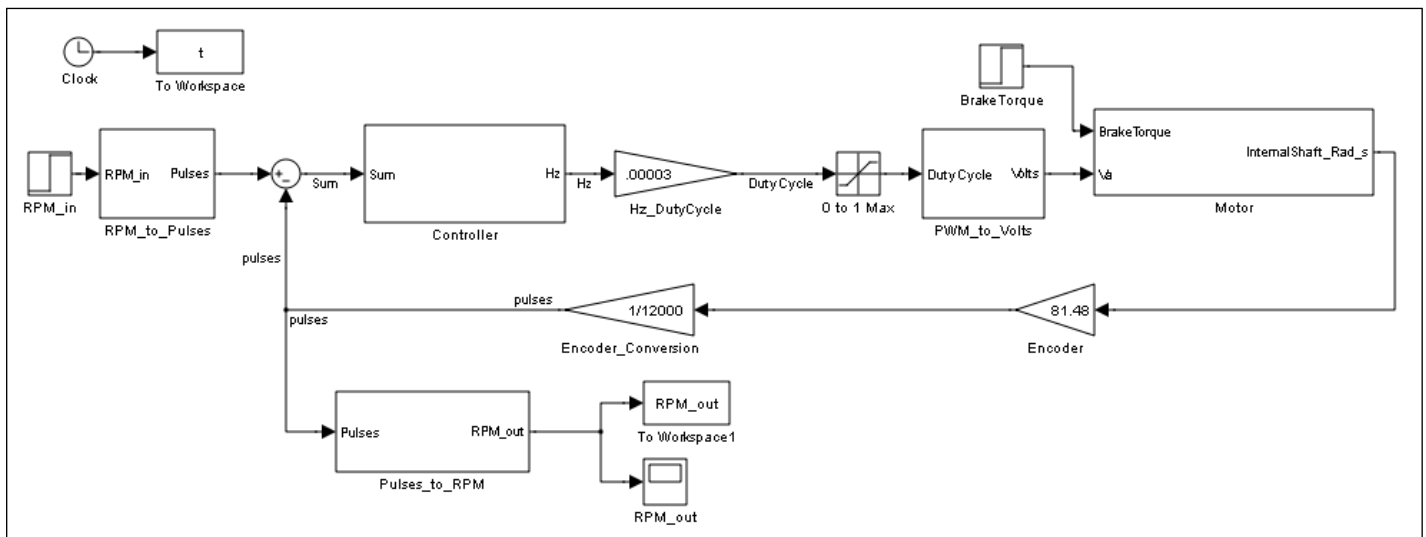


Figure 8: Simulink Model used to verify controller functionality.

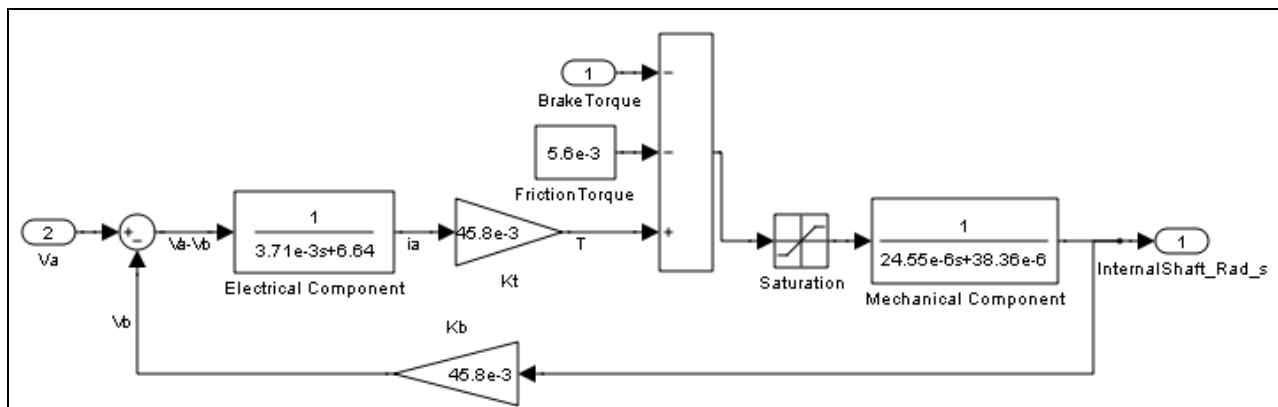


Figure 9: Motor Subsystem for Simulink model in figure 8 above.

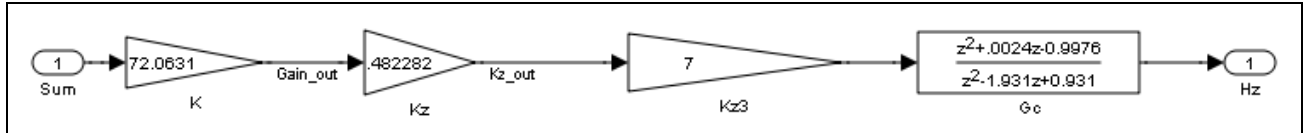


Figure 10: Controller Subsystem for Simulink model in figure 8 one the previous page.

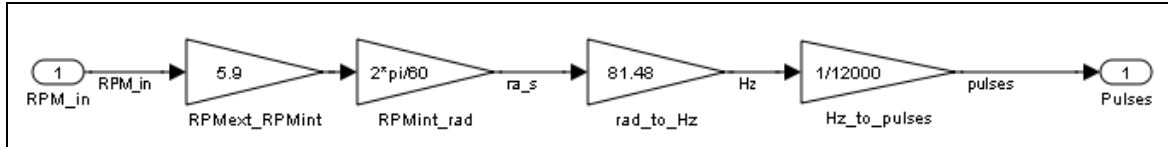


Figure 11: RPM_to_Pulses Subsystem for Simulink model in figure 8 on the previous page.

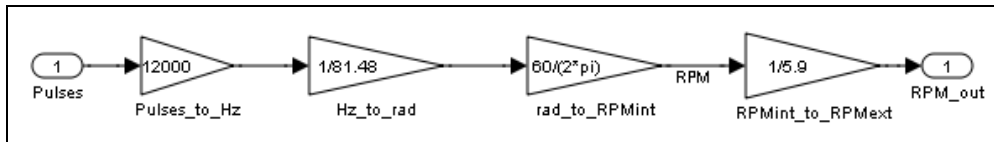


Figure 12: Pulses_to_RPM Subsystem for Simulink model in figure 8 on the previous page.

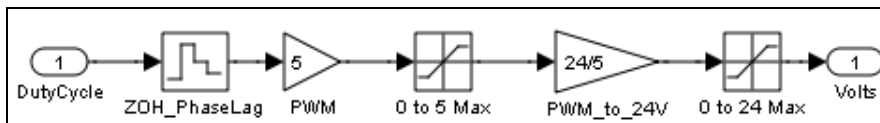


Figure 13: PWM_to_Volts Subsystem used to model the effects of the PWM and hardware connected to the motor for Simulink model in figure 8 on the previous page.

The approximate step response for the Simulink simulation was obtained for an RPM_in of 238.57 revolutions per minute in order to compare it to the MATLAB results which corresponds to the desired Pulses into the summer block from both the input and the bottom portion of the controller loop. RPM_in of 239.57 revolutions per minute is approximately equal to one Pulse.

The step response for the simulation is shown in figure 14 below with measurements shown to obtain five percent (5%) overshoot and approximately 110 ms rise time calculations:

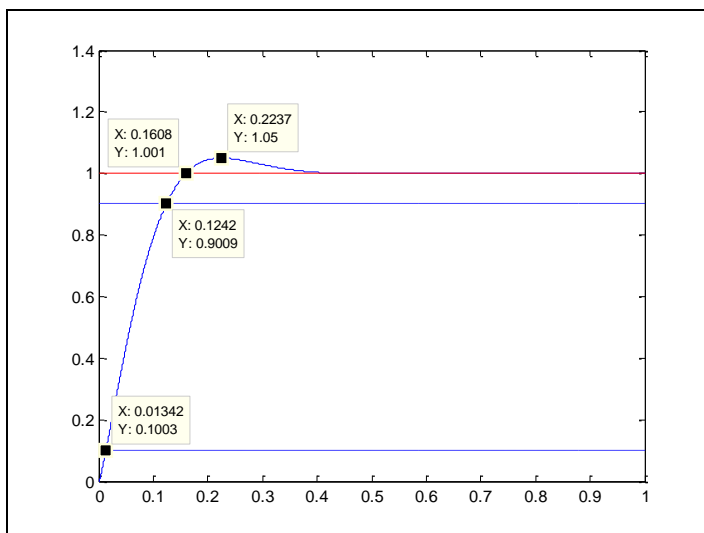


Figure 14: Step response for Simulink simulation.

The possible effects of the “BrakeTorque” input shown in figure 8 on page 11 were also observed for RPM_in of 200 revolutions per minute. The results are shown in figures 15-17 that follow. For lower “BrakeTorque”, there is a level at which RPM_out does not go directly to 200 revolutions per minute, but spikes at a lower value before dropping and then rising up to the desired level. Eventually, for higher levels of “BrakeTorque”, the desired speed is not reached until enough torque is applied and the output only spikes and the motor stops completely. The torque values do not correspond to those that affect the actual system and the initial spikes almost never if ever seen. The transitions in the actual system are generally fairly smooth.

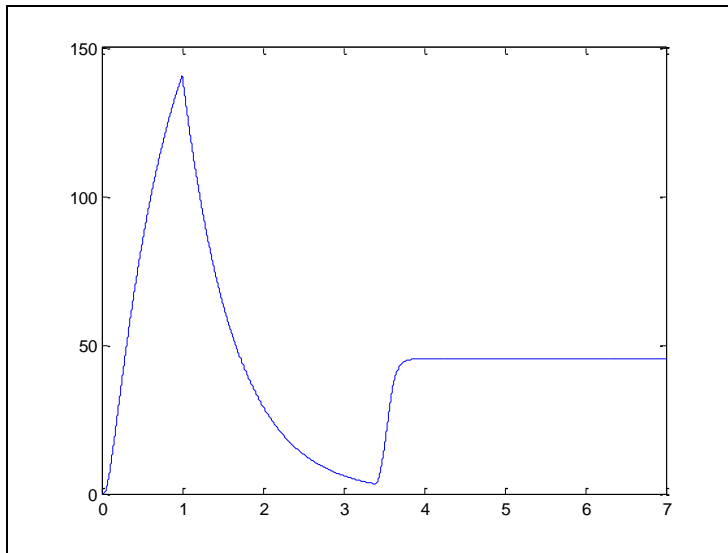


Figure 15: RPM_out when RPM_in = 200 RPM and BrakeTorque = 0.15N-m for the Simulink Model shown in Figure 4

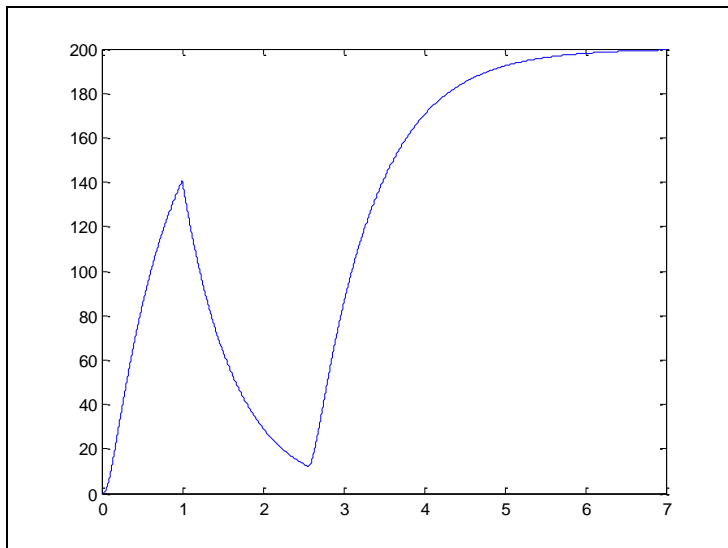


Figure 16: RPM_out when RPM_in = 200 RPM and BrakeTorque = 0.1N-m for the Simulink Model shown in Figure 8

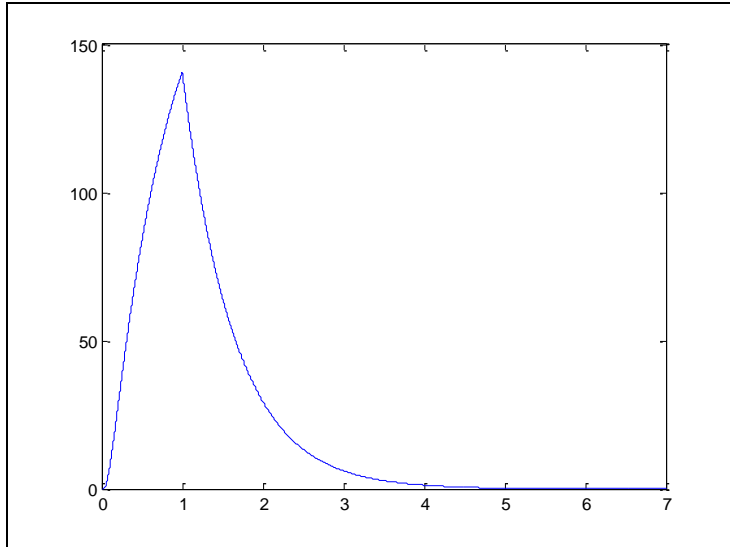


Figure 17: RPM_out when RPM_in = 200 RPM and BrakeTorque = 0.2N-m for the Simulink Model shown in Figure 8

Actual System

The Simulink model downloaded to the DS1003 board is very similar to the Simulink simulation in the previous section, but there are some differences. Of particular importance are the dSPACE provided blocks for outputting the PWM to the Motor System and inputting the encoder output to the model through the CLP1103 Connector Panel. For more information about how the actual system was implemented, see the “Motor Speed Control Application” section of the tutorial written as a part of this project [4]. The Simulink model downloaded to the DS1103 board is shown in figures 18-19 below.

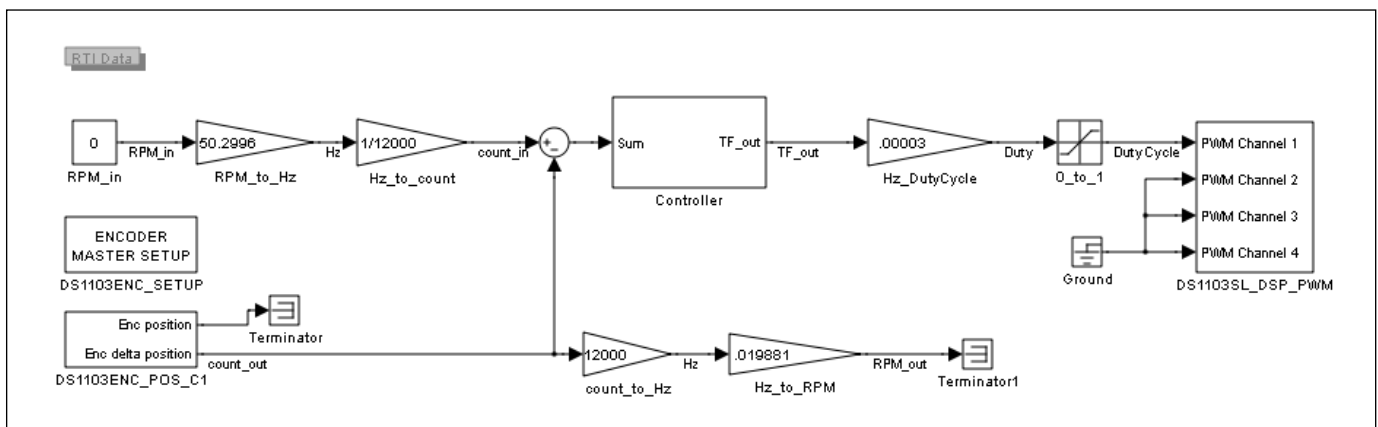


Figure 18: Simulink model downloaded to the DS1103.

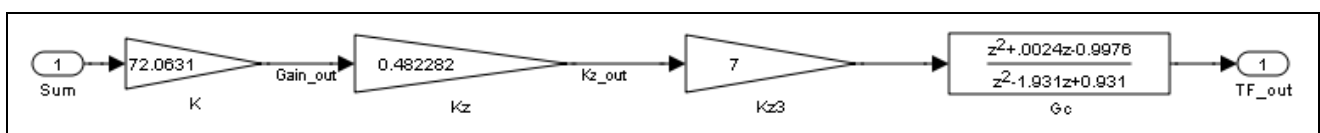


Figure 19: Controller subsystem shown in figure 18 above.

When the controller from the Simulink simulation, which is shown in figure 10 on page 12, was added to the Simulink model, a very similar rise time was observed in the data recorded in ControlDesk for a unit step input, but no overshoot seemed to exist. These results are shown in figures 20-22 below.

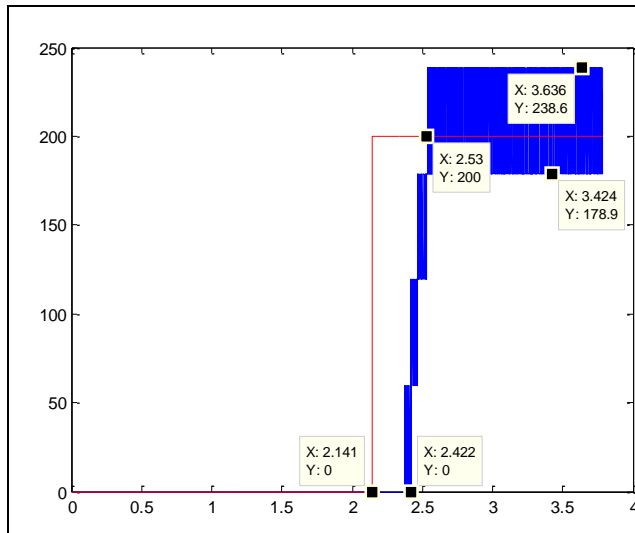


Figure 20: RPM_in and RPM_out for 200RPM input recorded using ControlDesk and plotted using MATLAB.

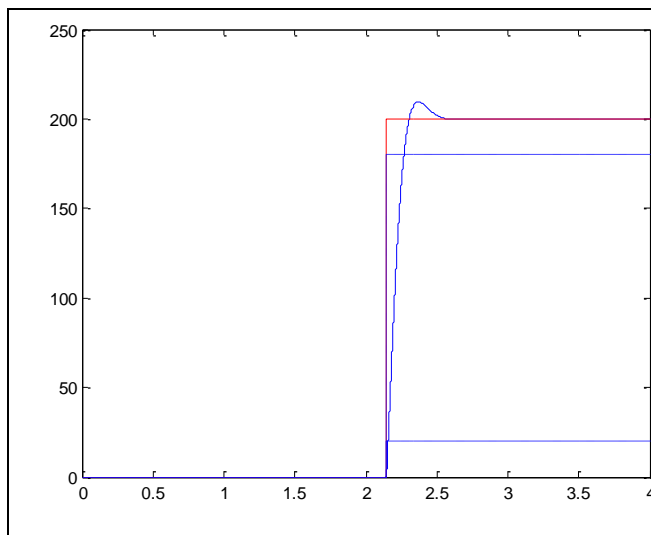


Figure 21: Simulink simulation results when RPM_in of 200RPM is applied at 2.14 s similar to figure 20 above are plotted in MATLAB.

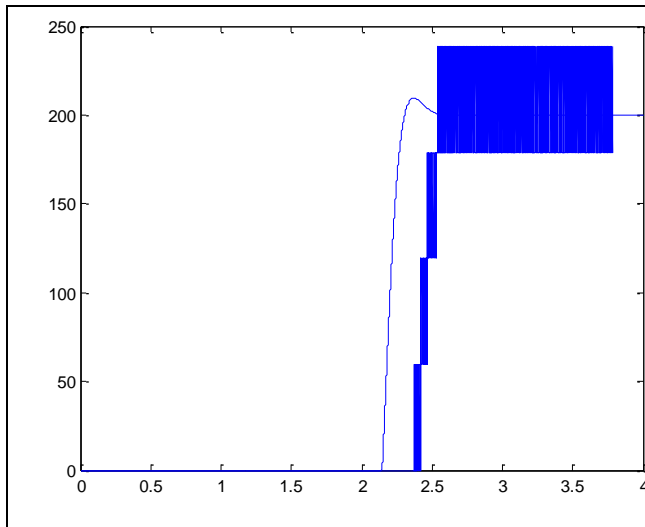


Figure 22: MATLAB plot of RPM_out from figures 20-21 on the previous page.

Figure 22 above shows that the rise times for the Simulink simulation and the actual system are very similar, but it appears there may be some time delay in the actual system. The measurements taken in figure 20 on the previous page from the actual system show a rise time of less than 108 ms and no visible overshoot. This plot does not prove there is no overshoot in the actual system since the error that cannot be entirely removed in the system due to various types of rounding causes the output to alternate between a value above and a value below the desired speed in an aperiodic fashion that could hide overshoot. A frequency-to-voltage converter circuit was attached to the encoder output and the oscilloscope to see if any overshoot could be observed. The results are shown in figure 23 below.

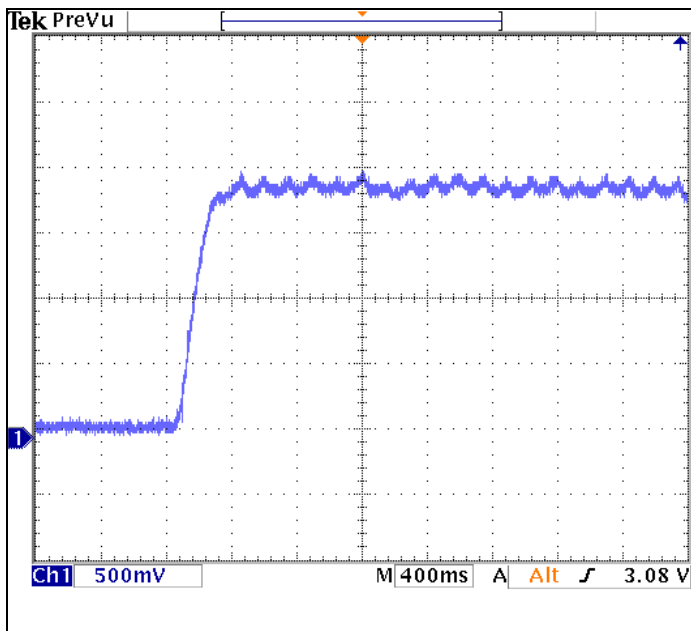


Figure 23: Frequency-to-voltage converter output for a 200 RPM input.

There is still not any overshoot visible. All plots up until this point have had no brake torque applied. Attempts were made to take additional measurements using the frequency-to-voltage converter for several input speeds and voltages applied to the brake. Taking measurements from these plots was still difficult and uncertain. Estimations of rise time measurements from these plots had slower rise times than desired. No overshoot was visible at lower input speeds, and it was only slightly visible, probably in the three to four percent (3-4%) range for higher frequencies when the torque of the brake was near the maximum the system could handle before actually stopping. Additional measurements were also taken at various points in the hardware system for a 238.57 RPM input which is approximately equal to a step input. The results are shown in figures 24-25 below.

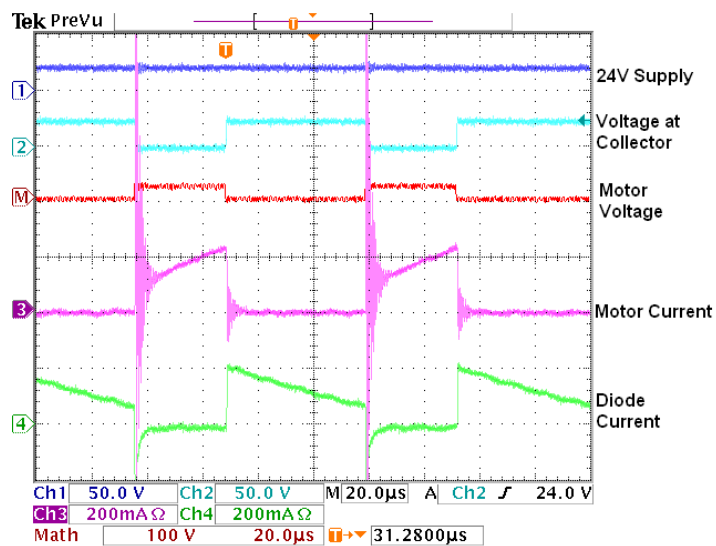


Figure 24: Collector Voltage, Motor Voltage, Motor Current, and Diode Current for a 238.57 RPM input.

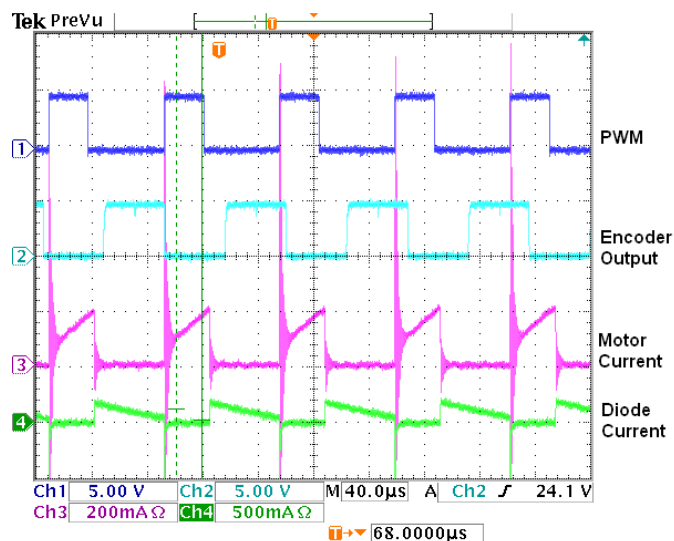


Figure 25: PWM, Encoder output, Motor Current, and Diode Current for a 238.57 RPM input.

From the plots in figures 24-25 above, it is shown that when the PWM is on, the motor current increases, the diode current is zero and the collector voltage of the transistor is small. When the PWM is off, the decreasing motor current flows through the diode.

The approximate maximum torque the system could handle was found to occur when somewhere between 10.5-11+ Volts was applied to the brake. Generally the system would not start up again until the voltage input to the brake dropped below about 8 Volts. These values correspond to about 0.375 Nm or 50 OzIn of torque being applied by the brake. These values are shown in figure 26 below. These two voltage values are different due to the hysteresis caused by the structure and components of the brake.

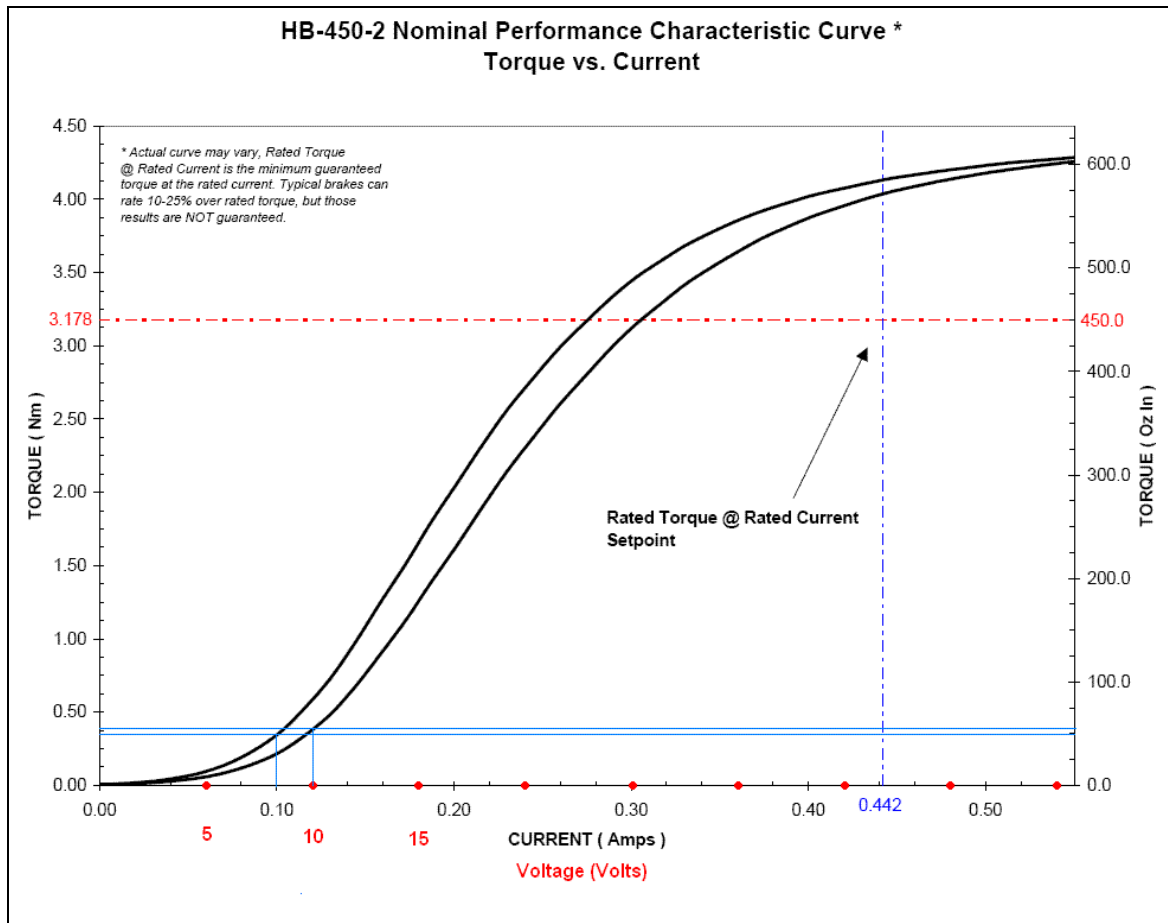


Figure 26: Plot of Torque vs. Current/Voltage response of the brake with approximate cutoff torque/voltage of the brake plotted.

Additional measurements were taken from the output of the frequency to voltage converter circuit and a Pioneer Digital Photo Tach DT-36M. Some of these measurements are shown in table 1 below.

Desired Speed (RPM)	Brake Voltage (V)	Actual Speed Range (RPM)	Speed Error (RPM)	Percent Overshoot (%)	Rise Time (ms)
238.57	0.00	238-239	± 0.5	0	160
	7.12	216-225	-22.5	0	216
	10.64	238-239	± 0.5	0	200
400	0.00	397-400	-3	0	195
	7.19	391-400	-7/+4	0	195
	10.60	398-401	-2/+1	3.20	344
600	0.00	598-602	± 2	0	120
	7.15	568	-12	0	171
	10.47	414-416	-186	4	325

Table 1: Measurements taken from frequency-to-voltage converter output and Digital Photo Tach.

Rise times recorded in table 1 above are longer than the desired specifications. It is possible the system does not meet the specification for 110 ms rise time, but it might also be an issue related to the frequency-to-voltage converter circuit, since the measurements generated from the data recorded using ConrolDesk shows the response for a 200 RPM input is well within specifications. The output of the frequency-to voltage converter was also not very smooth which made taking measurements difficult for table 1 above. If there was any overshoot with lower voltages applied to the brake, it was impossible to see. Even the overshoots that were seen at near cutoff brake values were hard to detect and are approximations, but they do show that even for the higher speed inputs, the specification for less than five percent (5%) overshoot seems to still be met.

Conclusions

The goal of this project was to design and simulate a motor speed controller using MATLAB and Simulink and then implement the controller on the DS1103 board using the dSPACE software. Use of the dSPACE DS1103 workstation to perform this work was also to be documented in the form of a tutorial, and this has been completed as well. The controller requirements were relaxed when discrepancies were found between that MATLAB results and the actual system/Simulink simulation. It is still uncertain as to whether the rise time and overshoot specifications are met or not. Writing the tutorial [4] is the most important focus of this project due to its possible aide to future students in designing and implementing more complex projects such as magnetic levitation systems using the DS1103 workstation. It will allow them to spend significantly less time learning how to use the workstation which will provide them with more time for design and testing their system or learning about some of its higher-level abilities.

References

- [1] "DS1103 PPC Controller Board", Germany: dSPACE, July 2008.
- [2] "Connector and LED Panels," *Catalog 2008*, Germany: dSPACE GmbH, 2008, p. 302.
- [3] *ControlDesk Experiment Guide For ControlDesk 3.2*, Germany: dSPACE GmbH, 2008, Release 6.1.
- [4] Annemarie Thomas. "dSPACE DS1103 Control Workstation Tutorial and DC Motor Speed Control: Tutorial", Senior Project Report, Bradley University ECE Department, May 2009.
- [5] *dSPACE System First Work Steps For DS1103, DS1104, DS1005, DS1006, and Micro Auto Box*, Germany: dSPACE GmbH, 2007, Release 6.0.
- [6] *Real-Time Interface (RTI and RTI-MP) Implementation Guide*, Germany: dSPACE GmbH, 2008, Release 6.1.
- [7] Sabbiseti, Amulya Sabbiseti. "Discrete Time Gain Scheduled Adaptive Control of DC Motor Speed", Masters Project Report, Bradley University ECE Department, December 2008.
- [8] *DS1103 PPC Controller Board Hardware Installation and Configuration*, Germany: dSPACE GmbH, 2007, Release 6.0.

Appendix A: MATLAB Code and Results

MATLAB Code

```

display(' ')
display('*****START SIMULATION*****')
ts=1/12000;%Sampling Time = 1/12kHz
wc=157.08;%wc being designed for

%Analog Plant Transfer Function
display('*****Plant*****')
gpanum=[502853];gpaden=[1 1791.32 25827.1];
gpa=tf(gpanum,gpaden)

%Digital Plant Transfer Function with prewarping at wc
[gpdnum,gpdden]=c2dm(gpanum,gpaden,ts,'zoh');
gpd=tf(gpdnum,gpdden,ts)%Digital Transfer Function for plant
zpk(gpd)%Factored Digital Transfer Function for plant

figure(1),rlocus(gpdnum,gpdden),title('Digital Plant Root Locus')
figure(2),axis('square'),zgrid('new'),rlocus(gpdnum,gpdden),title('Digital
Plant Root Locus with Damping Ratio Lines')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%This Section may NOT always be necessary and may need to be commented
out.%%%%

%Analog Controller Transfer Function
display('*****Controller*****')
gcanum=[1 28.7718];gcaden=[1 857.578 0];
gca=tf(gcanum,gcaden)

%Digital Controller Transfer Function for Analog Controller with prewarping
at wc
[gcdnum,gcddden]=c2dm(gcanum,gcaden,ts,'prewarp',wc);
gcd=tf(gcdnum,gcddden,ts)%Digital Transfer Function for controller
zpk(gcd)%Factored Digital Transfer Function for controller

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Open Loop Combined Plant,Controller, and Optical Encoder Gain
display('*****Open Loop Response*****')

ka=72.0631;%Controller Analog gain - Slight Change in gain should not make a
difference.
kd=0.482282/0.000040278;%Controller Digital gain - MUST BE adjusted for
digital gain included in section above?
koe=81.48;%Optical Encoder gain

k2=koe*ka*kd*.000006825;%Remove last gain term later.*****

```



```

gpdnum2=gpdnum*k2;%Add complete Open Loop Gain
[GPCnum, GPCden]=series(gpdnum2, gpdden, gcdnum, gcdden);
GPC=tf(GPCnum, GPCden, ts)
zpk(GPC)

figure(3), rlocus(GPCnum, GPCden), title('Open Loop Root Locus')
figure(4), axis('square'), zgrid('new'), rlocus(GPCnum, GPCden), title('Open Loop
Root Locus with Damping Ratio Lines')

w=0.1:0.1:2000;
[mag, phase, w]=dbode(GPCnum, GPCden, ts, w);

figure(5), margin(mag, phase, w)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Closed Loop Response
display('*****Closed Loop Response*****')

[CLnum, CLden]=feedback(GPCnum, GPCden, 1, 1);
CLtf=tf(CLnum, CLden, ts)
zpk(CLtf)

figure(6), dstep(CLnum, CLden, 1000), xlabel('Samples')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

display('*****END SIMULATION*****')

```

MATLAB Results

*****START SIMULATION*****

*****Plant*****

Transfer function:

$$502853$$

$$s^2 + 1791 s + 2.583e004$$

Transfer function:

$$0.001662 z + 0.001582$$

$$z^2 - 1.861 z + 0.8613$$

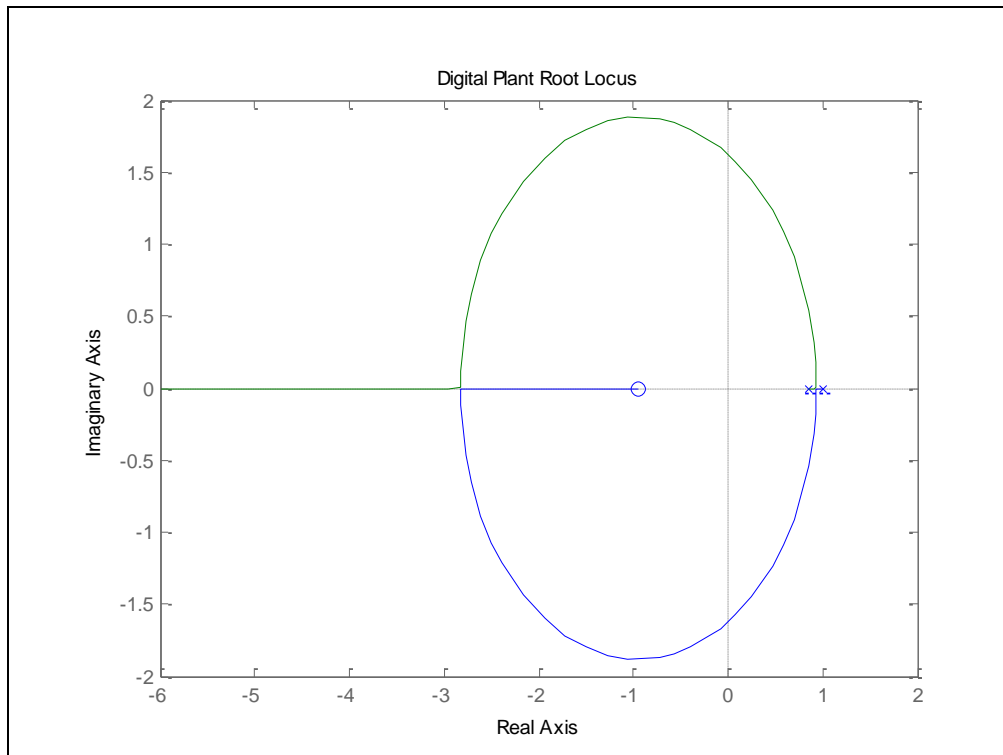
Sampling time: 8.3333e-005

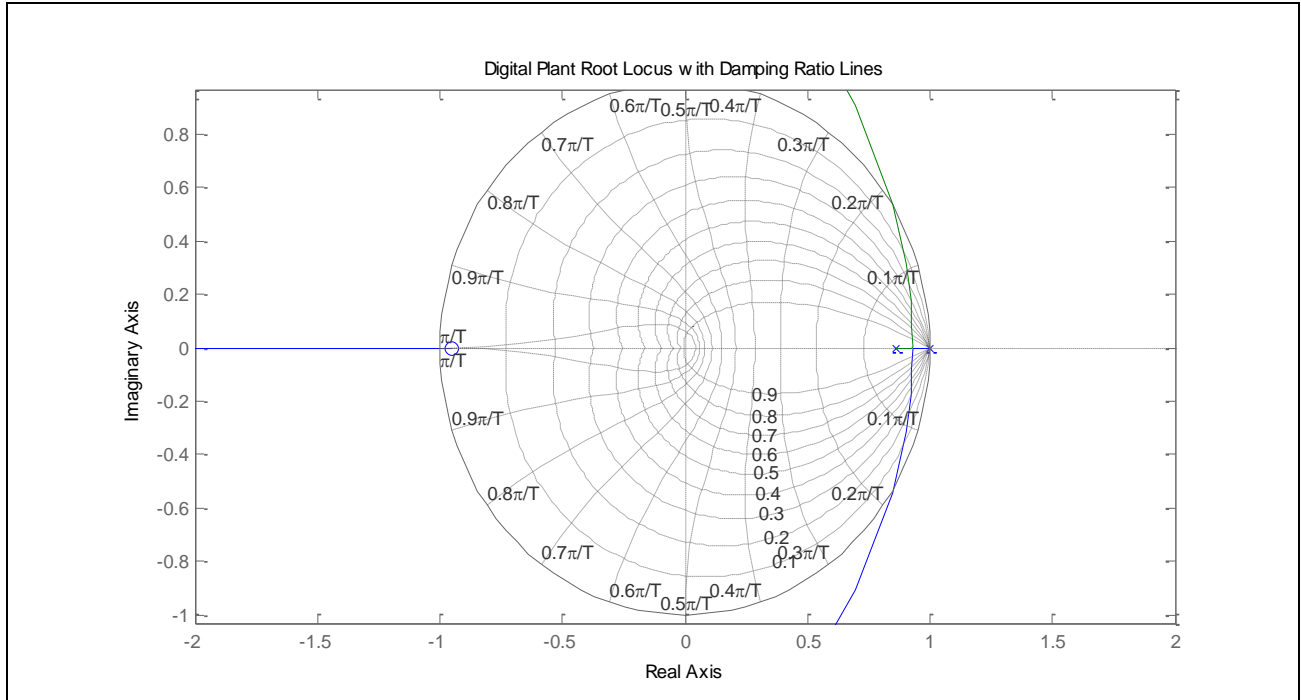
Zero/pole/gain:

$$0.0016623 (z+0.9515)$$

$$(z-0.9988) (z-0.8624)$$

Sampling time: 8.3333e-005





*****Controller*****

Transfer function:

$$s + 28.77$$

 $s^2 + 857.6 s$

Transfer function:

$$4.028e-005 z^2 + 9.646e-008 z - 4.018e-005$$

 $z^2 - 1.931 z + 0.931$
 Sampling time: 8.3333e-005

Zero/pole/gain:

$$4.0278e-005 (z+1) (z-0.9976)$$

 $(z-1) (z-0.931)$
 Sampling time: 8.3333e-005

*******Open Loop Response*******

Transfer function:

$$3.213e-005 z^3 + 3.064e-005 z^2 - 3.198e-005 z - 3.049e-005$$

$$z^4 - 3.792 z^3 + 5.386 z^2 - 3.396 z + 0.8019$$

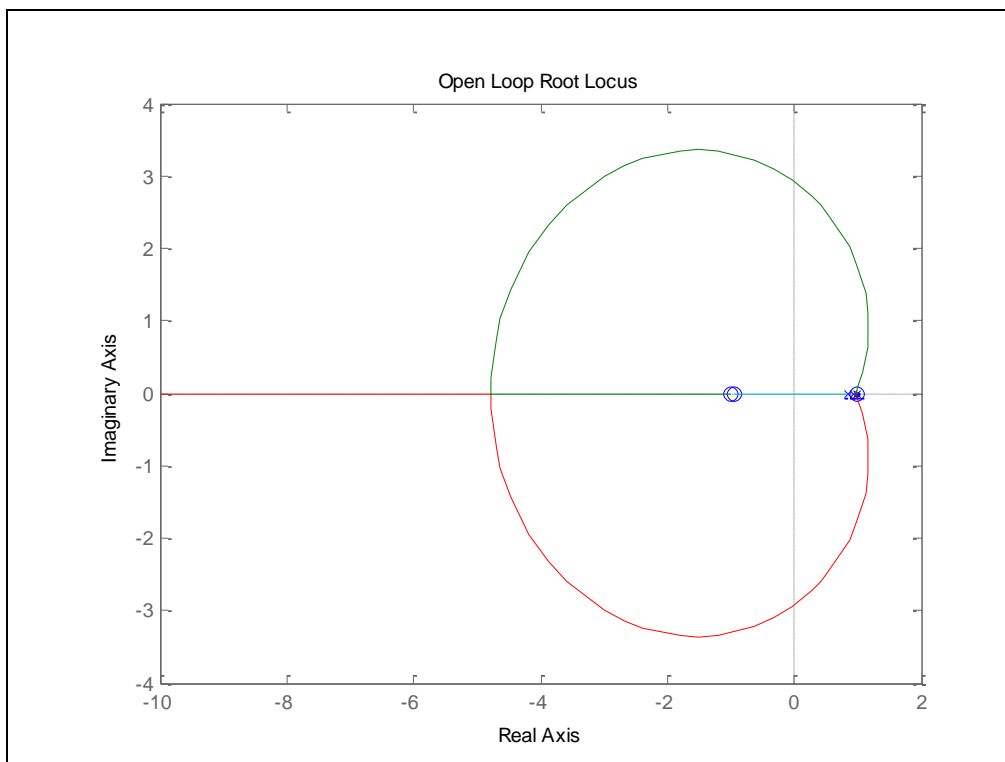
Sampling time: 8.3333e-005

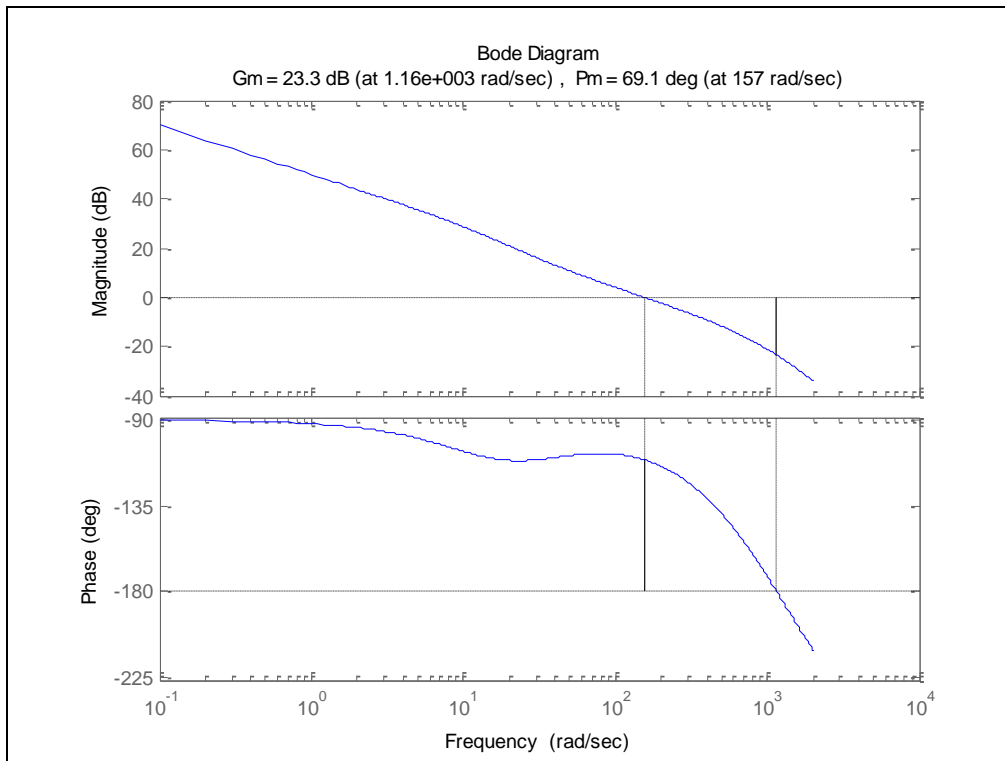
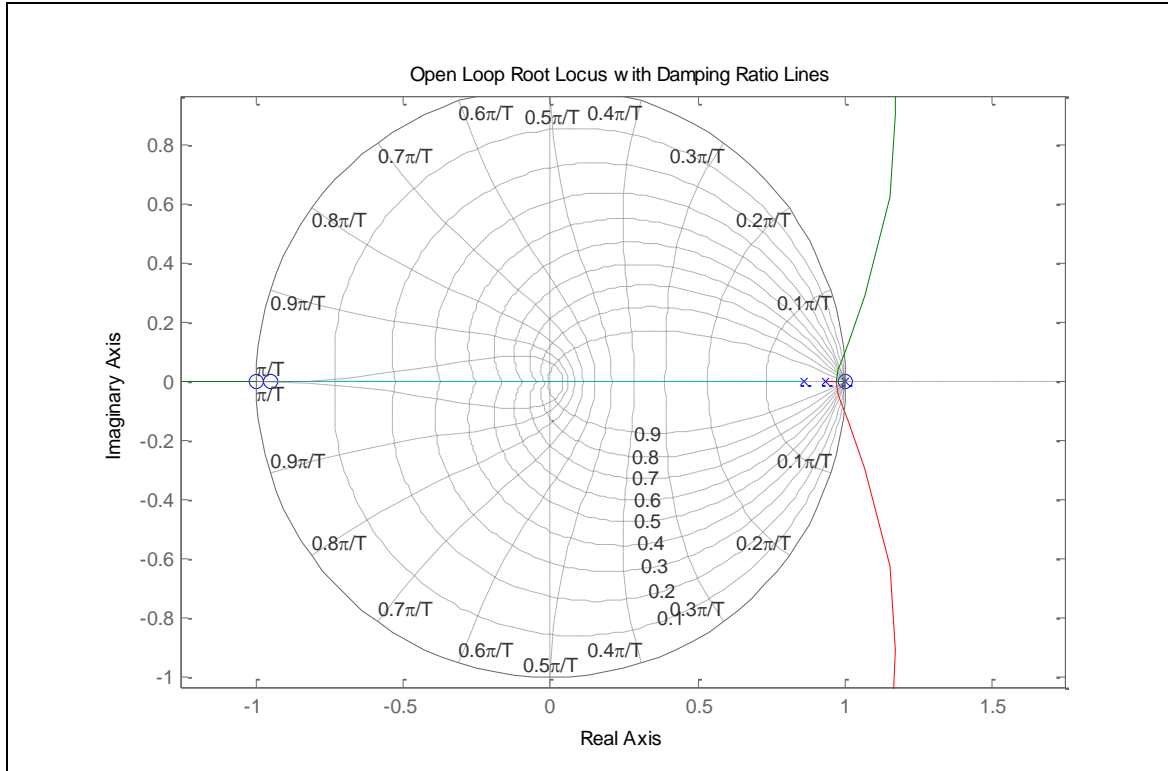
Zero/pole/gain:

$$3.2127e-005 (z-0.9976) (z+1) (z+0.9515)$$

$$(z-1) (z-0.9988) (z-0.931) (z-0.8624)$$

Sampling time: 8.3333e-005





*******Closed Loop Response*******

Transfer function:

$$3.213e-005 z^3 + 3.064e-005 z^2 - 3.198e-005 z - 3.049e-005$$

$$z^4 - 3.792 z^3 + 5.386 z^2 - 3.396 z + 0.8019$$

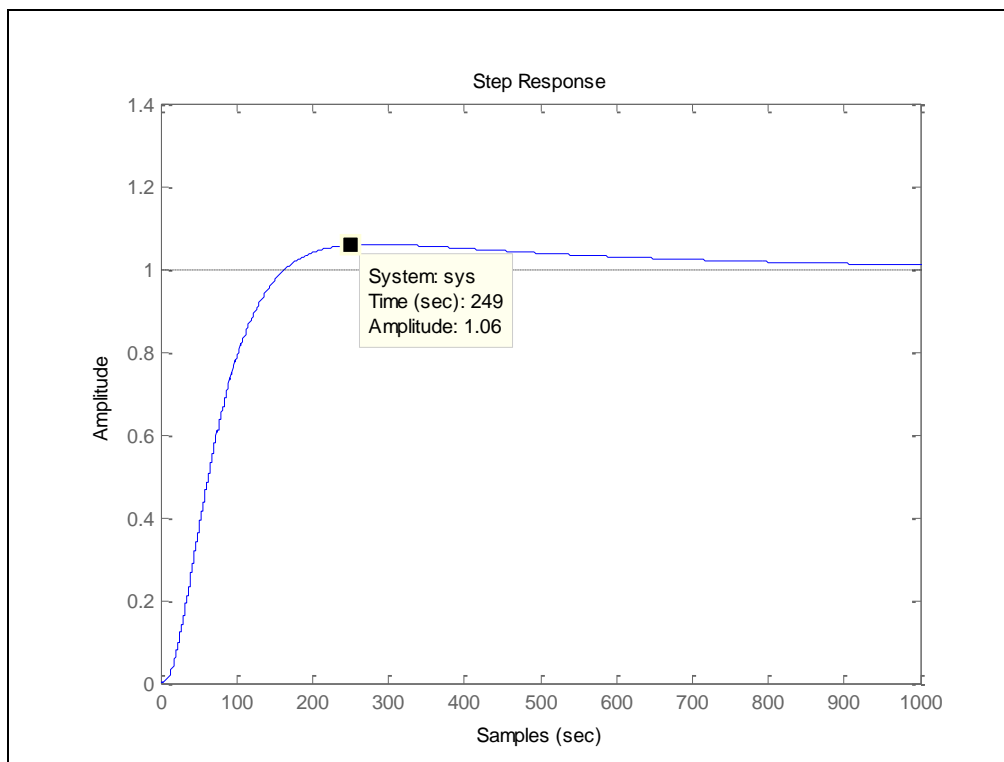
Sampling time: 8.3333e-005

Zero/pole/gain:

$$3.2127e-005 (z-0.9976) (z+1) (z+0.9515)$$

$$(z-0.9973) (z-0.9801) (z-0.9617) (z-0.8531)$$

Sampling time: 8.3333e-005

*******END SIMULATION*******


Appendix B: Project Completion Timeline

Week	Goal	Task	Completed
1	January 27	Write tutorial introduction	January 22
2	February 3	Verify motor parameters	Not Completed
3	February 10	Simulate motor model in MATLAB	February 10
		Design/Simulate controller in MATLAB	March 10
		Design/Simulate controller in Simulink	April 9
4 - 5	February 24	Make required adjustments to model and download to DS1103	March 12
6	March 3	Design/Build/Test hardware for motor subsystem	February 12
7	March 10	Design/Build/Test hardware for optical encoder system	February 17
8 - 10	March 31	Make required adjustments to controller model to work with motor and hardware and download to DS1103	March 12
11 - 13	April 21	Improve controller or Design/Build/Test additional controllers	April 23
14	April 28	Work on presentation/final report/tutorial	May 12
15	May 5	Presentation	May 5
16	May 11	Final Report Due	May 12

Table B1: Schedule of Tasks and actual completion dates for this project for the Spring 2009 semester.

Appendix C: Datasheet Excerpts

Transistor



TIP120/121/122

Medium Power Linear Switching Applications

- Complementary to TIP125/126/127

NPN Epitaxial Darlington Transistor

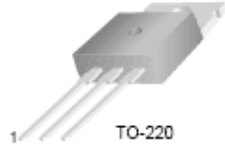
Absolute Maximum Ratings $T_C=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units
V_{CBO}	Collector-Base Voltage	: TIP120	80 V
		: TIP121	80 V
		: TIP122	100 V
V_{CEO}	Collector-Emitter Voltage	: TIP120	80 V
		: TIP121	80 V
		: TIP122	100 V
V_{EBO}	Emitter-Base Voltage	5	V
I_C	Collector Current (DC)	5	A
I_{CP}	Collector Current (Pulse)	8	A
I_B	Base Current (DC)	120	mA
P_C	Collector Dissipation ($T_a=25^\circ\text{C}$)	2	W
P_C	Collector Dissipation ($T_C=25^\circ\text{C}$)	65	W
T_J	Junction Temperature	150	$^\circ\text{C}$
T_{STG}	Storage Temperature	- 65 ~ 150	$^\circ\text{C}$

Electrical Characteristics $T_C=25^\circ\text{C}$ unless otherwise noted

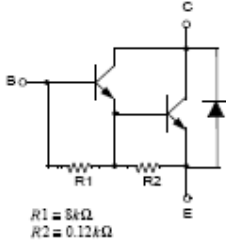
Symbol	Parameter	Test Condition	Min.	Max.	Units
$V_{CEO(sus)}$	Collector-Emitter Sustaining Voltage	$I_C = 100\text{mA}, I_B = 0$: TIP120	60	V
			: TIP121	80	V
			: TIP122	100	V
I_{CEO}	Collector Cut-off Current	$V_{CE} = 30\text{V}, I_B = 0$: TIP120	0.5	mA
			: TIP121	0.5	mA
			: TIP122	0.5	mA
I_{CBO}	Collector Cut-off Current	$V_{CB} = 80\text{V}, I_E = 0$: TIP120	0.2	mA
			: TIP121	0.2	mA
			: TIP122	0.2	mA
I_{EBO}	Emitter Cut-off Current	$V_{EE} = 5\text{V}, I_C = 0$		2	mA
h_{FE}	* DC Current Gain	$V_{CE} = 3\text{V}, I_C = 0.5\text{A}$ $V_{CE} = 3\text{V}, I_C = 3\text{A}$	1000		
$V_{CE(sat)}$	* Collector-Emitter Saturation Voltage	$I_C = 3\text{A}, I_B = 12\text{mA}$ $I_C = 5\text{A}, I_B = 20\text{mA}$		2.0 4.0	V
$V_{BE(on)}$	* Base-Emitter ON Voltage	$V_{CE} = 3\text{V}, I_C = 3\text{A}$		2.5	V
C_{ob}	Output Capacitance	$V_{CB} = 10\text{V}, I_E = 0, f = 0.1\text{MHz}$		200	pF

* Pulse Test - PW=300 μs , Duty cycle $\leq 2\%$



TO-220
1.Base 2.Collector 3.Emitter

Equivalent Circuit



$R1 = 8k\Omega$
 $R2 = 0.12k\Omega$

TIP120/121/122

**SN5404, SN54LS04, SN54S04,
SN7404, SN74LS04, SN74S04
HEX INVERTERS**

S0LS029C - DECEMBER 1983 - REVISED JANUARY 2004

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage, V_I : '04, 'S04	5.5 V
'LS04	7 V
Package thermal impedance, θ_{JA} (see Note 2): D package	86°C/W
DB package	96°C/W
N package	80°C/W
NS package	76°C/W
Storage temperature range, T_{stg}	-65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. This are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES: 1. Voltage values are with respect to network ground terminal.
2. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions (see Note 3)

		SN5404			SN7404			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			2			V
V_{IL}	Low-level input voltage	0.8			0.8			V
I_{OH}	High-level output current	-0.4			-0.4			mA
I_{OL}	Low-level output current	16			16			mA
T_A	Operating free-air temperature	-55			70			°C

NOTE 3: All unused inputs of the device must be held at V_{CC} or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS‡	SN5404			SN7404			UNIT
		MIN	TYP§	MAX	MIN	TYP§	MAX	
V_{IK}	$V_{CC} = \text{MIN}$, $I_I = -12 \text{ mA}$	-1.5			-1.5			V
V_{OH}	$V_{CC} = \text{MIN}$, $V_{IL} = 0.8 \text{ V}$, $I_{OH} = -0.4 \text{ mA}$	2.4	3.4		2.4	3.4		V
V_{OL}	$V_{CC} = \text{MIN}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 16 \text{ mA}$	0.2	0.4		0.2	0.4		V
I_I	$V_{CC} = \text{MAX}$, $V_I = 5.5 \text{ V}$	1			1			mA
I_{IH}	$V_{CC} = \text{MAX}$, $V_I = 2.4 \text{ V}$	40			40			µA
I_{IL}	$V_{CC} = \text{MAX}$, $V_I = 0.4 \text{ V}$	-1.8			-1.8			mA
$I_{OS}¶$	$V_{CC} = \text{MAX}$	-20		-55	-18		-55	mA
I_{CCH}	$V_{CC} = \text{MAX}$, $V_I = 0 \text{ V}$	6	12		6	12		mA
I_{CCL}	$V_{CC} = \text{MAX}$, $V_I = 4.5 \text{ V}$	18	33		18	33		mA

‡ For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

§ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

¶ Not more than one output should be shorted at a time.

**SN5404, SN54LS04, SN54S04,
SN7404, SN74LS04, SN74S04**
HEX INVERTERS

SDLS029C - DECEMBER 1983 - REVISED JANUARY 2004

switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	SN5404 SN7404			UNIT
				MIN	TYP	MAX	
t_{PLH}	A	Y	$R_L = 400\ \Omega$, $C_L = 15\ \text{pF}$				ns
t_{PHL}				12	22	8	

recommended operating conditions (see Note 3)

		SN54LS04			SN74LS04			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			2			V
V_{IL}	Low-level input voltage	0.7			0.8			V
I_{OH}	High-level output current	-0.4			-0.4			mA
I_{OL}	Low-level output current	4			8			mA
T_A	Operating free-air temperature	-55	125		0	70		$^\circ\text{C}$

NOTE 3: All unused inputs of the device must be held at V_{CC} or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	SN54LS04			SN74LS04			UNIT
		MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IK}	$V_{CC} = \text{MIN}$, $I_I = -18\ \text{mA}$	-1.5			-1.5			V
V_{OH}	$V_{CC} = \text{MIN}$, $V_{IL} = \text{MAX}$, $I_{OH} = -0.4\ \text{mA}$	2.5	3.4		2.7	3.4		V
V_{OL}	$V_{CC} = \text{MIN}$, $V_{IH} = 2\ \text{V}$							V
	$I_{OL} = 4\ \text{mA}$	0.25			0.4			
	$I_{OL} = 8\ \text{mA}$				0.25			0.5
I_I	$V_{CC} = \text{MAX}$, $V_I = 7\ \text{V}$	0.1			0.1			mA
I_{IH}	$V_{CC} = \text{MAX}$, $V_I = 2.7\ \text{V}$	20			20			μA
I_{IL}	$V_{CC} = \text{MAX}$, $V_I = 0.4\ \text{V}$	-0.4			-0.4			mA
$I_{OS}§$	$V_{CC} = \text{MAX}$	-20	-100		-20	-100		mA
I_{CCH}	$V_{CC} = \text{MAX}$, $V_I = 0\ \text{V}$	1.2			1.2			2.4
I_{CCL}	$V_{CC} = \text{MAX}$, $V_I = 4.5\ \text{V}$	3.6			3.6			6.6

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5\ \text{V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time, and the duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5\ \text{V}$, $T_A = 25^\circ\text{C}$ (see Figure 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	SN54LS04 SN74LS04			UNIT
				MIN	TYP	MAX	
t_{PLH}	A	Y	$R_L = 2\ \text{k}\Omega$, $C_L = 15\ \text{pF}$				ns
t_{PHL}				9	15	10	



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

**SN5404, SN54LS04, SN54S04,
SN7404, SN74LS04, SN74S04**
HEX INVERTERS

SDLS029C - DECEMBER 1983 - REVISED JANUARY 2004

recommended operating conditions (see Note 3)

		SN54S04			SN74S04			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			2			V
V_{IL}	Low-level input voltage			0.8			0.8	V
I_{OH}	High-level output current			-1			-1	mA
I_{OL}	Low-level output current			20			20	mA
T_A	Operating free-air temperature	-55		125	0		70	°C

NOTE 3: All unused inputs of the device must be held at V_{CC} or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	SN54S04			SN74S04			UNIT
		MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IK}	$V_{CC} = \text{MIN.}$, $I_I = -18 \text{ mA}$			-1.2			-1.2	V
V_{OH}	$V_{CC} = \text{MIN.}$, $V_{IL} = 0.8 \text{ V}$, $I_{OH} = -1 \text{ mA}$	2.5	3.4		2.7	3.4		V
V_{OL}	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 20 \text{ mA}$			0.5			0.5	V
I_I	$V_{CC} = \text{MAX.}$, $V_I = 5.5 \text{ V}$		1			1		mA
I_{IH}	$V_{CC} = \text{MAX.}$, $V_I = 2.7 \text{ V}$		50			50		μA
I_{IL}	$V_{CC} = \text{MAX.}$, $V_I = 0.5 \text{ V}$		-2			-2		mA
$I_{OS}§$	$V_{CC} = \text{MAX.}$	-40		-100	-40		-100	mA
I_{CCH}	$V_{CC} = \text{MAX.}$, $V_I = 0 \text{ V}$		15	24		15	24	mA
I_{CCL}	$V_{CC} = \text{MAX.}$, $V_I = 4.5 \text{ V}$		30	54		30	54	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time, and the duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$ (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	SN54S04 SN74S04			UNIT
				MIN	TYP	MAX	
t_{PLH}	A	Y	$R_L = 280 \Omega$, $C_L = 15 \text{ pF}$		3	4.5	ns
t_{PHL}					3	5	
t_{PLH}	A	Y	$R_L = 280 \Omega$, $C_L = 50 \text{ pF}$		4.5		ns
t_{PHL}					5		





Two Channel Optical Incremental Encoder Modules

Technical Data

HEDS-9000
HEDS-9100

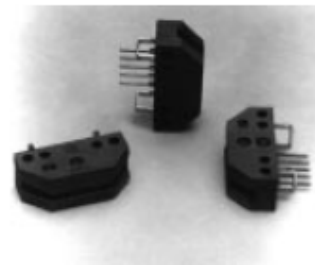
Features

- High Performance
- High Resolution
- Low Cost
- Easy to Mount
- No Signal Adjustment Required
- Small Size
- -40°C to 100°C Operating Temperature
- Two Channel Quadrature Output
- TTL Compatible
- Single 5 V Supply

Description

The HEDS-9000 and the HEDS-9100 series are high performance, low cost, optical incremental encoder modules. When used with a codewheel, these modules detect rotary position. The modules consist of a lensed (LED) source and a detector IC enclosed in a small C-shaped plastic package. Due to a highly collimated light source and unique photodetector array, these modules are extremely tolerant to mounting misalignment.

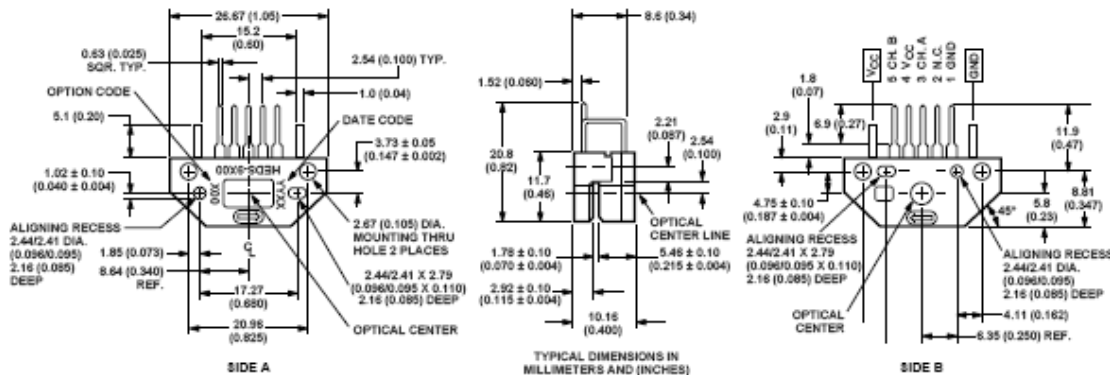
The two channel digital outputs and the single 5 V supply input are accessed through five 0.025



inch square pins located on 0.1 inch centers.

Standard resolutions for the HEDS-9000 are 500 CPR and 1000 CPR for use with a HEDS-6100 codewheel or equivalent.

Package Dimensions



ESD WARNING: NORMAL HANDLING PRECAUTIONS SHOULD BE TAKEN TO AVOID STATIC DISCHARGE.

For the HEDS-9100, standard resolutions between 96 CPR and 512 CPR are available for use with a HEDS-5120 codewheel or equivalent.

Applications

The HEDS-9000 and 9100 provide sophisticated motion detection at a low cost, making them ideal for high volume applications. Typical applications include printers, plotters, tape drives, and factory automation equipment.

Note: Agilent Technologies encoders are not recommended for use in safety critical applications. Eg. ABS braking systems, power steering, life support systems and critical care medical equipment. Please contact sales representative if more clarification is needed.

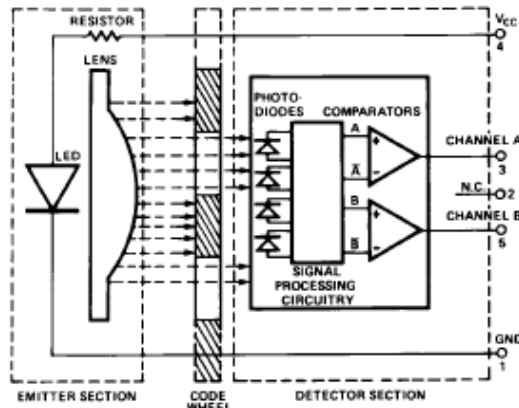
Theory of Operation

The HEDS-9000 and 9100 are C-shaped emitter/detector modules. Coupled with a codewheel, they translate the rotary motion of a shaft into a two-channel digital output.

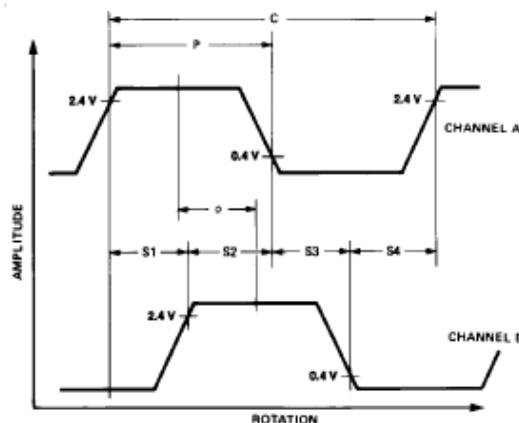
As seen in the block diagram, each module contains a single Light Emitting Diode (LED) as its light source. The light is collimated into a parallel beam by means of a single polycarbonate lens located directly over the LED. Opposite the emitter is the integrated detector circuit. This IC consists of multiple sets of photodetectors and the signal processing circuitry necessary to produce the digital waveforms.

The codewheel rotates between the emitter and detector, causing the light beam to be interrupted

Block Diagram



Output Waveforms



by the pattern of spaces and bars on the codewheel. The photodiodes which detect these interruptions are arranged in a pattern that corresponds to the radius and design of the code-wheel. These detectors are also spaced such that a light period on one pair of detectors corresponds to a dark period on the adjacent pair of detectors. The photodiode

outputs are then fed through the signal processing circuitry resulting in A, \bar{A} , B, and \bar{B} . Two comparators receive these signals and produce the final outputs for channels A and B. Due to this integrated phasing technique, the digital output of channel A is in quadrature with that of channel B (90 degrees out of phase).

Definitions

Count (N): The number of bar and window pairs or counts per revolution (CPR) of the codewheel.

1 Shaft Rotation = 360
mechanical
degrees,
= N cycles.

1 cycle (C) = 360
electrical
degrees (°e),
= 1 bar and
window pair.

Pulse Width (P): The number of electrical degrees that an output is high during 1 cycle. This value is nominally 180°e or 1/2 cycle.

Pulse Width Error (ΔP): The deviation, in electrical degrees of the pulse width from its ideal value of 180°e.

State Width (S): The number of electrical degrees between a

Absolute Maximum Ratings

Storage Temperature, T_S -40°C to 100°C
Operating Temperature, T_A -40°C to 100°C
Supply Voltage, V_{CC} -0.5 V to 7 V
Output Voltage, V_O -0.5 V to V_{CC}
Output Current per Channel, I_{out} -1.0 mA to 5 mA

transition in the output of channel A and the neighboring transition in the output of channel B. There are 4 states per cycle, each nominally 90°e.

State Width Error (ΔS): The deviation, in electrical degrees, of each state width from its ideal value of 90°e.

Phase (φ): The number of electrical degrees between the center of the high state of channel A and the center of the high state of channel B. This value is nominally 90°e for quadrature output.

Phase Error (Δφ): The deviation of the phase from its ideal value of 90°e.

Direction of Rotation: When the codewheel rotates in the direction of the arrow on top of the module, channel A will lead channel B. If the codewheel rotates in the opposite direction, channel B will lead channel A.

Optical Radius (R_{op}): The distance from the codewheel's center of rotation to the optical center (O.C.) of the encoder module.

Recommended Operating Conditions

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
Temperature	T	-40		100	°C	
Supply Voltage	V_{CC}	4.5		5.5	Volts	Ripple < 100 mV _{pp}
Load Capacitance	C_L			100	pF	3.3 kΩ pull-up resistor
Count Frequency	f			100	kHz	$\frac{\text{Velocity (rpm)} \times N}{60}$

Note: The module performance is guaranteed to 100 kHz but can operate at higher frequencies.

Encoding Characteristics

Encoding Characteristics over Recommended Operating Range and Recommended Mounting Tolerances. These Characteristics do not include codewheel/codestrip contribution.

Description	Sym.	Typ.	Case 1 Max.	Case 2 Max.	Units	Notes
Pulse Width Error	ΔP	30	40		°e	
Logic State Width Error	ΔS	30	40		°e	
Phase Error	Δφ	2	10	105	°e	

Case 1: Module mounted on tolerance circle of ±0.13 mm (±0.005 in.).

Case 2: HEDS-9000 mounted on tolerances of ±0.50 mm (0.020").

HEDS-9100 mounted on tolerances of ±0.38 mm (0.015").

Electrical Characteristics

Electrical Characteristics over Recommended Operating Range, typical at 25°C.

Parameter	Symbol	Min.	Typical	Max.	Units	Notes
Supply Current	I_{CC}		17	40	mA	
High Level Output Voltage	V_{OH}	2.4			Volts	$I_{OH} = -40 \mu\text{A max.}$
Low Level Output Voltage	V_{OL}			0.4	Volts	$I_{OL} = 3.2 \text{ mA}$
Rise Time	t_r		200		ns	$C_L = 25 \text{ pF}$ $R_L = 11 \text{ k}\Omega \text{ pull-up}$
Fall Time	t_f		50		ns	

Derating Curves over Extended Operating Frequencies (HEDS-9000/9100)

Below are the derating curves for state, duty, phase and V_{OH} over extended operating frequencies of up to 240 kHz (recommended maximum frequency is 100 kHz). The curves were derived using standard TTL load. -40°C operation is not feasible above 160 kHz because V_{OH} will drop below 2.4 V (the minimum TTL for logic state high) beyond that frequency.

