The Performance and Efficiency of Hydraulic Pumps and Motors

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

David Rossing Grandall

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Co-advisers: Thomas Chase, Perry Li

January, 2010

Acknowledgments

I would like to thank my two advisers, Profs. Chase and Li, for their continued support of my studies at the University of Minnesota.

I also wish to thank my, now many, fellow students for their wonderful help when I needed it, and for the great camaraderie that we shared when things worked and more importantly when things did not work.

This research was funded by the Center for Compact and Efficient Fluid Power (CCEFP), a National Science Foundation Engineering Research Center.

Abstract

This research consists of predicting the performance and efficiency of hydraulic pumps and motors, both with experiments and modeling. A pump and motor test stand is constructed to measure the efficiency of an axial piston swashplate pump/motor unit. A regenerative loop hydraulic system is used to reduce the power requirements of the test stand. The test stand uses an xPC Target data acquisition system. Test conditions focused on low displacement and low speed regimes. Efficiency values ranged from less than 0% to 82%. An existing efficiency model in the literature is fit to the data. Several improvements to the model are suggested. The correlation was satisfactory, but room for improvement still exists. Displacement sensors are recommended in the pump/motor units being tested. This is to avoid the significant uncertainty associated with calculating the derived volume based on the data.

Contents

List of Tables

List of Figures

Nomenclature

ΔP $p_{high} - p_{low}$

$\eta_{mechanical}$ Efficiency, mechanical

$\eta_{overall}$ Efficiency, overall

$\eta_{volumetric}$ Efficiency, volumetric

μ Dynamic viscosity [Pa-s]

ω Speed [RPM]

ε Uncertainty in efficiency measurement

ξ Manufacturer supplied uncertainty

C_f Coulomb friction coefficient

C_s Slip coefficient

C_v Viscous friction coefficient

C_x^* A particular value of a coefficient, C_x, for a given operating condition.

D Full displacement [cc/rev]

K Temperature [Celsius]

$L_{mechanical}$ Loss, mechanical [N-m]

$L_{overall}$ Loss, overall [W]

$L_{volumetric}$ Loss, volumetric [lpm]

p_{high} High pressure [MPa]

p_{low} Low pressure [MPa]

q_{high} Flow, high pressure [lpm]

q_{low} Flow, low pressure [lpm]

T Torque [N-m]

$V_{Clearance}$ Volume of cylinder at top center

V_d Derived volume (displacement) [cc/rev]

V_r Ratio of clearance volume to swept volume

V_{Swept} Volume swept by the piston in a single stroke

X Fractional displacement (0-1)

B Bulk modulus [Pa]

CCEFP Center for Compact and Efficient Fluid Power

CCW counterclockwise

CW clockwise

DAQ data acquisition

DPDT Dual-pole, dual-throw relay

GM1, GM2, GM3, GM4 Grandall Modified 1, 2, 3, or 4, torque or flow models created by the author

HHPV Hydraulic Hybrid Passenger Vehicle

I/O Input/Output

P/M Pump/Motor

PC Personal Computer

S42 Sauer-Danfoss Series 42 28cc/rev pump/motors used on the vehicle

SST Sum of squares total

Chapter 1

Introduction

Chapter 1 presents a brief background of the importance of pump and motor data, and how it can be collected. Relevant previous work is also included in the literature review. Finally, a preview of the rest of the thesis is presented.

1.1 Background

Hydraulic pumps and motors are almost exclusively positive displacement devices. They use a moving boundary to trap a packet of fluid, and then force the fluid into the outlet. Unfortunately, hydraulic pumps and motors have many leakage paths from high pressure to low pressure, so a significant amount of energy is lost to leakage. This loss is known as the volumetric loss. Hydraulic pumps and motors are also mechanical devices, with many moving parts operating at high loads. Consequently, friction develops between the many moving parts, despite hydrodynamic lubrication between many of those parts. This loss is known as the mechanical or torque loss. These losses have their associated efficiencies, and when taken together as an aggregate figure, yield total efficiency. We are generally

interested in this number.

This research was funded under the auspices of the Center for Compact and Efficient Fluid Power (CCEFP). The University of Minnesota is the lead institution in the organization. One of the primary goals of the Center is to migrate fluid power, particularly hydraulics, into the transportation sector as a means of dramatically increasing efficiency. Concerns of climate change have fueled this research.

Specifically, this project is a part of Test Bed 3: The Hydraulic Hybrid Passenger Vehicle (HHPV). It is a small prototype hydraulic hybrid vehicle based on the platform of a Polaris Ranger, a utility vehicle. The HHPV has four hydraulic pump/motors (P/M's), all of the same type, manufacturer series, and displacement. We were able to obtain some sparse efficiency data from the manufacturer, but it proved to be inadequate for constructing an accurate model. In addition, the manufacturer did not actually test this particular size in the series. Rather, a Polymod [20] polynomial-based model was used to extrapolate data from a larger P/M.

In any case, the HHPV requires accurate P/M efficiency data for three purposes: proper component selection, accurate performance and efficiency predictions of the vehicle, and control system optimization. The purpose of the research presented in this thesis is to determine the performance and efficiency of hydraulic pumps and motors by both experimentation and modeling, addressing those three concerns. A P/M test stand was constructed for efficiency measurement purposes. We then used this efficiency data to construct models that predict the efficiency of the P/M's at any operating condition. The test stand is described in Chapter 2, while the modeling process is described in Chapter 5.

1.2 Literature Review

This literature review highlights the work of others previously in three primary areas. Section 1.2.1 describes works pertaining to P/M testing, including the hydraulic design, and the treatment of data. Section 1.2.2 details several P/M efficiency models, and Section 1.2.3 highlights just a few of many methods of improving P/M efficiency.

1.2.1 Pump/Motor Testing

This section describes works pertaining to P/M testing, including treatment of uncertainty, regenerative circuit designs, and test methods. Two relevant ISO Standards are also included.

Manring [17] describes the details of handling uncertainty and suggests that efficiency data must be looked upon skeptically, often with a confidence interval of 10% of the efficiency figures. The paper details all of the potential areas for error to appear, and more importantly, the relative sizes of each error. Manring also suggests the use of several sets of instruments with different full scale readings to reduce the uncertainty. This thesis includes a discussion of uncertainty in Chapter 4.

Zloto [32] describes the computer data acquisition and control system used on a pump test stand. The system described in the paper bears a strong resemblance to the one created for the test stand. It used a personal computer with proprietary software and data acquisition hardware to collect data.

Three papers by Stanzial and Zarotti [31, 30, 26] all present the same information. The papers present the theoretical basis of a hydrostatic regenerative system from a controls, dynamic modeling, and energy regeneration perspective. The system consists of an

electric motor powering a variable displacement pump which behaves as a pressure compensated power supply, a fixed displacement motor and a variable displacement pump. The steady state and dynamic equations are derived, and a methodology for determining the optimal control gains is described. The concept of a regeneration factor is defined, the ratio of the power supplied by the power supply to the internal power recirculated from the pump being tested. While we faced similar tasks in the control of our test stand, the functions of the power supply were outside the scope of our control system as the power supply is a complete packaged unit. The controller for this test stand is described in Chapter 3.

Renvert and Weiler [22] detail the deficiency of starting torque and low speed performance problems in hydraulic motors, several test methods to determine starting torque, the test stand needed to carry out those tests, and the real world relevance of those tests. Three tests are presented: a locked shaft test, a true starting test, and a low speed test. The paper suggests that the low speed test at 1 RPM is the best trade-off between test accuracy and cost. The test stand presented contained equipment for all three tests. The low speed test requires the least equipment since it only requires the motor under test, a tachometer, a torque sensor, the load pump, and a large-ratio planetary reducing gear. Low-speed performance was a concern for the HHPV because version one of the vehicle used P/M's operating at low speeds yet high displacements.

Williamson [27] presents the same procedures as Renvert and Weiler and reaches the same conclusions.

Ichiryu et al. [6] present a significantly different type of test for determining pump efficiency. The test presented uses a thermodynamical method to determine power inflows and outflows. Several thermocouples and flow meters are used to determine input

and output enthalpies of all fluid entering and leaving the pump. The pump case is sealed adiabatically for the test. Results for both a conventional mechanical-based test and the thermodynamic test are presented, and agreement is good. Agreement is best when the pump case and fluid temperature are low, because of reduced heat loss. While conventional mechanical testing was used in this research, this thermodynamic method provides an alternative.

ISO Standard 4409 [1] defines the necessary measurements, symbols, measurement accuracy and suggested tests for pumps and motors. We used these guidelines as a basis for the test procedures presented in chapter 3.

ISO Standard 8426 [2] presents a method of determining the actual displacement of a pump or motor during a test. For a certain displacement set point, several pressure measurements are used to extrapolate a line whose intercept is the actual displacement. We used this method to determine the derived displacement in our tests (Section 4.2).

1.2.2 Pump/Motor Efficiency Models

Section 1.2.2 reviews several models of P/M's, mostly of efficiency. These mathematical models primarily fall into one of three categories: physical, analytical, or empirical. The physical models are based on actual physical dimensions. Gap area could be one of those dimensions. Analytical models fall somewhere between physical and empirical models, often using coefficients to fit general behavior to a limited set of data. Empirical models are little more than curve fits and require large amounts of data and computation time.

Wilson [28] set the foundation for all P/M models. He introduced the concept of torque and volumetric efficiencies as components of overall efficiency, a construct that is still used today. Wilson's model is also the first analytical model constructed. More

importantly, Wilson divided the behavior into three regimes, with different governing equations for each regime. Range 1 covers low speed operation and all flow is assumed to be turbulent. Range 2 covers medium speeds, and the flow is assumed to be laminar. Range 3 covers high speeds, and the flow is also assumed to be laminar. Wilson also claims a decrease in viscosity due to local heating in range 3. Models developed more recently call into question the three regime behavior.

Schlösser [23, 24] describes a physical model. It contains terms which include leakage gap dimensions and areas subject to friction. These parameters would be difficult to determine without data from the manufacturer. The model contains loss terms. But these loss terms correspond only to losses varying with pressure, viscosity, or density. The loss terms do not represent a flow or torque loss due to compressibility or viscous friction, etc. Because it is a physical model, no coefficients are used. This leaves no room for adjustment in fitting the model to the data. Physical models generally have little possibility for adjustment. Because of the lack of adjustment, this model may not be useful.

McCandlish and Dorey [18] is an analytical model. This model seeks to improve the constant loss coefficients found in the Wilson model with the introduction of variable loss coefficients. The McCandlish model also takes into account variable displacement units, whereas the Wilson model was built around fixed displacement gear pumps and motors. One distinct advantage of the McCandlish model is that it requires a minimum of only nine data points for a variable coefficient non-linear model. The graphs presented in the paper show excellent agreement with the data, especially the non-linear loss coefficient predictions. The model is also general enough to be useful for several types of P/M's.

Dorey [4] is a conference paper describing the McCandlish and Dorey journal paper in more detail. It also explains some of the effects of including or not including some specific

non-linear terms. Like McCandlish, this model can be used for several types of P/M's. Dorey also reformulated the McCandlish model slightly for use with least squares fitting with all data points simultaneously, instead of the bins that the McCandlish formulation requires. This is the model chosen for use in this research, and it will be described in detail in chapter 5.

Zarotti and Nervegna [29] provides several analytical models of losses, with some containing as many as seven coefficients. No graphs are presented showing the accuracy of the models, so it is difficult to determine whether the model equations are valid. In any case, these models only apply to axial piston swash plate units. This model was rejected for our use because of its complexity and lack of generality.

Kohmäscher et al. [13] and Kohmäscher [12] are particularly relevant for the Hydraulic Hybrid Passenger Vehicle project and hydraulic hybrid vehicles in general. The paper presents the necessity of accurate data and models for simulating hydraulic hybrid vehicle performance. The most useful portion of the paper is the comparison of the many historical P/M models including physical, analytical, and empirical, and the methods used to calculate the models. The model of choice for this research, Dorey, does not appear in the comparisons of the historical models. Kohmäscher [12] goes further to stress the need for displacement sensors in the P/M units when testing.

Kauranne et al. [11] presents pump efficiency information which is similar to the information produced in this research. Data was collected at many operating conditions. Considerable emphasis is placed on the effect of different hydraulic fluids on efficiency. The importance of fluid temperature is also emphasized. A 20% decrease in torque loss after the 25 hour break-in period is also presented. No change in flow loss was observed over the break-in period. This corresponds to a three percentage point increase in overall

efficiency. Some results of modeling are presented. The Schlösser model was used to model the performance and efficiency of the P/M's tested.

Johnson and Manring [9] presents a model of the torque on the swash plate of variable displacement axial piston pumps. It does not include any information on efficiency. This is an example of a model which explains the behavior of the internal parts of a P/M, but does not directly address performance and efficiency.

Karkoub, Gad, and Rabie [10] details a neural network model of the dynamic behavior of a bent axis pump. While dynamic behavior is outside the scope of this thesis, dynamic models enable simulating transient behavior of hydraulic hybrid vehicles.

Dobchuk et al. [3] does not address P/M losses or efficiency modeling. Instead, he presents a detailed history of dynamic modeling of variable displacement axial piston pumps. A dynamic model based on the consensus of the previous publications is also developed. This dynamic model is also relevant to the operation and control of hydraulic hybrid vehicles.

Hibi and Ichikawa [5] and Inaguma and Hibi [7] both present detailed dynamic models of vane pumps and motors. They also develop the beginnings of a physical-based efficiency model. Both derive full free body diagrams for the vanes in different portions of rotation. We did not use any vane pumps in the HHPV, but the method outlined in the model could be extended to an axial piston P/M for use in a physical-based model.

Mikeska and Ivantysynova [20] describe an empirical method known as "Polymod". It is a polynomial based approach. The coefficients of the polynomials are calculated by a least squares fit to the flow and torque data. The losses are modeled, not the efficiencies. This approach requires a significant number of data points. The model uses three parameters: speed, displacement, and pressure. Temperature and viscosity are not easily

modeled because of the immense computational requirement for a fourth dimension. The model appears to work very well compared to other methods. Like all empirical models, no insight into device operation can be gained by looking at the coefficients.

1.2.3 Pump/Motor Efficiency Improvements

Section 1.2.3 provides a short review of a few of many ideas to improve the efficiency of axial piston P/M units. These improvements generally fall into three categories: decreasing friction, decreasing leakage at the pistons, and decreasing leakage at the port plate.

Manring [16] focuses on the shape of the valve plate slots, with special attention to low displacements. An analytical model is produced, and three slot geometries are tested. P/M's on the HHPV will spend significant time at low displacements to meet the torque requirements at the wheels. Improvements in efficiency at low displacements are especially beneficial to the fuel economy of the vehicle.

Hong and Lee [14] focus on decreasing friction between the valve plate and cylinder barrel at low speeds with the application of a thin CrSiN film on those two parts. The authors predict an improvement in torque efficiency of 1.3%. Wear was negligible.

Payne et al. [21] describes the use of a Digital Displacement pump by Artemis Intelligent Power in a tidal energy converter. Novel high-speed valves are described, along with the general principles of operation. These high-speed valves and intricate valve timing are used to create a variable displacement pump without using a swashplate. No efficiency numbers are presented for the pump.

Seeniraj and Ivantysynova [25] focus on the importance of valve plate design. The design has notable impacts on noise, volumetric efficiency, and control effort. A new

computational tool, CASPAR, is presented. CASPAR was used to create a new valve plate design which reduced swashplate oscillation. Oscillation decreased by 96%. Volumetric efficiency decreased by 0.5%. Swashplate oscillation occurs when the forces on the swashplate are not balanced. This oscillation is the leading cause of structure borne noise from the pump case.

1.3 Overview

Chapter 2 describes the test stand hardware including specifications, hydraulic schematic, sensors, and the test fixture. The controls system is highlighted in Chapter 3, including the Simulink diagram, the control algorithm, and the experimental method. Chapter 4 consists of an explanation of the data reduction method, the derived volume calculations, and a discussion of uncertainty. Chapter 5 covers the topic of modeling, including the Dorey model and some modifications for improving that model. Chapter 6 presents results obtained from the model developed in Chapter 5, and it includes several loss and efficiency graphs. Finally, Chapter 7 presents several conclusions coming from this research, and it also makes several recommendations for future work. Appendix A includes all fixture part drawings. A printed circuit board contains much of the sensor interface circuitry, and the schematic for the board is shown in Appendix B. Appendix C contains important examples of the MATLAB computer code used for data acquisition, data reduction, and modeling in this research.

Chapter 2

Test Stand Hardware

Chapter 2 provides the details of the P/M test stand physical apparatus. The term "test stand" refers to the entire assembly of all equipment including P/M units, mounting brackets, sensors, computers, etc. "Fixture" refers only to the assembly of mounting brackets and plates which support the P/M units and torque sensor. Figure 2.1 contains a flow diagram of how Chapter 2 will proceed, and how the various parts interact. Section 2.1 shows a photograph of the complete test stand and a brief introduction to its parts. Section 2.2 lists specifications for the test stand. These specifications then define the arrangement of the hydraulic components in the hydraulic schematic, explained in Section 2.3. Details about the fixture are found in Section 2.4. The complete list of sensors and relevant information about them is in Section 2.5. Section 2.5 also details the required signal conditioning hardware and the electrical block diagram. Section 2.6 describes the data acquisition (DAQ) hardware.

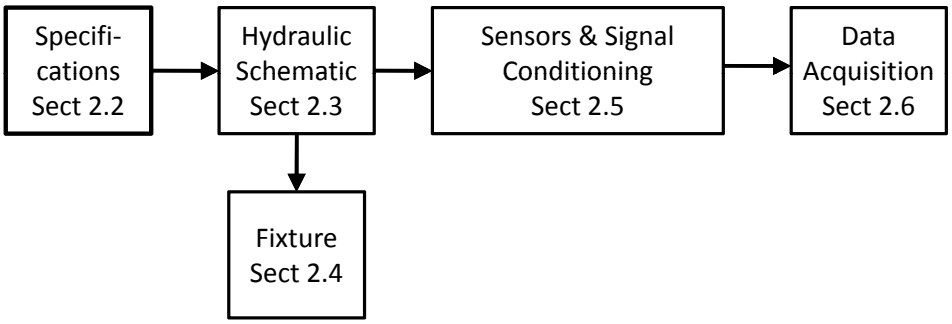


Figure 2.1: Chapter 2 Specifications Inheritance Diagram

2.1 Complete Test Stand Photo and Diagram

Figure 2.2 shows a photo of the complete test stand. A block diagram showing the equipment pictured in the photograph is also included. The "Host" and "Target" personal computers (PC) and monitors are a part of the xPC Target data acquisition system explained in Section 2.6. The "Electronics Box" houses most of the signal conditioning equipment explained in Section 2.5. The two P/M's labeled "S2" and "S" serve as the Source and the Sink, and their use will be explained with the hydraulic schematic in Section 2.3. P/M "T" is the unit being tested. Shaft couplings connect the P/M units and the torque sensor.

2.2 Specifications

This section explains the specifications used in the development of the test stand.

We decided to construct our own P/M test stand to test the Sauer-Danfoss Series 42 28cc/rev units on the HHPV and any others that might be used on the vehicle in the future. Initial models of the vehicle showed that a larger 55 cc/rev unit may be beneficial. We decided to set the torque requirements based on a 55 cc/rev unit at 35 MPa (5000 psi) to ensure extra capacity for future tests. This torque specification is 306 N-m. We also

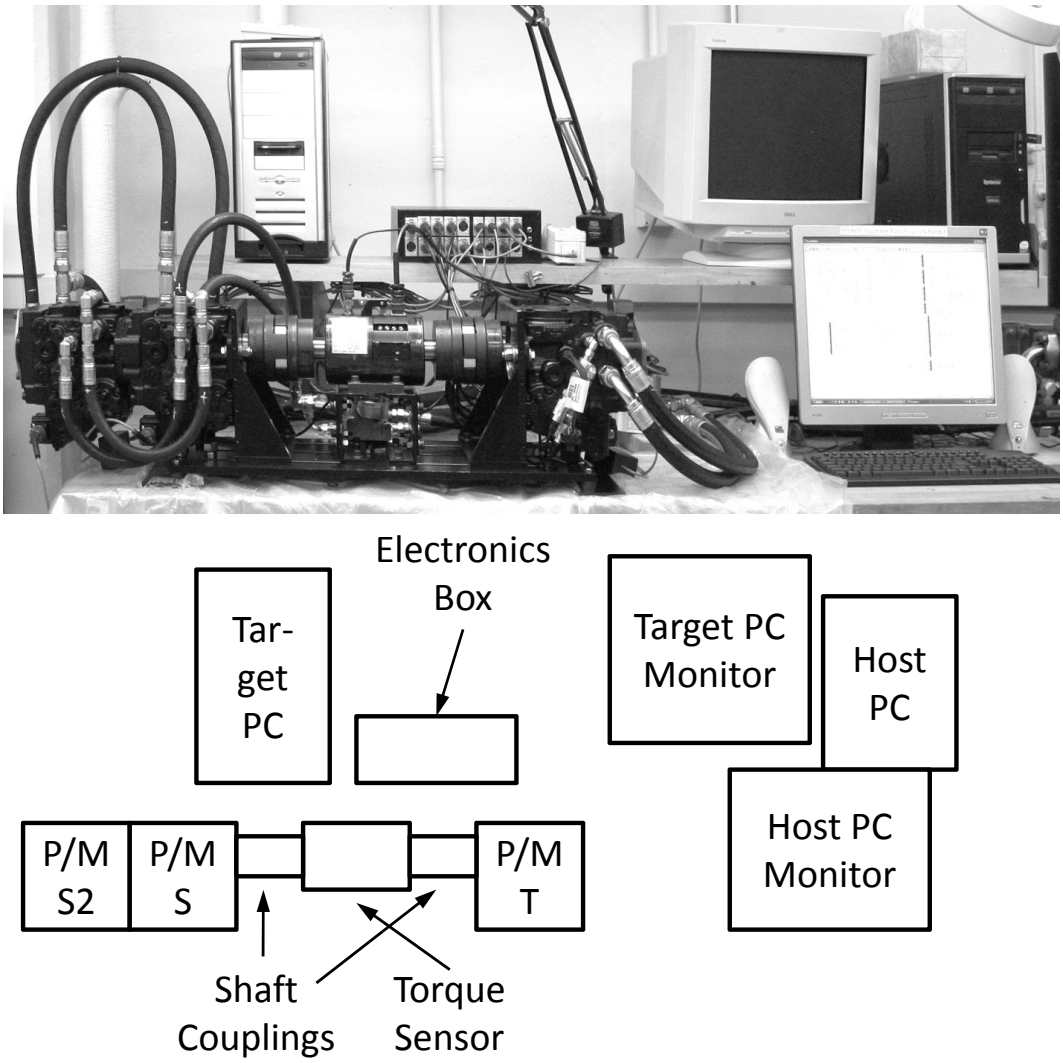


Figure 2.2: Photograph of complete test stand (upper), diagram of equipment shown in photograph (lower).

wanted to test the full speed range of any P/M tested. Most P/M units in this size range have speed capacity of at least 3600 RPM, with excursions permitted beyond 4000 RPM. The maximum speed specification was set at 4500 RPM.

The test stand also required pressure and flow sensors, and we wanted to size these appropriately for the Series 42 (S42) P/M's. The laboratory hydraulic power supply was rated at 20 MPa (3000 psi) at 75.7 lpm (20 gpm), thereby establishing ceilings for pressure and flow. Pressure sensors were selected accordingly. The relatively low flow rating required the use of a regenerative test stand, which recirculates fluid within the stand. This requires the power supply to only make up the losses in the test stand, saving energy. The lab already had two 75.7 lpm (20 gpm) flow sensors. These units were well aligned with the flow requirements of the 28 cc/rev P/M's. A 28 cc/rev P/M operating at 3600 RPM and full displacement has a theoretical flow of 100.8 lpm. The flow sensors would be slightly undersized for this condition. In actual use, however, the power supply proved to be the limiting factor, not allowing us to reach full displacement at full speed because of insufficient flow.

2.3 Hydraulic Schematic

Section 2.3 describes the hydraulic schematic and P/M layout.

Figure 2.3 shows the full hydraulic schematic, as used in the S42 tests. The P/M test stand connects to the lab power supply via a pressure reducing valve where the desired pressure can be set. The power supply and pressure reducing valves are shown on the left side of the schematic. Charge pressure for all of the P/M units is provided by a second tap from the power supply and second pressure reducing valve, always set at 1.3 MPa

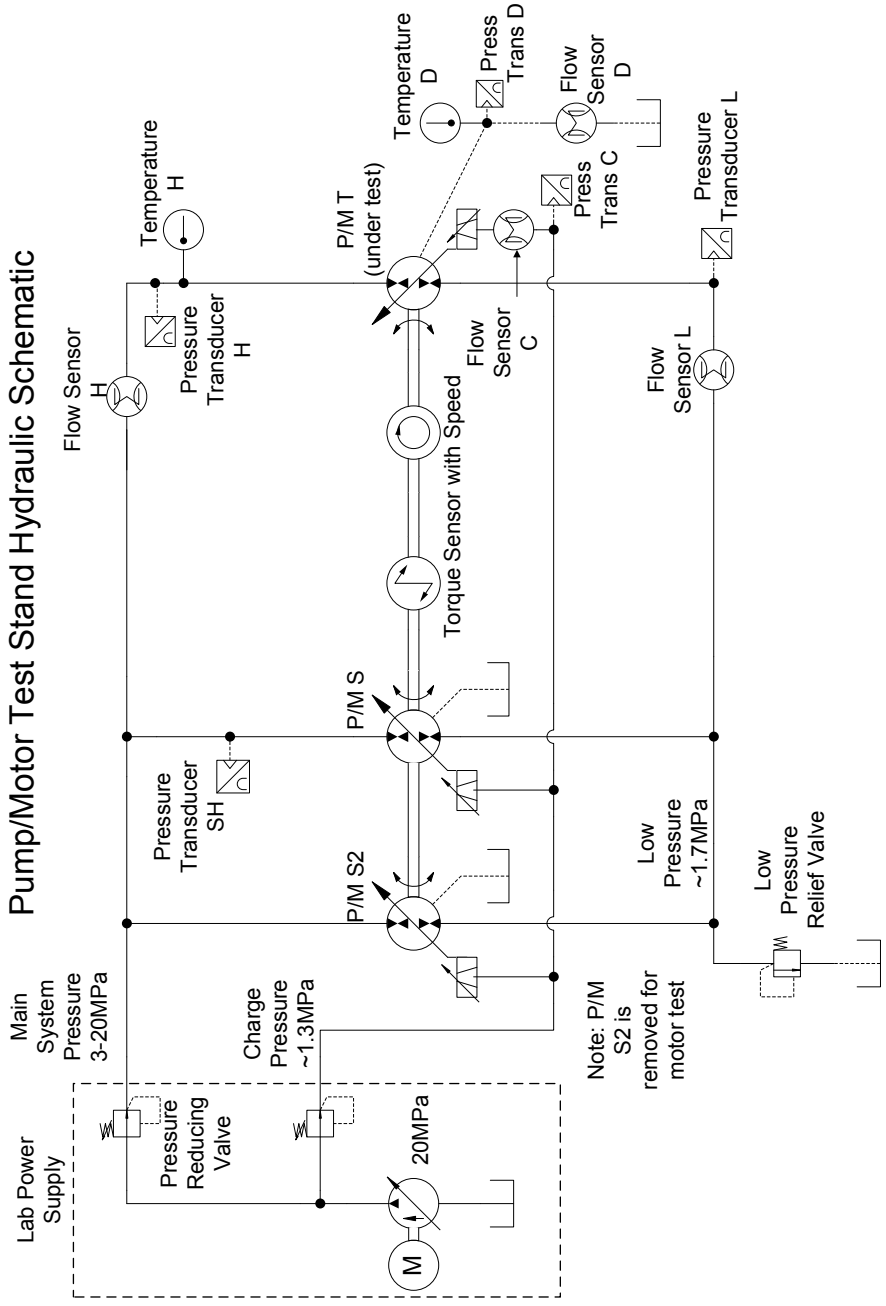


Figure 2.3: Test stand hydraulic schematic for S42 testing

(180 psi).

Three S42 P/M's are required for these tests. For testing pump mode, two S42's acting as motors are required in addition to the S42 being tested. This is because the losses in motors reduce the torque output, and the losses in pumps increase the torque input. Even with two units acting as motors, several test conditions at low pressure were not achievable because of insufficient motor torque. The three S42 units are shown at the center of the schematic, all plumbed in parallel. Four positive displacement flow sensors were placed at the four hydraulic connections on the unit under test, P/M T: High pressure (H), Low pressure (L), Charge/control pressure (C), and case Drain (D). Pressure sensors were also placed at those four locations using the gauge ports directly on P/M T. An additional pressure sensor was placed on the high pressure port of P/M S (Source/Sink). P/M S2 is removed for motoring tests, since it is not needed and only adds stiction to the system. Type K thermocouples were placed at the high pressure port and the case drain outlet.

2.4 Fixture

This section details the layout and purpose of the fixture system.

Figure 2.4 is a numbered part drawing, which will be used to describe the fixture. In its simplest form, the fixture consists of two L-brackets on which the P/M's are mounted, and a long plate that the L-brackets attach to via long slots. The slots allow for adjustment left and right. The torque sensor mounts to a set of adjustable brackets in the center. Shop drawings and a three dimensional color view can be found in appendix A.

The main base plate (10) is that plate that everything else mounts to. The plate has six long slots that the L-brackets mount to via T-slot nuts. In the center are the holes

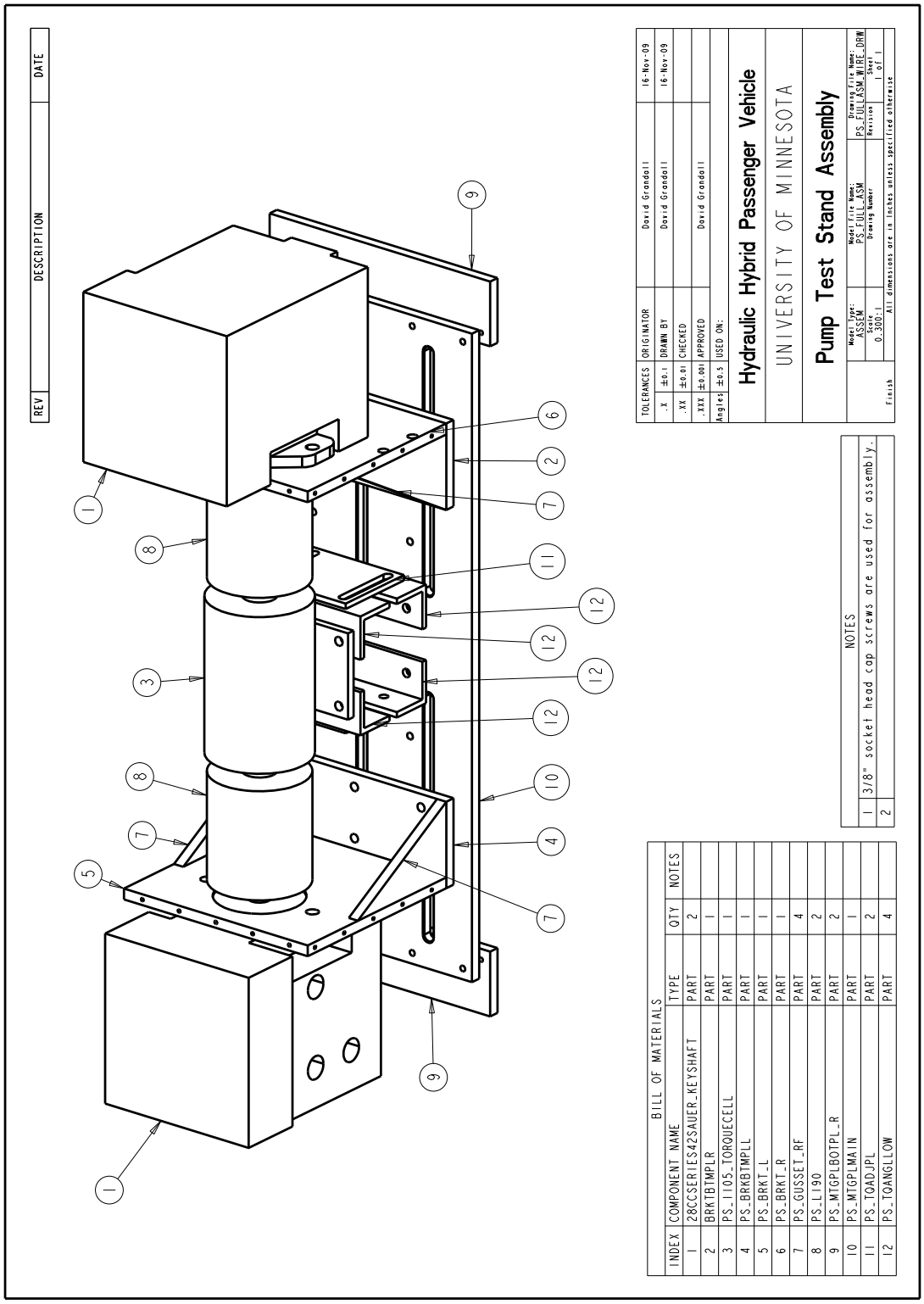


Figure 2.4: Pump test stand fixture numbered part drawing

for mounting the torque sensor bracket assembly. Finally, on the extreme left and right sides, five holes are drilled for two sets of five rubber vibration isolators. A pair of small mounting feet (9) on either side receive the other end of the isolators and contact the table top on which the entire apparatus rests.

The second major portion of the fixture is the two L-bracket assemblies. Each is made up of four parts: a mounting plate for the P/M's (5 & 6), a bottom plate that mounts to the main base plate's slots (2 & 4), and two gusset plates to add rigidity (7). The assembly is held together with counterbored socket head cap screws. The bottom plate has six holes to interface with the main base plate's slots, two per slot. The mounting plate has a pattern of holes around three of its four edges. These holes are for future use, perhaps a guard. The mounting plate's main feature is a set of through holes arranged and sized for the mounting flange of the P/M (1). The large center hole is for the P/M's flange boss, and the remaining smaller holes are for bolts to attach the P/M to the plates.

The last major portion of the fixture is the torque sensor mounting stand. It consists of four pieces of angle bracket (12), two mounting to the main base plate (10) and two mounting to the torque sensor (3). The torque sensor angle bracket and base plate angle bracket are then connected by two small plates (11) with long slots. The slots allow for a large amount of vertical adjustment. The stand uses two pairs of Lovejoy L190 shaft couplings (8) between the torque sensor and the two P/M's.

2.5 Sensors, Signal Conditioning & Block Diagram

Section 2.5 lists the sensors used and their properties. The signal conditioning used for some of the sensors is also explained. The electrical block diagram for all sensors is

shown. Tables 2.1 & 2.2 show the placement and function of each sensor along with relevant electrical and calibration data.

The test stand uses a significant amount of circuitry between some of the sensors or end devices and the DAQ boards. Most of this circuitry adapts the voltage or current from the sensor outputs or controller inputs into a form that the DAQ inputs and outputs can handle. The four flow sensors, the speed sensor, and the pump direction command signal all pass through some form of signal conditioning. Figures 2.5 and 2.6 show block diagrams of the DAQ system.

Figure 2.5 shows the block diagram for the sensors. All five of the pressure sensor outputs are connected directly to the DAQ inputs. The DAS1602/12 and QUAD04 are the DAQ boards used in the Target PC, and are explained in Section 2.6. The two thermocouples and the torque sensor each require amplifiers. The torque sensor uses Honeywell's model UV strain gauge amplifier, which is designed to work with the Lebow 1105 torque sensor. The 1105 torque sensor is a strain gage based torque sensor with a slip ring. This amplifier amplifies the signal from approximately $\pm$50mV to $\pm$5V full scale. It also contains a shunt calibration resistor. Both the span and the zero setting can be changed as needed. The span setting changes the gain of the amplifier to correspond to full scale torque at 5V output by using the shunt calibration resistor. The zero setting adjusts the 0V output point to correspond to zero torque on the torque sensor shaft. In actual use, the zero and the span setting were checked daily, but they only required adjustment about once per week. The thermocouples are connected to an Analog Devices AD595CQ which simulates the ice point and also acts as an amplifier.

4N35 opto-isolators are used with the four flow sensors and the pump direction command to protect the DAC6702 DAQ boards from any damaging voltages resulting from

Measure-ment	ID#	Function	Range	Sensitivity	Type	Mfr. Accuracy	Mfr.	Model
Flow	H	High Pressure	75.7 lpm (20 gpm)	0.453 counts/cc	Pos Disp	0.1% full scale	AW Lake	JV-KG
Flow	D	Case Drain	26.5 lpm (7 gpm)	1.708 counts/cc	Pos Disp	0.1% full scale	AW Lake	JV-KG
Flow	C	Charge Pressure	7.6 lpm (2 gpm)	4.259 counts/cc	Pos Disp	0.1% full scale	AW Lake	JV-KG
Flow	L	Low Pressure	75.7 lpm (20 gpm)	0.455 counts/cc	Pos Disp	0.1% full scale	AW Lake	JV-KG
Pressure	H	High Pressure	0-34.5 MPa (0-5000 psi)	0-5V Full Scale	Diaphragm & Strain Gage	0.25% full scale	Omegadyne	PX309
Pressure	D	Case Drain	0-3.45 MPa (0-500 psi)	0-5V Full Scale	Diaphragm & Strain Gage	0.25% full scale	Omegadyne	PX309
Pressure	L	Low Pressure	0-3.45 MPa (0-500 psi)	0-5V Full Scale	Diaphragm & Strain Gage	0.25% full scale	Omegadyne	PX309
Pressure	SH	Source & Sink High Pressure	0-68.9 MPa (0-10 ksi)	0-5V Full Scale	Diaphragm & Strain Gage	0.25% full scale	Omegadyne	PX309
Pressure	C	Charge Pressure	0-3.45MPa (0-500 psi)	0-5V Full Scale	Diaphragm & Strain Gage	0.25% full scale	Omegadyne	PX309

Table 2.1: Flow and pressure sensor listing and description

Measure-ment	ID#	Function	Range	Sensitivity	Type	Mfr. Ac-curacy	Mfr.	Model
Torque	-	Shaft Torque	564.9N-m (416.7 lb-ft)	0-5V Full Scale	Full Bridge Strain Gage	0.25% full scale	Lebow Honeywell	1105
Speed	-	Shaft Speed	20-10000 RPM	60 counts/rev	Speed Ring & Inductive Pickup	0.01% full scale	Lebow Honeywell	-
Temp.	H	High Pressure	0-500 ºC	0.01V/ºC	Thermocouple & Amplifier	1ºC	Omega	TC-K-NPT-E-72
Temp.	D	Case Drain	0-500 ºC	0.01V/ºC	Thermocouple & Amplifier	1ºC	Omega	TC-K-NPT-E-72

Table 2.2: Torque, speed, and temperature sensor listing and description

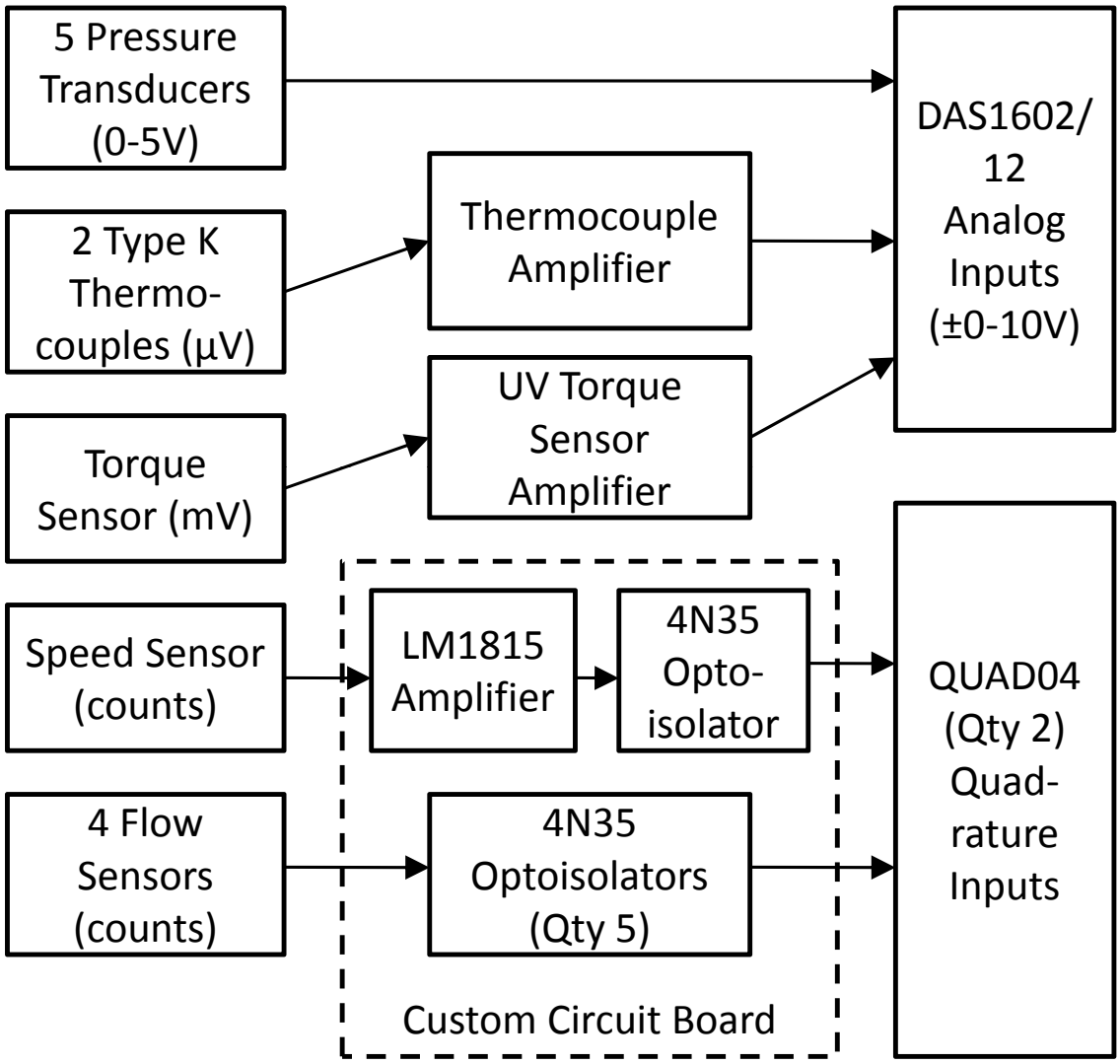


Figure 2.5: Block diagram for all sensors (inputs).

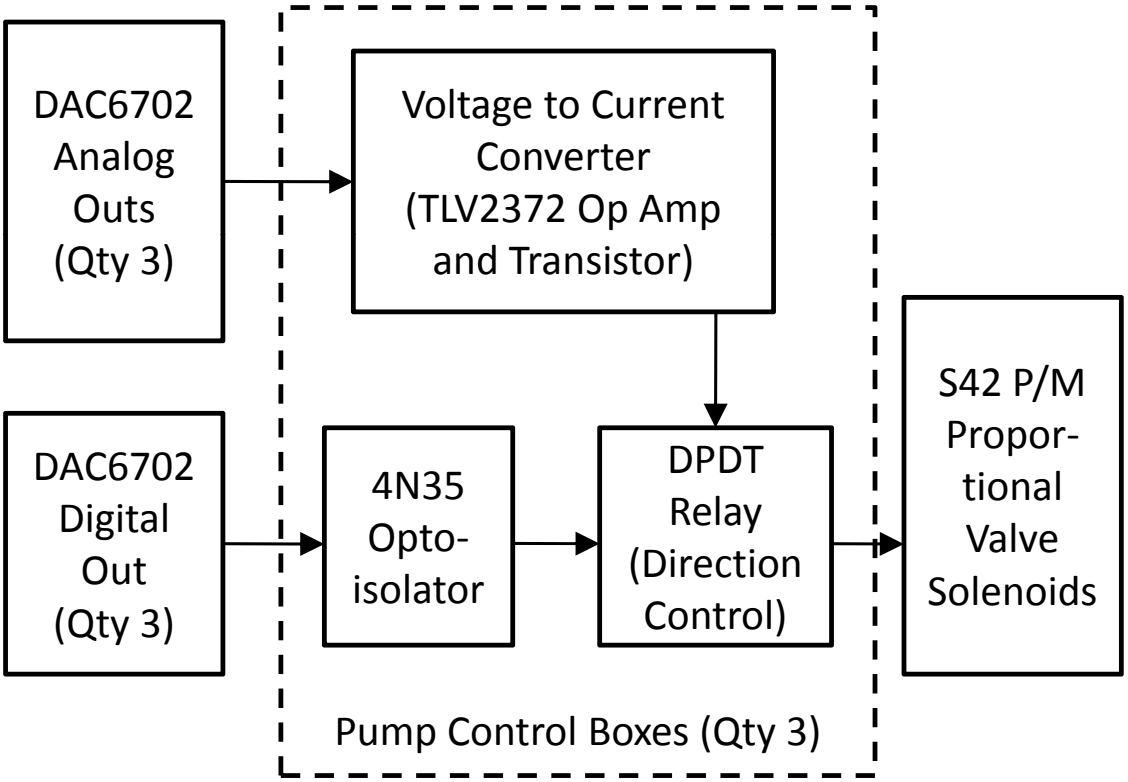


Figure 2.6: Block diagram for P/M controls (outputs).

improper connections or electrostatic discharge. The speed sensor is a special case, since it is a variable reluctance sensor. Both the voltage output and the pulse frequency increase with increasing shaft speed. A National Semiconductor LM1815 adaptive variable reluctance sensor amplifier is built specifically to work with this type of sensor. The amplifier filters the signal and also clamps the output voltage to a constant level. All of the opto-isolators and the LM1815 adapter are housed on a custom printed circuit board. The schematic for this circuit board is found in Appendix B.

Figure 2.6 shows the block diagram for the P/M controls. One analog output and one digital output is used for each P/M unit. The analog output runs to a voltage to current converter built with a 2372 op amp and a transistor. The digital output runs through an

Figure 2.7: Neutrik XLR male connector.

optoisolator which provides sufficient current to run a DPDT relay. This relay switches the direction of the current supplied to the P/M control solenoids. This allows the P/M unit to be controlled in both displacement directions. A small pump control box houses this circuitry for each unit used.

All of the remaining circuitry resides in an aluminum box, which also houses the terminal connectors of the DAQ boards. Neutrik XLR connectors are used for every sensor input and output from the box. These connectors were used because of their locking capability and low-noise characteristics. An example of this connector is shown in Figure 2.7. The maximum current required per pin by the op-amp and transistor was 500 mA. Each pin of the XLR connectors is rated at 10 A.

2.6 Data Acquisition Hardware

Section 2.5 explains the data acquisition hardware.

Model	I/O	Functions
PCI-QUAD04 (1)	Quadrature in	4 Flow sensors
PCI-QUAD04 (2)	Quadrature in	Speed sensor
PCI-DAC6702	Analog Out, Digital I/O	Pump direction and displacement commands
PCI-DAS1602/12	Analog in	Pressure, torque, temperature

Table 2.3: DAQ boards and functions

The test stand uses an xPC Target/Simulink [15] based control and data acquisition (DAQ) system. More details about this system can be found in Chapter 3. Here, the hardware side will be explained.

An xPC Target system consists of two PC's. The first is the Host PC. This PC runs Simulink on a Windows platform and serves as the primary user interface. The second is the Target PC, which is connected via Ethernet to the Host PC. The Target PC runs a proprietary real-time operating system and contains all of the DAQ boards.

Four DAQ boards from Measurement Computing were used. Their model numbers and primary functions are listed in Table 2.3.

Chapter 3

Controls & Data Acquisition Software

Chapter 3 explains all of the software used in the data acquisition system for the test stand, while Chapter 4 describes the software used for data reduction. Section 3.1 presents the Simulink diagram designed for controls and data acquisition in detail. Section 3.2 explains the controls method used for the test stand. Finally, the detailed experimental procedure is outlined in Section 3.3.

3.1 Simulink Diagram

This section shows the Simulink diagram for the test stand. It also details the important portions of the diagram and how they work. The test stand uses an xPC Target system. The hardware portion of the system has been described in Section 2.6. Simulink is used as the front end user interface software and is described here.

Figure 3.1 shows the Simulink diagram used for the test stand. Notes and ovals on the diagram show logical groupings of the icons by purpose. The diagram is broken into smaller portions in Figures 3.2 to 3.4 according to the ovals. Shaded blocks indicate those

blocks which are frequently changed by the user in the operation of the test stand.

Figure 3.2 shows the P/M control section. The displacement of P/M T is controlled with the slider labeled "Pump T Disp". The large rectangle labeled "Pump Commands" is a Simulink subsystem that was developed for use on the HHPV, and is reused on the test stand. It interprets the displacement commands and translates them into useful output signals to go to the P/M control boxes. The "Ref Speed" block is where the user inputs the desired shaft speed. The "Pump Commands" subsystem interprets the command from the controller, detailed in Section 3.2, and sends the appropriate command to P/M's S and S2. P/M S is stroked first, and when it reaches full displacement, P/M S2 is then stroked. The large black bar is a multiplexing block which combines the four signals into one data stream. This data stream is then saved to a file using the "File Scope" block. It is simultaneously shown on the Target PC's monitor via the "Target Scope" block. This enables the user to monitor the current operation of the test stand. The large "Pump Commands" block was originally designed to control four P/M's. The constant, slider gain, and input at the bottom left of the block are not used in this implementation.

The analog sensor input section is shown in Figure 3.3. The outputs from the DAQ board coming into Simulink are shown in the large block on the left. Each signal is multiplied by a gain block. This block represents the conversion factor to convert the voltage signal to a useful unit. The signal is then multiplexed. It then passes to the file and target scopes where it is saved to a file and displayed on the monitor.

Figure 3.4 shows the reference save and flow sensor section. The reference save section saves the commanded pressure, speed, and displacement settings to a file. This marks each data point with the desired values so that the points may be easily organized later. The "Set Pressure" is simply entered by the user, but does not have any effect on the

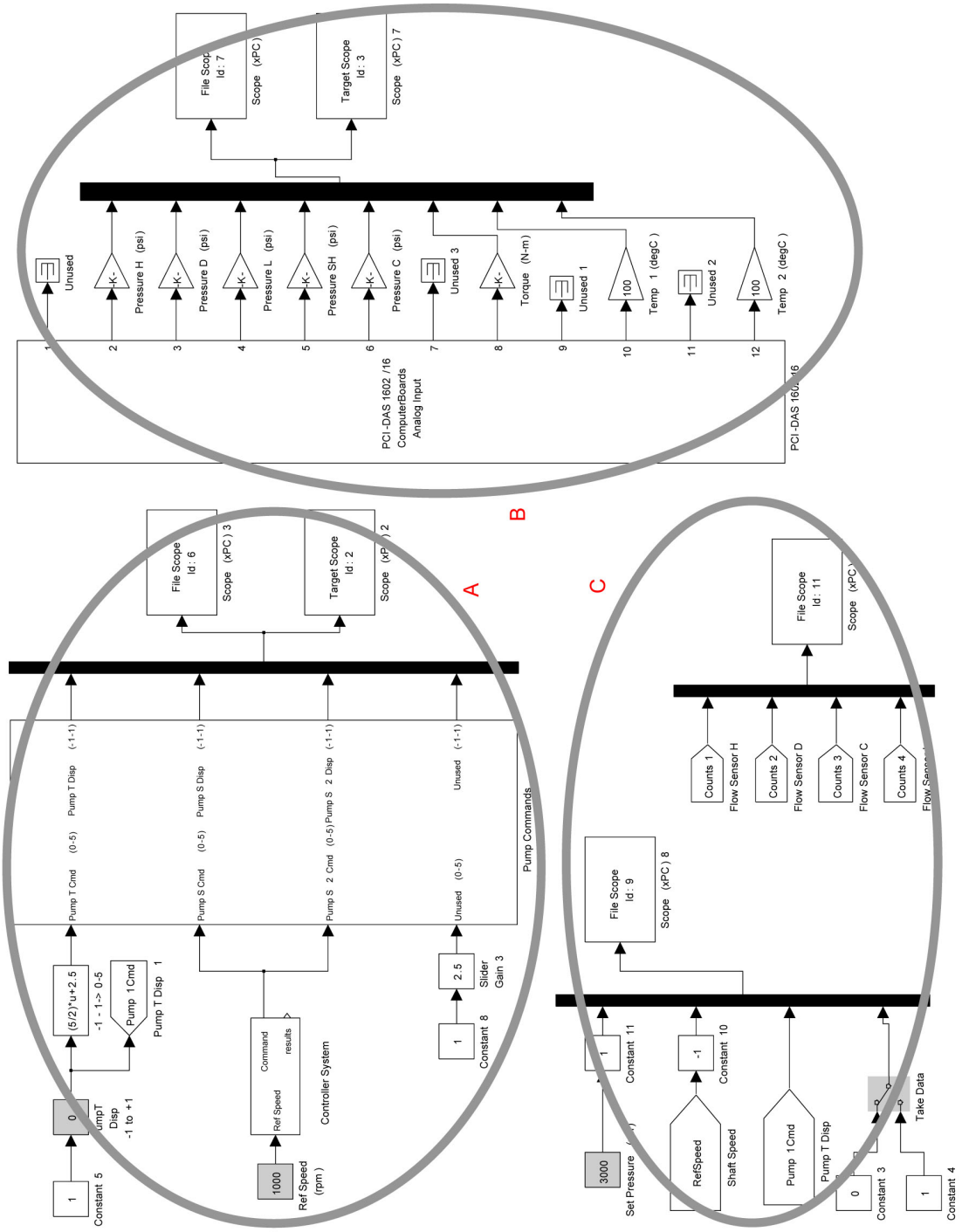


Figure 3.1: Simulink diagram used for the test stand.

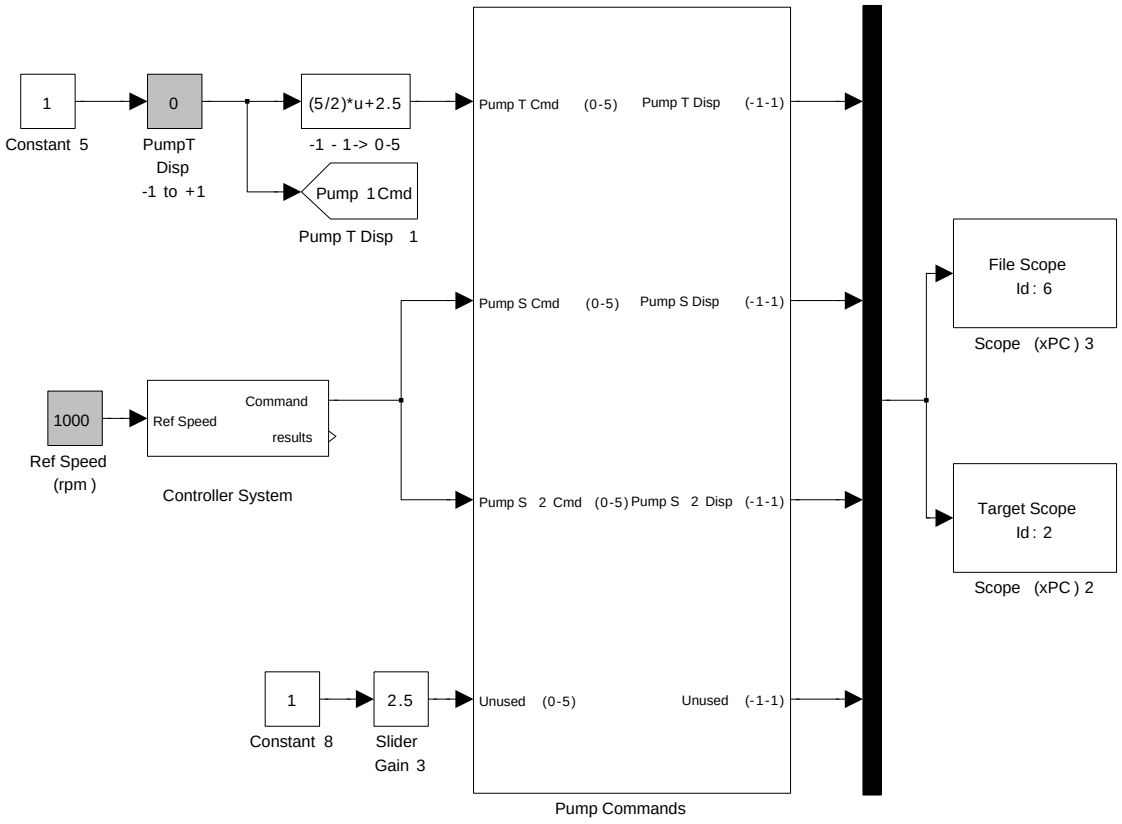


Figure 3.2: Simulink diagram, part A, pump command and control

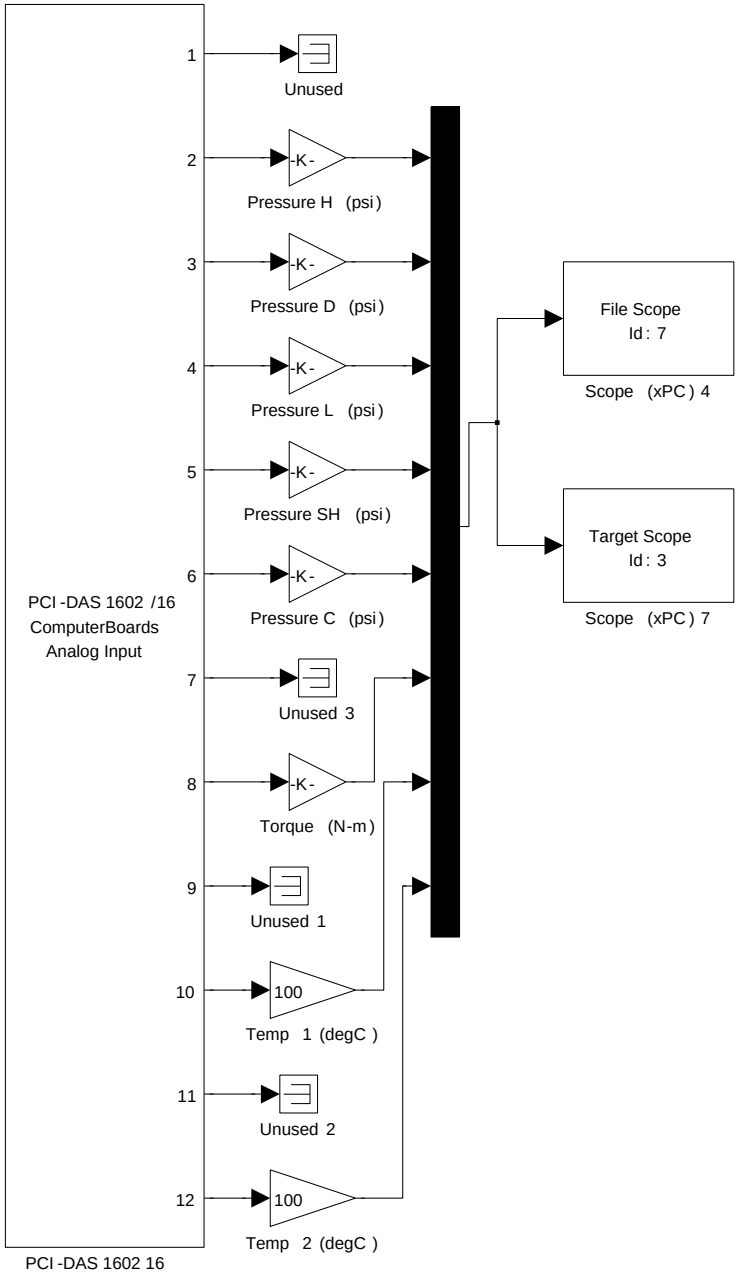


Figure 3.3: Simulink diagram, part B, sensor interface

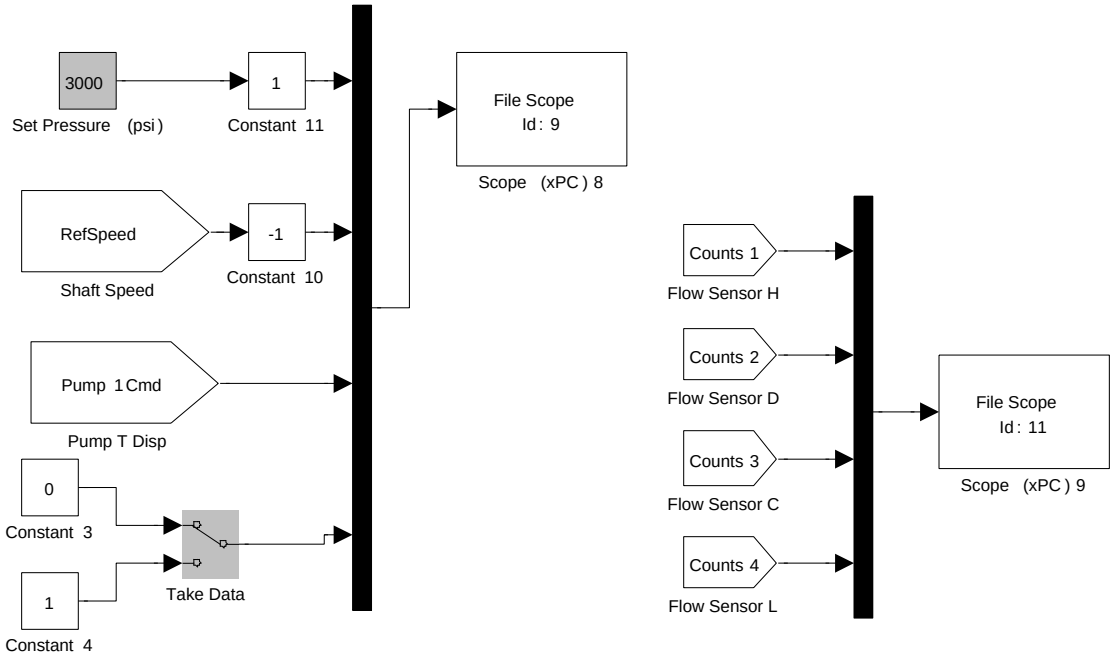


Figure 3.4: Simulink diagram, part C, data saving and flow sensors

actual system pressure setting. This value is only used to organize the data. The actual system pressure measurement is automatically logged by the DAQ system. The "Take data" switch is of particular importance. After initial speed transients have subsided, the user flips the "Take data" switch, and leaves the switch in the "1" position for approximately seven seconds. This triggers the data reduction software to extract those data points for analysis. Further details may be found in Section 4.1. The system saves data to the file scope any time the program is running, regardless of the position of the "Take Data" switch.

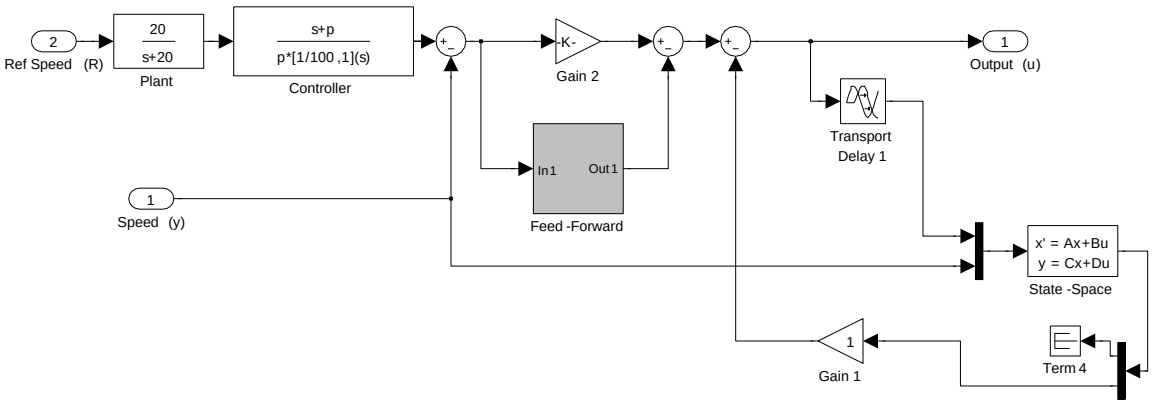


Figure 3.5: Simulink diagram of controller

3.2 Controls Algorithm

Section 3.2 explains the speed control method for the P/M units and how the method was developed.

We discovered early in testing that the P/M swashplate control system has a delay of approximately 10ms. A simple proportional-integral (PI) controller is not sufficient in this case. The system was slow and had significant ringing. Instead, a Smith predictor controller is used. It compensates for the time delay, allowing the system to be fast with minimal ringing. The plant is assumed to be an integrator with a delayed output. Simulink is used to implement this controller, and the implementation is shown in Figures 3.5 and 3.6. Figure 3.5 contains a feed-forward subsystem indicated by a gray box. Figure 3.6 shows the contents of that subsystem.

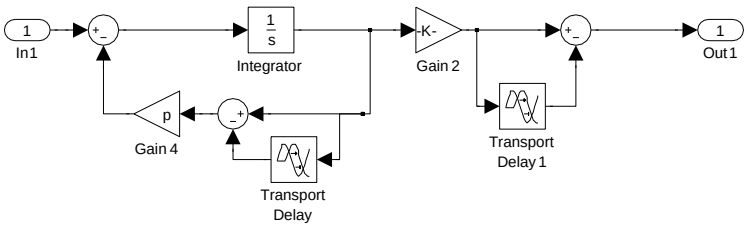


Figure 3.6: Contents of feed-forward subsystem in Figure 3.5

3.3 Experimental Procedure

Section 3.3 explains the experimental procedure used to collect data. The procedure is presented chronologically.

After starting the computers and hydraulic power supply, the Simulink model is opened and compiled. Compilation must occur each time the computers are powered on. This process downloads the model information onto the Target PC of the xPC Target system. When the user is ready to begin, the "Play" button (not pictured) in Simulink is pressed which starts the program. The xPC Target system collects data continuously whenever the program is running.

The desired high and charge pressures are set at the pressure reducing valves.

The hydraulic fluid used in the system is Mobil DTE 25. If the system is cool, below 40 ºC, the test unit is stroked to half displacement to heat the system quickly. Experiments are only conducted if the high pressure oil temperature is between 45 and 55 ºC. Once the system reaches the desired temperature, experiments can begin.

Data collection begins with entering the desired speed and test unit displacement into the Simulink diagram. After any transient speed behavior dies away, the user toggles the "Take data" switch in the Simulink diagram to "1". After approximately seven seconds, the user toggles the "Take data" switch back to "0". The position of this switch instructs

Parameter	Values Tested
Speed	250, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000 RPM
Displacement	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.65, 0.8, 1
Pressure	5.17, 8.62, 12.0, 15.5, 20.7 MPa (750, 1250, 1750, 2250, 3000 psi)
(ΔP)	3.47, 6.92, 10.3, 13.8, 19.0 MPa (500, 1000, 1500, 2000, 2750 psi)
Modes	Pump CCW, Pump CW, Motor CCW, Motor CW

Table 3.1: Experimental parameter values

the data reduction program to extract only the data where the "Take data" switch was set to "1". The remaining data is saved but not used by the data reduction program. The user can then proceed to the next speed setting and repeat the steps in this paragraph.

A sampling rate of 100Hz is recommended. The data reduction system, described in Chapter 4, uses five seconds worth of data, or 500 samples, to create an average for that particular data point. Data collection pauses if the high pressure oil temperature moves outside the desired range.

Experimental parameters were changed in the order presented in Table 3.1. First, the speed was changed through the nine values, then displacement was changed, followed by pressure and finally mode. This resulted in 405 possible test points per mode. Insufficient power supply flow and insufficient source motor torque for pump mode tests limited the number of test points taken per mode to about 300. Test points consisting of high speed and high displacement could not be tested. ΔP becomes a useful measurement in Chapters 5 and 6. Values for ΔP in Table 3.1 are provided for convenience. The low pressure branch was always set at 1.7 MPa (250 psi).

After a group of data points are collected, the data is saved. This process moves the data from the Target PC to the Host PC and reorganizes the data into a data structure in MATLAB. Further information on the data reduction procedure is contained in Section 4.1.

Chapter 4

Data Reduction and Analysis

The data reduction method is explained in Chapter 4. Data reduction concerns all of the procedures used to transform the raw data into a useful form for modeling purposes. Section 4.1 explains the raw data refinement process and the different methods used for the different sensors. The method for determining the derived volume of the P/M units is shown in Section 4.2. A summary of data including some graphs is shown in Section 4.3. Section 4.4 addresses temperature effects by describing a short experiment conducted during the P/M tests. Finally, Section 4.5 addresses the topic of uncertainty.

4.1 Data Reduction

Section 4.1 explains the process of converting the raw test data into usable form.

The primary task of data reduction was calculating a single averaged data point from many actual data points over a five second interval. The sampling rate was 100 Hz, so the 5 second average included 500 samples. The data reduction algorithm used the toggling of the "Take data" switch (see Sections 3.1 and 3.3) to the "1" position as the cue to use

the next 500 samples for one data point. The pressure, temperature, speed, and torque measurements are simply averaged over these 500 samples. This procedure assumes that all measured quantities are constant over the test interval. Speed varied no more than ± 3% in the interval. The only other variability is attributed to pressure ripple and noise.

The flow measurements are handled slightly differently. The flow sensors were of the positive displacement type and they output counts, with each count corresponding to a certain volume. The DAQ boards used with these flow sensors operated as counters. The cumulative number of counts registered, not the actual flow rate, was recorded at the start and end of each five second sampling interval. To determine the actual flow rate, the count total at the beginning of the 500 samples was subtracted from the count total at the end of the 500 samples. This difference was then divided by five seconds and multiplied by a conversion factor provided by the manufacturer. The DAQ boards employed 24-bit counters. Counter overflow was not a concern in this application. A Kalman filter was used to display the instantaneous flow rate on the user interface screen during data collection, but this data stream was not used for analysis purposes.

Overall efficiencies and losses were calculated for each data point using the following equations, as found in ISO Standard 4409 [1]. The superscript "P" indicates pump mode, while superscript "M" indicates motor mode. Explanations of each variable, including units, are provided in the nomenclature section in the front matter of this thesis. Conversion factors of 1000 generally convert liters to cubic meters, and 60 converts minutes to seconds.

$$\eta^{P}_{overall} = \frac{[(q_{high} \cdot p_{high}) - (q_{low} \cdot p_{low})] \cdot 1000}{2\pi \cdot \omega \cdot T} \tag{4.1}$$

$$\eta^{M}_{overall} = \frac{2\pi \cdot \omega \cdot T}{[(q_{high} \cdot p_{high}) - (q_{low} \cdot p_{low})] \cdot 1000} \tag{4.2}$$

$$L^{P}_{overall} = \frac{2\pi \cdot \omega \cdot T - [(q_{high} \cdot p_{high}) - (q_{low} \cdot p_{low})] \cdot 1000}{60} \tag{4.3}$$

$$L^{M}_{overall} = \frac{[(q_{high} \cdot p_{high}) - (q_{low} \cdot p_{low})] \cdot 1000 - 2\pi \cdot \omega \cdot T}{60} \tag{4.4}$$

After the derived volume was calculated, which will be explained in Section 4.2, mechanical and volumetric efficiencies and losses could be calculated using the following equations.

$$\eta^{P}_{volumetric} = \frac{q_{high}}{q_{theoretical}} = \frac{1000 q_{high}}{V_d \omega} \tag{4.5}$$

$$\eta^{M}_{volumetric} = \frac{q_{theoretical}}{q_{high}} = \frac{V_d \omega}{1000 q_{high}} \tag{4.6}$$

$$\eta^{P}_{mechanical} = \frac{T_{theoretical}}{T} = \frac{(p_{high} - p_{low}) V_d}{2\pi \cdot T} \tag{4.7}$$

$$\eta^{M}_{mechanical} = \frac{T}{T_{theoretical}} = \frac{2\pi T}{(p_{high} - p_{low}) V_d} \tag{4.8}$$

$$L^{P}_{volumetric} = \frac{V_d \omega}{1000} - q_{high} \tag{4.9}$$

$$L^{M}_{volumetric} = q_{high} - \frac{V_d \omega}{1000} \tag{4.10}$$

$$L^{P}_{mechanical} = T - \frac{(p_{high} - p_{low})V_d}{2\pi} \tag{4.11}$$

$$L^{M}_{mechanical} = \frac{(p_{high} - p_{low})V_d}{2\pi} - T \tag{4.12}$$

4.2 Derived Volume Calculations

Section 4.2 details the process of calculating the derived volume for each group of data points. A model for calculating the derived displacement given command displacement and speed is also presented.

The displacement of the S42 units does not necessarily follow the command signal precisely. The manufacturing tolerances and deformations of interior components also cause the full displacement value to deviate slightly from the manufacturer's specified value. Therefore, a method to derive the volume of fluid displaced per shaft revolution must be used. ISO Standard 8426 [2] defines this process, and a synopsis is provided here.

This method makes the assumption that no leakage or compressibility losses exist at zero pressure differential. At higher pressures, leakage and compressibility losses decrease the amount of fluid delivered to the outlet in pump mode. Leakage and compressibility losses increase the flow to the inlet in motor mode. For a set of test points at the same displacement command, the values of flow per revolution for each point are plot-

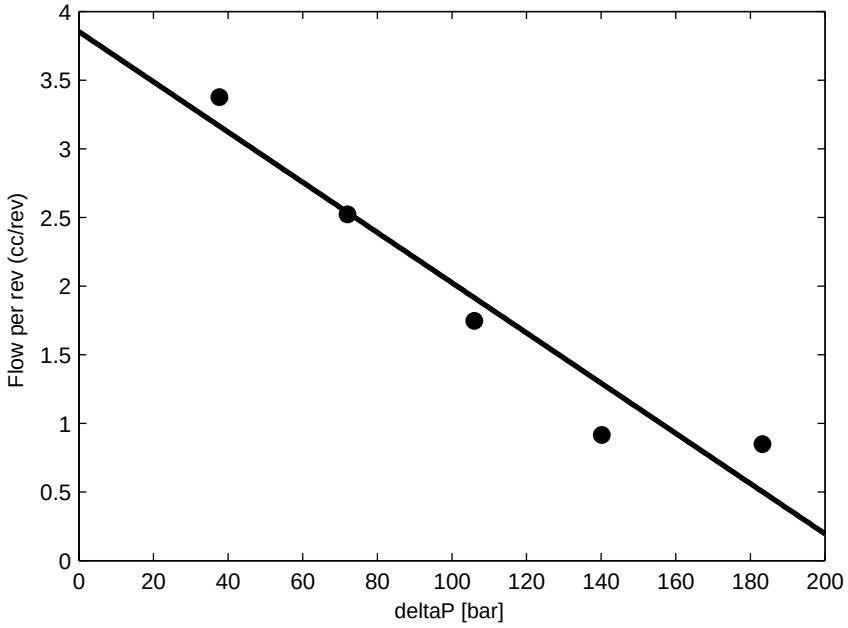


Figure 4.1: Derived displacement example, pump CW, 0.3 displacement command, 500 RPM.

ted at different pressures. A line is extrapolated back through the flow values of the test points to the y-intercept. The y-intercept represents zero pressure, and it is the intersection of the line and the y-intercept which represents the estimated derived volume for that displacement command.

Fig 4.1 shows an example plot using this method. The example shown is for 0.3 command displacement, which should correspond to 8.4 cc/rev for a 28 cc/rev P/M. The graph shows that the estimated displacement is only 3.85 cc/rev, less than half of the command value. The points and extrapolated line show strong agreement with this estimated value. This 3.85 cc/rev value is assigned to all of the points shown on the graph. A similar procedure is used for every other speed and displacement combination for each of the four operating modes.

Mode	a	b	c	d	e	f
Pump CCW	-0.414	-0.667	-0.022	1.30	0.130	0.045
Pump CW	0.561	0.022	0.032	0.378	-0.053	0.021
Motor CCW	0.445	0.798	0.259	0.319	-1.05	0.316
Motor CW	0.106	0.036	0.854	0.038	-0.203	0.117

Table 4.1: Derived vs command displacement coefficient values for each mode.

The relationship between derived displacement and command displacement appears to be non-linear and is also a function of speed. The cause of this behavior is not fully understood. The P/M's internal displacement control system is likely either designed imperfectly or faulty. Polynomial curve fits of the following form were fit to the data.

$$X_{Der} = aX_{Cmd}^2 + b(\frac{\omega}{\omega_{max}})^2 + cX_{Cmd}(\frac{\omega}{\omega_{max}}) + dX_{Cmd} + e(\frac{\omega}{\omega_{max}}) + f \tag{4.13}$$

The correlation appears to be excellent. X_{Der} and X_{Cmd} are derived and command displacements respectively. ω_{max} is 4000 RPM. Table 4.1 shows the value for coefficients a-f for each mode. Figures 4.2 and 4.3 show surface plots for each mode.

4.3 Summary of data

This section presents some of the data in graphical form without the use of models.

Figure 4.4 shows the overall efficiencies from the data points for pump and motor mode. Highest overall efficiencies observed were 82% for pump mode and 80% for motor mode. Both occurred at 20.7 MPa (3000 psi), full displacement, and 2500 RPM. Zero percent efficiency indicates that no energy was transferred from input to output, while negative efficiencies indicate that both the input and output were absorbing power. The input of a pump is its shaft while the output is the high pressure port. The input and

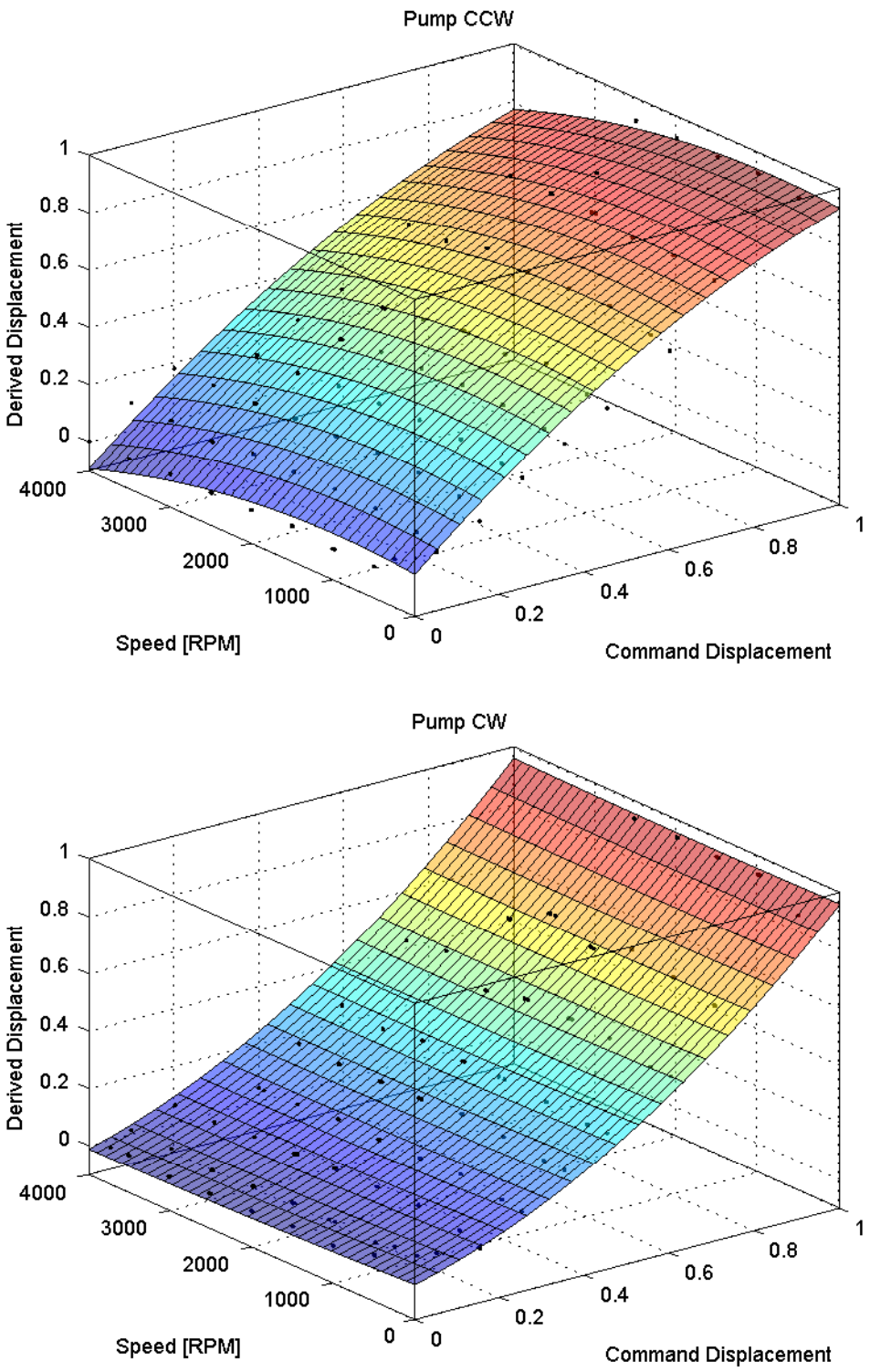


Figure 4.2: Derived displacement vs. command displacement and speed, pump mode

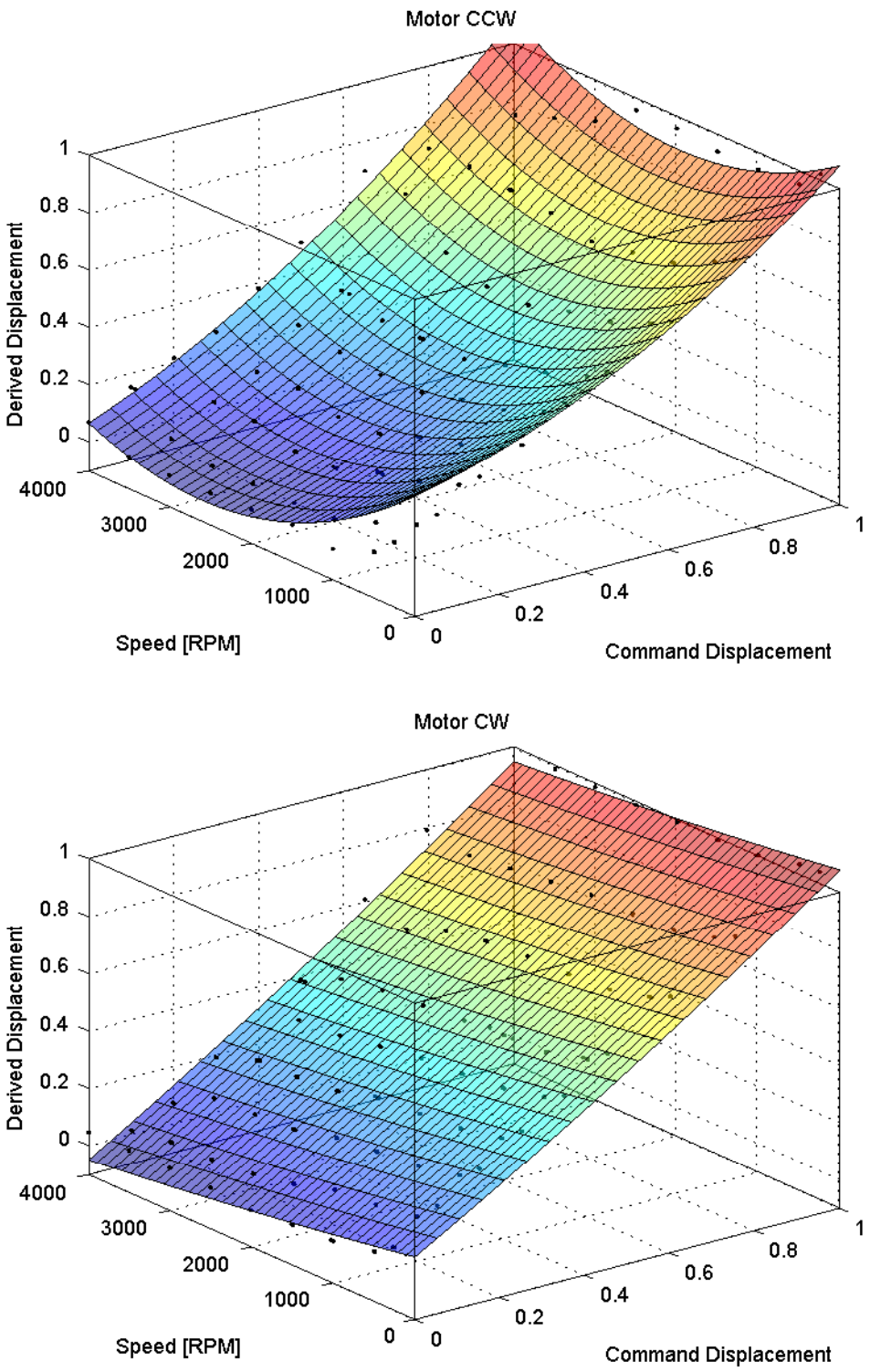


Figure 4.3: Derived displacement vs. command displacement and speed, motor mode

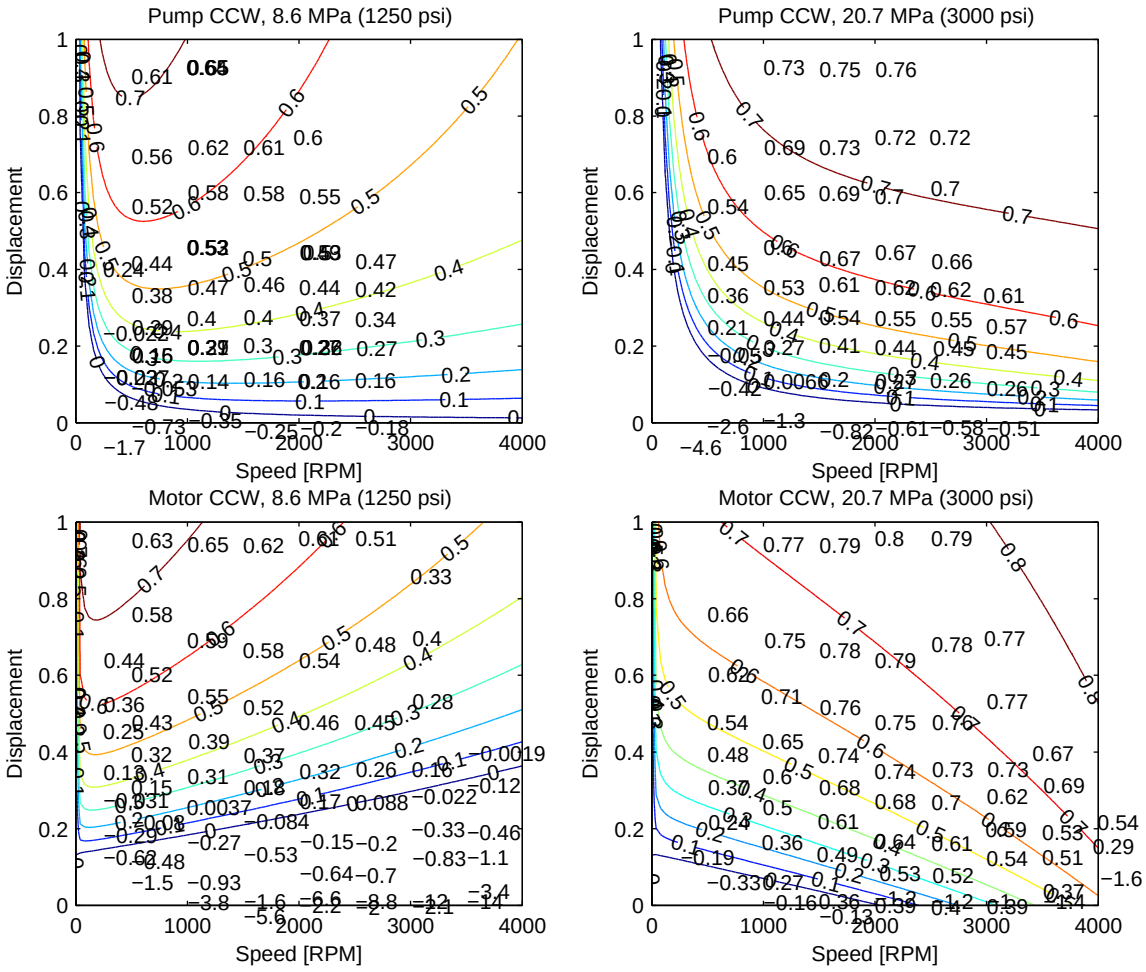


Figure 4.4: Overall efficiencies for pump and motor mode. Displacements are derived.

output of a motor is the reverse of those for a pump. A value of 0.61 indicates 61% efficiency. The contour plots shown in Figure 4.4 are generated by the flow and torque models described in Chapters 5 and 6.

4.4 Temperature effects

An experiment was conducted to determine the effect of fluid temperature on efficiency. The experiment measured overall efficiency as the high pressure oil temperature increased from room temperature to approximately 60ºC. Pressure, speed, and command displacement remained constant for the duration of the test which required approximately 15 minutes to complete. Data was analyzed using the methods described in Section 4.1.

The results are shown in Figure 4.5. Efficiency appears to peak at about 50ºC after a steady rise from room temperature. Above 55ºC, the efficiency begins to decrease. Efficiency appears to be approximately constant within the temperature range from 45ºC to 55ºC. This is the range that was chosen to be acceptable for taking measurements as described in Section 3.3. The change in fluid viscosity is responsible for this behavior. As viscosity decreases, flow losses increase, and torque losses decrease. Flow losses are increasing slower than torque losses are decreasing. Above this temperature region flow losses increase faster than torque losses decrease. This results in an optimal temperature and viscosity region where efficiencies are greatest.

4.5 Uncertainty discussion

Manring [17] describes a method for determining uncertainty in P/M testing, and a summary is shown here along with results of the analysis. The following equation is for overall efficiency. Prime notation indicates the maximum measured value, the subscript "max" indicates the instrument's range, and ξindicates the manufacturer's suggested uncertainty. Manring began with partial derivatives. Those partial derivatives were then

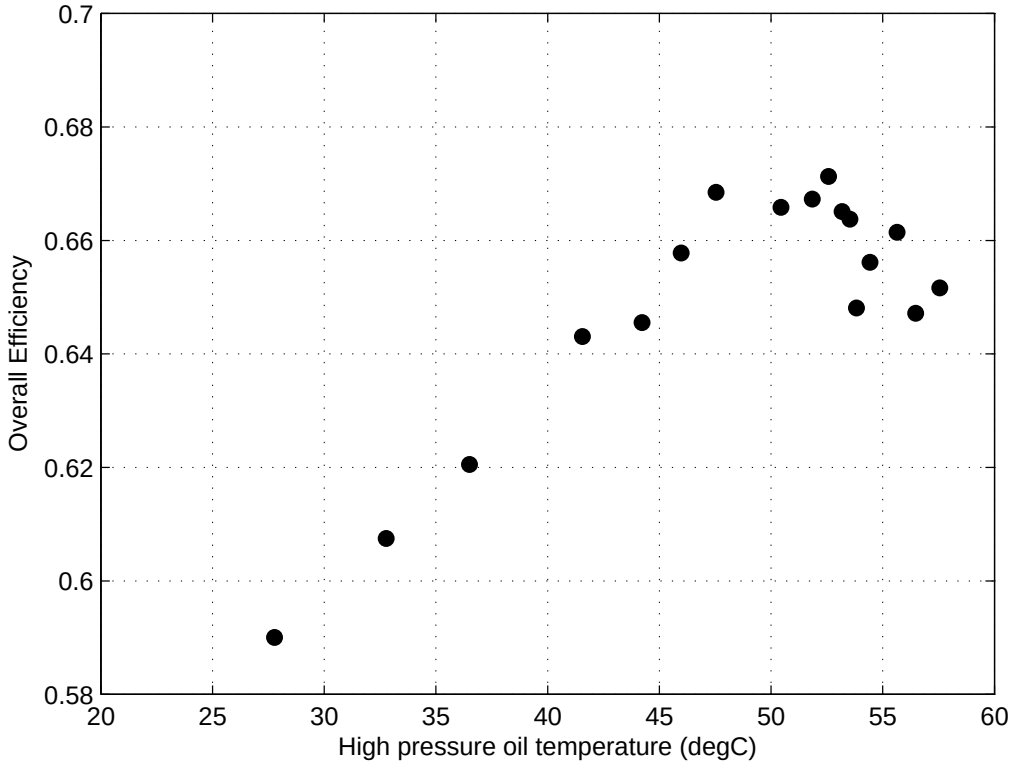


Figure 4.5: Results of temperature effects experiment at 2000 RPM, 0.5 command displacement, 20.7MPa

Measurement	Range "max"	Measured "prime"	Manufacturer accuracy, ξ
Pressure (p)	35 MPa	20 MPa	$\pm$0.25%
Flow (q)	75.7 lpm	85 lpm	$\pm$0.1%
Torque (T)	564.9 N-m	96.9 N-m	$\pm$0.25%
Speed (ω)	10000 RPM	4000 RPM	$\pm$0.01%
Displacement (X)	28 cc/rev	29.2 cc/rev	5% (estimated)

Table 4.2: Values used in uncertainty analysis

simplified to the following three equations.

$$\varepsilon_{overall} = \pm \left\{ \xi_p \frac{p_{max}}{p'} + \xi_q \frac{q_{max}}{q'} + \xi_t \frac{T_{max}}{T'} + \xi_s \frac{\omega_{max}}{\omega'} \right\} \tag{4.14}$$

The fraction amplifies the manufacturer's suggested uncertainty if the full range of the instrument is not used.

Similarly, volumetric and mechanical uncertainty may be calculated as the following:

$$\varepsilon_{volumetric} = \pm \left\{ \xi_q \frac{q_{max}}{q'} + \xi_s \frac{\omega_{max}}{\omega'} + \xi_X \frac{X_{max}}{X'} \right\} \tag{4.15}$$

$$\varepsilon_{mechanical} = \pm \left\{ \xi_p \frac{p_{max}}{p'} + \xi_t \frac{T_{max}}{T'} + \xi_X \frac{X_{max}}{X'} \right\} \tag{4.16}$$

The values for X, the exact displacement, are estimated based on the scatter in the data observed in Section 4.2 and the observation of P/M behavior in general. The third term in both of these equations represents the derived volume error. Table 4.2 shows the values used in this analysis, and the results are presented in table 4.3. Overall efficiency error is lower than volumetric or mechanical efficiency error because the displacement is not needed in the calculation. To reduce the volumetric and mechanical efficiency errors, displacement sensors should be used in the P/M units.

Repeatability was addressed with a small experiment in which data was collected over

$\varepsilon_{overall}$	$\varepsilon_{volumetric}$	$\varepsilon_{mechanical}$
$\pm 2\%$	$\pm 5.3\%$	$\pm 6.9\%$

Table 4.3: Results of uncertainty analysis

three days in Pump CCW mode. At the beginning of each day the same four data points were tested. Oil temperature was in acceptable limits for these tests. Overall efficiencies were tabulated. Mean and standard deviation were calculated. A confidence interval was created to determine the interval in which the actual mean probably exists. Efficiency values appear to be within 0.02 (2%) of the confidence interval. The test points collected along with the results are found in Table 4.4. Values in the "Command Displacement" column are not derived values.

Pressure (ΔP)	Command Displacement	Speed [RPM]	$\eta_{overall}$, Day 1	$\eta_{overall}$, Day 2	$\eta_{overall}$, Day 3	Mean	Standard Dev.	95% Confidence Interval
6.92 MPa	0.2	1000	0.41	0.41	0.43	0.42	0.012	0.42 ± 0.01
6.92 MPa	0.2	2000	0.71	0.70	0.74	0.72	0.021	0.72 ± 0.02
13.8 MPa	1	1000	0.27	0.28	0.31	0.29	0.021	0.29 ± 0.02
13.8 MPa	1	2000	0.31	0.26	0.27	0.28	0.026	0.28 ± 0.03

Table 4.4: Results of repeatability experiment

Chapter 5

Modeling

This chapter explains the model used in this research in detail. Section 5.1 explains the model for an ideal P/M unit with no losses. Section 5.2 details the Dorey [4] model and its formulation. The process for fitting the model to the data is described in Section 5.3. Finally, Section 5.4 explains the improvements made to the Dorey model.

5.1 The Ideal Model

Section 5.1 presents the ideal model.

The ideal flow and torque models containing no loss terms can be described with the following equations:

$$q_{high} = \frac{\omega DX}{1000} \tag{5.1}$$

$$T = \frac{\Delta P \cdot DX}{2\pi} \tag{5.2}$$

Any of the analytical models which exist in the literature begin with these expressions. Torque does not depend on speed or viscosity. The models contain various loss terms which add or subtract from the ideal values above. Flow does not depend on pressure or viscosity.

5.2 The Dorey Model

Section 5.2 presents the Dorey model which is used in this analysis.

The losses in P/M units can be divided into two categories: volumetric (flow) and torque. The volumetric losses may be divided into leakage and compressibility. Leakage is the passage of fluid through the small clearances between moving parts due to a pressure differential. Compressibility losses occur because the hydraulic fluid, oil, is slightly compressible. A packet of fluid changes volume when it is subjected to a pressure change.

Torque losses mainly arise because of friction between translating and rotating components. Dorey divides these sources of friction into two groups, viscous and coulomb. Viscous friction exists between two surfaces moving with respect to each other which have a thin film of fluid between them in a hydrodynamic bearing. This type of friction increases with speed, but does not depend on load. Coulomb, or dry, friction exists between two moving surfaces in contact with one another.

The Dorey model contains terms that account for all four of these specific losses. Each term contains the variables that influence the effect and a coefficient which describes the magnitude of the influence. The coefficient is determined in the model fitting process, described in Section 5.3.

The flow model for a pump is:

$$q^P_{high} = \frac{\omega DX}{1000} - C^*_s \frac{1000 \cdot 60 \cdot \Delta P \cdot D}{2\pi \cdot \mu} - \frac{\Delta P \cdot \omega D}{10^6 \cdot B}\left(V_r + \frac{1+X}{2}\right) \tag{5.3}$$

C^*_s is the leakage coefficient, $\mu$is the fluid viscosity, B is the fluid bulk modulus, and V_r is

$$V_r = \frac{V_{Clearance}}{V_{Swept}} \tag{5.4}$$

V_r is the ratio of clearance (dead) volume to swept volume. The clearance volume of a P/M may be large because each piston has a hole drilled in its center along its complete length. This hole provides a passage for oil for internal lubrication. $\frac{\omega DX}{1000}$ is the ideal flow, as described in Section 5.1. $C^*_s \frac{1000 \cdot 60 \cdot \Delta P \cdot D}{2\pi \cdot \mu}$ is the leakage term, while $\frac{\Delta P \cdot \omega D}{10^6 \cdot B}\left(V_r + \frac{1+X}{2}\right)$ is the compressibility term.

The flow model for a motor is similar to that of the pump, but the minus signs separating the terms are changed to plus signs. This is because volumetric losses in pumps decrease the outlet flow, while volumetric losses in motors increase the inlet flow.

$$q^M_{high} = \frac{\omega DX}{1000} + C^*_s \frac{1000 \cdot 60 \cdot \Delta P \cdot D}{2\pi \cdot \mu} + \frac{\Delta P \cdot \omega D}{10^6 \cdot B}\left(V_r + \frac{1+X}{2}\right) \tag{5.5}$$

The leakage coefficient, C^*_s, can be fixed or variable, depending on the complexity desired. Dorey presents two forms for C^*_s, one for an axial piston P/M, the other for a gear P/M. Both will be compared in section 5.4, but only the expression for the axial piston P/M will be used here. In the piston form, a and b are coefficients determined by least squares fitting, described in section 5.3. For the gear form, a, b, c, and n are coefficients determined by the same method. The star in the equations above and below

denote a value that is specific to one operating condition. For example, C_s is one value for all of the conditions, but C_s^* is a specific value for one condition.

$$C^*_{s,piston} = C_s(\frac{\Delta P}{P_{atmospheric}})(a + b\frac{\omega}{\omega_{max}}) \tag{5.6}$$

$$C^*_{s,gear} = C_s(\frac{\Delta P}{P_{atmospheric}})^n[a + b\frac{\omega}{\omega_{max}} + c(\frac{\omega}{\omega_{max}})^2] \tag{5.7}$$

In both cases, C_s^* increases with speed and pressure. The piston model has a linear speed relationship, while the gear model has a quadratic speed relationship. The piston model has a linear pressure relationship. The gear model has an exponent, n, which allows the nature of the relationship to be determined by the model fitting procedure. When the expressions for C_s^* are inserted into the flow equations, the piston model gains a second degree pressure dependence, which may be undesirable. The gear model's exponent eliminates this problem. C_s is determined by the following equation when evaluated at maximum pressure and speed per Dorey's instructions:

$$C_s = (\frac{dQ}{dP})(\frac{\mu}{D})(60 \cdot 2\pi) = 1.046 \times 10^{-8} \tag{5.8}$$

This is obtained by plotting flow vs. pressure for that operating condition. The derivative is evaluated numerically at the highest pressure.

The torque models for pumps and motors are

$$T^P = \frac{\Delta P \cdot DX}{2\pi} + \frac{C_v^* \mu \omega D}{60 \cdot 10^6} + \frac{C_f^* \cdot \Delta P \cdot D}{2\pi} \tag{5.9}$$

$$T^M = \frac{\Delta P \cdot DX}{2\pi} - \frac{C_v^* \mu \omega D}{60 \cdot 10^6} - \frac{C_f^* \cdot \Delta P \cdot D}{2\pi} \tag{5.10}$$

Torque losses in motors decrease the torque at the shaft, while losses in pumps increase the shaft torque. $\frac{C_v^* \mu \omega D}{60 \cdot 10^6}$ is the viscous friction term, and $\frac{C_f^* \cdot \Delta P \cdot D}{2\pi}$ is the Coulomb friction term. $C_v^*$and C_f^* are the viscous and Coulomb friction terms, respectively. Similar to C_s^*, they may be fixed coefficients but are more useful if they are variable. Again, Dorey presents different forms for the gear and piston models:

$$C_{v,piston}^* = C_v(a + bX) \tag{5.11}$$

$$C_{v,gear}^* = C_v \tag{5.12}$$

C_v is determined by the following when evaluated at maximum pressure, and full displacement, per Dorey's instructions. It is obtained by plotting torque vs. speed. The derivative is calculated numerically.

$$C_v = (60 \cdot 10^6)(\frac{dT}{d\omega})(\frac{1}{\mu D}) = 2.163 \times 10^5 \tag{5.13}$$

Similarly, for C_f^*,

$$C_{f,piston}^* = C_f[a + b\frac{\omega}{\omega_{max}} + c(\frac{\omega}{\omega_{max}})^2](d + eX) \tag{5.14}$$

$$C_{f,gear}^* = C_f(\frac{\Delta P}{P_{atmospheric}})^n[a + b\frac{\omega}{\omega_{max}} + c(\frac{\omega}{\omega_{max}})^2] \tag{5.15}$$

Both versions of C_f^* have quadratic speed dependencies. The version for the piston model also has a displacement dependence. The version for the piston model also has a pressure dependence with an exponent. C_f is determined by the following when evaluated at approximately 2400 RPM, full displacement, and maximum pressure, per Dorey's instruc-

tions. Again, it is obtained by plotting torque vs. pressure. The derivative is calculated numerically.

$$C_f = 2\pi(\frac{dT}{dP})(\frac{1}{D}) = 0.9702 \tag{5.16}$$

5.3 Model Fitting Method

The method of least squares was used to fit the model to the data. The MATLAB function "lsqcurvefit" was used to fit all of the coefficients. It uses the "fminsearch" search algorithm but adds the least squares mathematics automatically. This method searches the solution space for the optimal set of coefficients. The optimum set minimizes the sum of squares total (SST) between the data points and the predicted values using the model and a set of coefficients. The MATLAB code listing may be found in Appendix C.

Two script files are used to perform the analysis. The first gathers and organizes the appropriate data for passage to the second, the function file, which contains the equations of the model. The first file also initializes the search algorithm by setting bounds on the initial guess and the valid range for each coefficient. The initial bounds were determined with some hand calculations to determine appropriate values which would allow the model to work. These calculations were corroborated with model runs where the bounds were set wide, at 1e6.

With the exception of the initial guesses and valid range for the coefficients, default values for lsqcurvefit were used in the analysis. The search algorithm worked best when the unit of flow rate was changed to cc/sec. This kept the magnitude of the flow around one which allowed the search algorithm to run with the fewest number of errors. The torque unit remained N-m.

The least squares fit minimized the sum of the squared error between the measured q_{high} or T and the predicted q_{high} or T generated by the model being tested.

ΔP data is used for the variable P, high pressure, in the models. The S42 P/M's require the low pressure port to be greater than atmospheric pressure. ΔP data accounts for this change.

Each model was run 100 times with random initial guesses to check for local minima. The lsqcurvefit function returned a sum of squares total (SST) value for each run. The run with the minimum SST value was selected as the coefficient set to use for the model. The efficiency and loss behavior of the four P/M modes was inconsistent. This necessitated the use of a separate set of coefficients for each of the four modes. This allowed the lsqcurvefit function to find the optimal set of coefficients for each mode.

Manufacturer data for V_r was not available. Initially we allowed V_r to be another variable. However, the optimization produced physically unlikely values for V_r. Values for V_r must lie between 0 and 1 to be physically meaningful. A more practical upper boundary for V_r would be 0.2, as the clearance volume would definitely be much smaller than the swept volume. The acceptable bounds for V_r were set at 0 to 0.2. This restriction made only a very small change to the SST values. For the GM3 flow model (see Section 5.4) V_r was constrained to be 0.1 as an experiment. This had no impact on the results because the coefficient, e, could offset the change in V_r.

Coefficients representing constants at zero speed or displacement were treated differently. This includes the coefficients a, g, l, r, and u_1. They were constrained to be greater than zero. This constraint ensures that the loss terms cannot be negative at zero speed or displacement. It was also used for the Dorey model improvements.

The method detailed in this section is used not only for the model equations from

Constant	Value
ω_{max}	4000 RPM
B	1.02 GPa
P_{atmos}	0.1 MPa

Table 5.1: Values of constants used in the model fitting.

Dorey, but also for the modifications to the model as explained in Section 5.4. The values of constants used in the model fitting are shown in Table 5.1. A constant value for the bulk modulus was used for this analysis. Bulk modulus is known to vary with pressure. This is because air is entrained in the oil. The effect is highly dependent on pressure. At low pressures, the air reduces the effective bulk modulus. At high pressures the air dissolves into the oil. The effective bulk modulus at high pressures is close to that of air-free oil [8]. Modeling this pressure dependence requires several assumptions of empirical values. These assumptions could introduce more error into the model than using a constant value.

The manufacturer supplied an air-free bulk modulus of 1.7 GPa. Yu [8] states that with an entrained air content of 1%, the effective bulk modulus is reduced by 25%. Merritt [19] explains that reliable results may be obtained with a reduction of 50% from the air-free value. These sources suggest that the bulk modulus should be reduced by 25-50%. A reduction of 40% from the air-free value was used in this analysis. This value was used for all flow model calculations.

A linear viscosity model was constructed to allow for changes in viscosity with changes in temperature. The manufacturer supplied two viscosity data points at two temperatures for the Mobil DTE 25 hydraulic oil, 40ºC and 100ºC. Viscosity was linearized between those two points. Viscosity, μ, is in Pa-s, and temperature, K, is in Celsius. The resulting equation is:

$$\mu = -0.000538K + 0.05949 \tag{5.17}$$

5.4 Improvements to the Dorey Model

This section explains several improvements to the Dorey model and how they were developed. An exhaustive set of graphs may be found in Chapter 6.

The first model used the Dorey piston pump models for both volumetric and torque losses. This combination fit the data satisfactorily, but there was significant room for improvement. Examples of this are contained in Sections 6.3 and 6.4.

The next step was to try the Dorey gear torque and flow models. In the discussion regarding model improvements, any flow model can be used with any torque model. The Dorey gear flow model is more complex than the piston model. It is of higher order than the piston flow model as it contains a second degree speed term. In implementation the second degree speed term caused more problems than it solved. It fit the peculiarities in the data set rather than following overall trends. In contrast the Dorey gear torque model is simpler than the piston torque model. The fit for this torque model was poor. In summary, the Dorey gear models led to degeneration, rather than improvement, of the fit.

Refinements of the flow model concentrated on taking the best of both the piston and gear flow models. This resulted in three additional models, labeled here as Grandall Modified 1, 2, 3, and 4 (GM1-4). Grandall Modified 1 sought to retain the best of both the piston and gear flow models. The first order speed relationship from the piston model was retained. The parameter n, the exponent of the pressure term, was retained from the gear model to provide flexibility in determining the pressure relationship of the slip coefficient, C_s. Most values for n were close to zero. This resulted in $(\frac{\Delta P}{P_{atmos}})^n$ equal to one, leaving the slip coefficient with a linear pressure relationship. A coefficient, d, was also placed on the final, compressibility, term. This was to add some flexibility to affect the entire compressibility term, not just V_r.

Grandall Modified 2 is the same as Grandall Modified 1 except that the coefficient on the compressibility term was removed. GM3 adds a separate coefficient (e,f) for each term in the compressibility portion of the equation. For GM3 a,b, and e were constrained to be greater than zero to minimize odd behavior by the models. GM4 uses a linear polynomial pressure relationship for C_s instead of $(\frac{\Delta P}{P_{atmos}})^n$. This simpler polynomial version was tested because the exponential version suggested a linear pressure relationship.

The equations for the Dorey piston and gear models along with Grandall Modified 1, 2, 3, and 4 are shown below. The expressions for C_s^* have been inserted into these equations.

$$q^P_{Dorey\,Piston} = \frac{\omega DX}{1000} - C_s(\frac{\Delta P}{P_{atmos}})(a + b(\frac{\omega}{\omega_{max}}))\frac{1000 \cdot 60 \cdot \Delta P \cdot D}{2\pi\mu} - \frac{\Delta P \cdot \omega D}{10^6 B}(V_r + \frac{1+X}{2}) \quad (5.18)$$

$$q^P_{Dorey\,Gear} = \frac{\omega DX}{1000} - C_s(\frac{\Delta P}{P_{atmos}})^n(a + b(\frac{\omega}{\omega_{max}}) + c(\frac{\omega}{\omega_{max}})^2)\frac{1000 \cdot 60 \cdot \Delta P \cdot D}{2\pi\mu} - \frac{\Delta P \cdot \omega D}{10^6 B}(V_r + \frac{1+X}{2}) \quad (5.19)$$

$$q^P_{Grandall\,1} = \frac{\omega DX}{1000} - C_s(\frac{\Delta P}{P_{atmos}})^n(a + b(\frac{\omega}{\omega_{max}}))\frac{1000 \cdot 60 \cdot \Delta P \cdot D}{2\pi\mu} - d\frac{\Delta P \cdot \omega D}{10^6 B}(V_r + \frac{1+X}{2}) \quad (5.20)$$

$$q^P_{Grandall2} = \frac{\omega DX}{1000} - C_s(\frac{\Delta P}{P_{atmos}})^n(a + b(\frac{\omega}{\omega_{max}}))\frac{1000 \cdot 60 \cdot \Delta P \cdot D}{2\pi\mu} - \frac{\Delta P \cdot \omega D}{10^6 B}(V_r + \frac{1+X}{2}) \quad (5.21)$$

$$q^P_{Grandall3} = \frac{\omega DX}{1000} - C_s(\frac{\Delta P}{P_{atmos}})^n(a + b(\frac{\omega}{\omega_{max}}))\frac{1000 \cdot 60 \cdot \Delta P \cdot D}{2\pi\mu} - \frac{\Delta P \cdot \omega D}{10^6 B}(eV_r + f\frac{1+X}{2}) \quad (5.22)$$

$$q^P_{Grandall4} = \frac{\omega DX}{1000} - C_s(u_1 + u_2\frac{\Delta P}{P_{atmos}})(a + b(\frac{\omega}{\omega_{max}}))\frac{1000 \cdot 60 \cdot \Delta P \cdot D}{2\pi\mu} - \frac{\Delta P \cdot \omega D}{10^6 B}(eV_r + f\frac{1+X}{2}) \quad (5.23)$$

The torque model refinement was similar to the flow model refinement. The best parts of the Dorey piston and gear models were combined. The piston model contains a linear displacement dependence in the viscous term. The Coulomb friction term includes a second order speed term with a linear displacement dependence. The gear model contains no displacement dependence in the viscous term. The coulomb friction term includes the same second order speed factor, but removes the linear displacement dependence. A nonlinear pressure factor is also added. The second degree speed factors in both the Dorey piston and gear models appeared to be causing unwanted nonlinear behavior and were simply not needed. The torque loss data showed only a slight speed dependence.

Again, two new torque models were created, GM1 & GM2. Both GM1 and GM2 use only a linear speed dependence in the coulomb friction term. They also contain a linear displacement dependence in the viscous term. This allows the model to fit the data better,

particularly in motor mode. GM2 also contains a nonlinear pressure term and a linear displacement term in the coulomb friction term. The equations for the Dorey piston and gear models along with GM1 & GM2 are shown below. Expressions for C_v^* and C_f^* have already been substituted.

$$T_{Dorey\,Pist} = \frac{\Delta P \cdot DX}{2\pi} + C_v(l+mX)\frac{\mu\omega D}{60\cdot 10^6} + C_f(g+h(\frac{\omega}{\omega_{max}})+k(\frac{\omega}{\omega_{max}})^2)(r+sX)\frac{\Delta P \cdot D}{2\pi} \tag{5.24}$$

$$T_{Dorey\,Gear} = \frac{\Delta P \cdot DX}{2\pi} + C_v\frac{\mu\omega D}{60\cdot 10^6} + C_f(\frac{\Delta P}{P_{atmos}})^n(g+h(\frac{\omega}{\omega_{max}})+k(\frac{\omega}{\omega_{max}})^2)\frac{\Delta P \cdot D}{2\pi} \tag{5.25}$$

$$T_{Grandall\,1} = \frac{\Delta P \cdot DX}{2\pi} + C_v(l+mX)\frac{\mu\omega D}{60\cdot 10^6} + C_f(g+h(\frac{\omega}{\omega_{max}}))\frac{\Delta P \cdot D}{2\pi} \tag{5.26}$$

$$T_{Grandall\,2} = \frac{\Delta P \cdot DX}{2\pi} + C_v(l+mX)\frac{\mu\omega D}{60\cdot 10^6} + C_f(\frac{\Delta P}{P_{atmos}})^n(g+h(\frac{\omega}{\omega_{max}}))(r+sX)\frac{\Delta P \cdot D}{2\pi} \tag{5.27}$$

As shown in Section 6.1, using GM3 for the flow model and GM1 for the torque model produced the best fit for the Sauer-Danfoss Series 42 28cc/rev P/M unit tested here.

Chapter 6

Modeling Results

Chapter 6 presents the results of the modeling found in Chapter 5 in the form of graphs and tables. The Chapter begins by showing the results of the best models in the form of efficiency graphs in Section 6.1. Some commentary on the results follows. The remaining parts of Chapter 6 detail the results of all of the flow and torque models. The method for selecting the best flow and torque models is also included. Section 6.2 includes a table listing all of the coefficients for each model for each mode. Section 6.3 includes graphs of the flow losses and volumetric efficiency. Torque losses and mechanical efficiencies are shown in Section 6.4. Section 6.5 explains the method for choosing the best flow and torque model. Section 6.6 shows three dimensional surface plots of loss vs. displacement and speed.

6.1 Overall Efficiency Results and Commentary

This section presents the overall efficiency results in graphical form. The best flow model was GM3, while the best torque model was GM1. The method used to select these models

is explained later in Section 6.5. The GM3 flow and GM1 torque models are used to construct these overall efficiency graphs. The actual measured values are presented on the graph as well. Figure 6.1 shows the graphs for pump modes, while Figure 6.2 shows the graphs for motor modes.

The overall efficiency graphs in Figures 6.1 and 6.2 would not be used directly in a simulation involving a hydraulic hybrid vehicle, or any other application using these P/M's. The losses are based on physical phenomena. The physical phenomena creating those losses must be modeled, not efficiency. The efficiency graphs are a convenient way to visualize the operating conditions where significant energy is being wasted. Operating conditions with low efficiency should be avoided. Conditions with high efficiency should be used as often as possible. Regions with zero efficiency should be avoided as much as possible, as no energy from the input is transferred to the output.

The method used to create the overall efficiency graphs in Figures 6.1 and 6.2 involves some calculation. The simple relation of $\eta_{volumetric} \cdot \eta_{mechanical} = \eta_{overall}$ is correct. But as seen in Figure 6.10, the torque efficiency graphs in motor mode have problems which will be explained in Section 6.4. A different method of calculating overall efficiency is used which avoids the motor mode torque efficiency problems. The overall efficiency graphs are calculated using the flow and torque losses. For example, if a motor received 100 W of power from the high pressure line, then lost 10 W to flow losses and 10 W to torque losses, the power output to the shaft would be 80 W. Dividing output power by the input power, the motor had an overall efficiency of 80%.

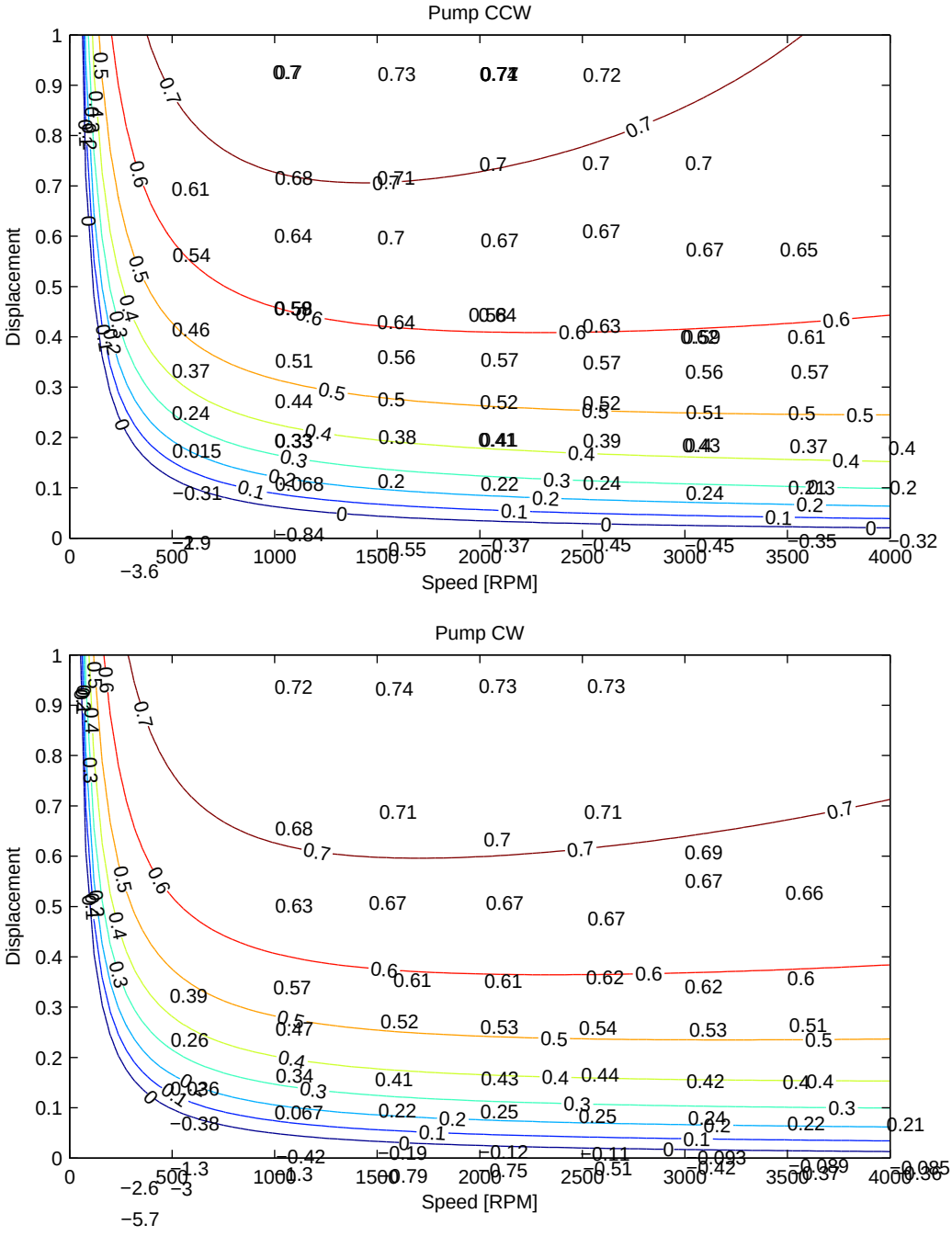


Figure 6.1: Pump overall efficiency results, 13.8 MPa (2000 psi) ΔP

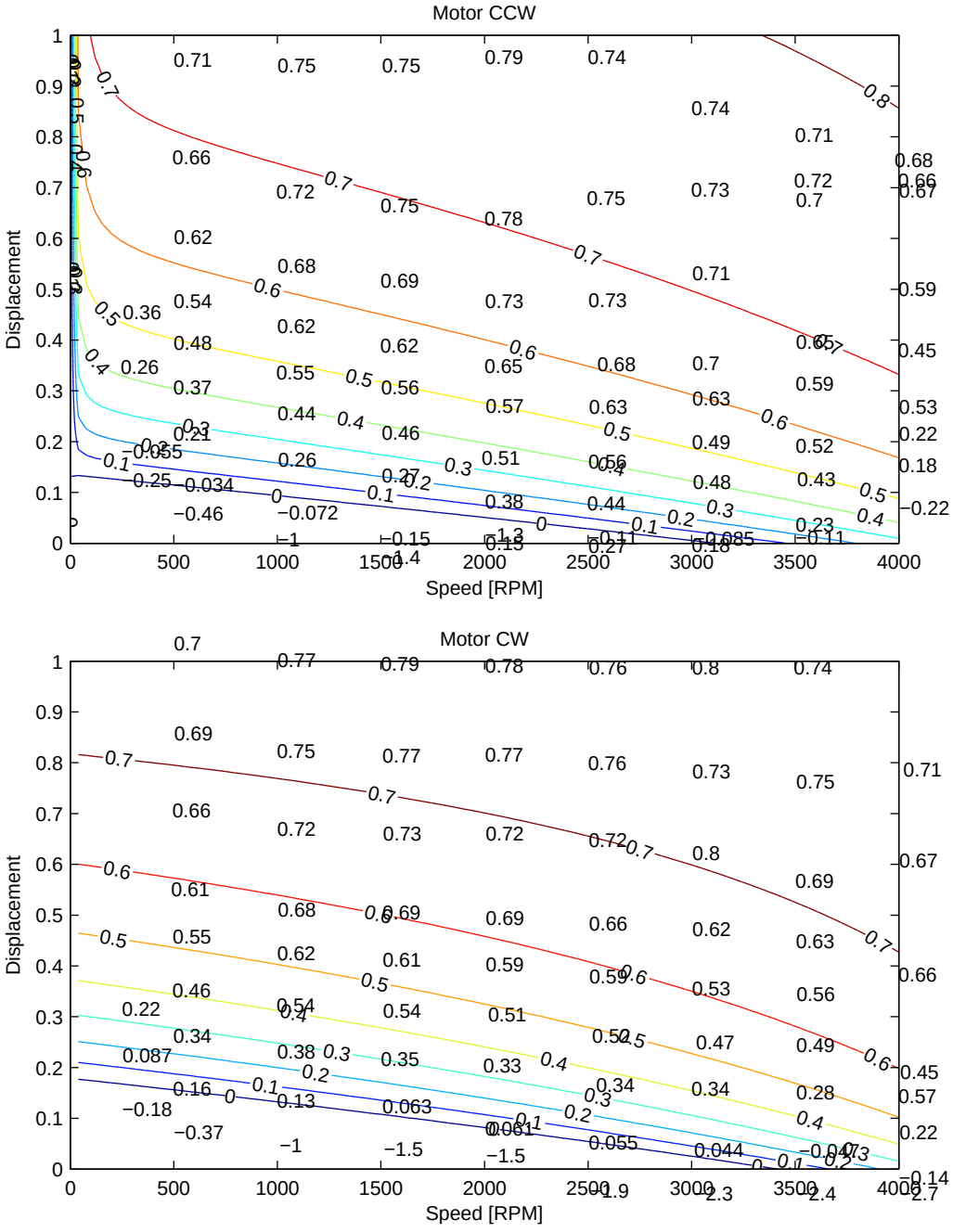


Figure 6.2: Motor overall efficiency results, 13.8 MPa (2000 psi) ΔP

6.2 Coefficient Results

This section lists the coefficients resulting from fitting the models described in Chapter 5 using the least squares fitting procedure described in Section 5.3.

Tables 6.1 and 6.2 show the coefficients for each model and mode. The equations for all of the models may be found at the end of Section 5.4. Table 6.1 contains the coefficients for the flow model, while Table 6.2 contains the coefficients for the torque model. Some models do not make use of all of the coefficients listed in the table. Those coefficients are left blank. "Piston" and "Gear" in the model column refer to the Dorey piston and gear models. The units of flow rate in this table are cc/sec.

Of particular note are the values of "h" in the torque tables. All values are negative. This indicates that the Coulomb friction torque losses decrease with increasing speed. This does not necessarily mean that torque losses in general decrease with increasing speed.

6.3 Flow Loss & Volumetric Efficiency Results

This section presents the flow loss and volumetric efficiency results in graphical form.

Figures 6.3 through 6.6 contain several graphs showing the flow loss modeling results and corresponding volumetric efficiency. The volumetric efficiency graph using the GM3 flow model at the lower left shows both the modeling results and the measured volumetric efficiency. One mode is shown per page. A legend for the four loss graphs is shown in the lower right corner of each page. GM3 was determined to be the best flow model, and the method for choosing the best model is found in Section 6.5. All plots have been generated at a single ΔP of 13.8 MPa (2000 psi). This pressure was chosen because it

Mode	Model	a	b	c	d	e	f	n	u_1	u_2	V_r	SST
Pump CCW	Piston	0.143	-0.028								0.2	9.11×10^4
	Gear	1.796	-2.166	2.23				-0.038			0	8.13×10^4
	GM1	1.527	0.056		0.847			-0.068			0	8.25×10^4
	GM2	1.503	0					-0.074			0	8.25×10^4
	GM3	1.527	0.057			0	1.412	-0.068			0.1	8.25×10^4
	GM4	1.291	0.047			0	0.840		1.121	-0.011	0.1	8.25×10^4
Pump CW	Piston	.1528	-0.178								0.2	1.72×10^5
	Gear	3.347	-0.463	-3.14				-0.493			0	1.62×10^5
	GM1	0.725	-3.946		6.56			0.266			0.2	1.52×10^5
	GM2	3.483	0					-0.689			0	1.67×10^5
	GM3	3.278	0			0	0.184	-0.507			0.1	1.66×10^5
	GM4	7.834	0			0	0.031		0.228	-0.0093	0.1	1.68×10^5
Motor CCW	Piston	0	1.563								0.2	1.64×10^6
	Gear	0	20.99	-6.16				-0.080			0	1.18×10^6
	GM1	0.192	18.06		-1.45			-0.070			0	1.23×10^6
	GM2	0.109	15.34					-0.062			0.2	1.21×10^6
	GM3	0.146	9.788			6.01	4.01	0.074			0.1	1.22×10^6
	GM4	0.093	7.239			31.9	-1.47		1.90	-0.0356	0.1	1.21×10^6
Motor CW	Piston	0	1.178								0.2	1.35×10^6
	Gear	1.380	-6.898	11.1				0.497			0.2	6.40×10^5
	GM1	0	39.69		-27.9			0.015			0.2	5.61×10^5
	GM2	0	5.886					0.259			0.2	1.26×10^6
	GM3	0	10.03	9.94		5.96	-10.1	0.204			0.1	9.64×10^5
	GM4	0	22.08			0	-26.7		1.47	0.0034	0.1	5.86×10^5

Table 6.1: Flow model coefficient results

Mode	Model	g	h	k	l	m	r	s	n	SST
Pump CCW	Piston	0.718	-1.662	0.560	1.269	1.344	0.233	0.187		948
	Gear	0.664	-0.449	-0.280					-0.63	1670
	GM1	0.182	-0.331		1.294	1.316				1170
	GM2	0.924	-0.515		0.505	0.399	0.679	0.574	-0.710	6047
Pump CW	Piston	0.417	-0.756	0.346	0.989	0.606	0.461	0.185		769
	Gear	0.445	-0.524	0.072					-0.413	971
	GM1	0.178	-0.223		1.012	0.743				872
	GM2	0.976	-6.367		4.14	1.686	1.473	0.281	-1.176	1269
Motor CCW	Piston	0.403	-1.841	1.055	1.491	0.459	0.673	-0.010		3070
	Gear	0.217	-0.878	0.574					0.113	4134
	GM1	0.139	-0.522		1.556	0.442				5229
	GM2	1.713	-9.092		2.188	1.332	0.199	0.089	-0.511	4798
Motor CW	Piston	0.517	-1.284	-0.007	1.107	0.876	0.356	-0.004		1180
	Gear	0.229	-0.354	-0.151					-0.129	2471
	GM1	0.184	-0.459		1.103	0.883				1176
	GM2	0.611	-1.702		1.331	0.838	0.559	-0.041	-0.301	1106

Table 6.2: Torque model coefficient results

is in the middle of the range tested. Four of the five graphs on each page have constant speed or constant displacement, indicated by the title of each graph. These constant values are near the extremes of the range tested to show the behavior of the models at multiple operating points. Figures 6.5 and 6.6 show an interesting speed behavior. The flow loss graphs suggest that flow loss is heavily dependent on speed. But the volumetric efficiency graphs suggest no speed dependence at all. This is because the losses are rising as fast as the speed and power are rising.

6.4 Torque Loss & Mechanical Efficiency Results

This section presents the torque loss and mechanical efficiency model results in graphical form.

Figures 6.7 through 6.10 contain several graphs showing the torque loss modeling results. Graphs of mechanical efficiencies using the GM1 model may be found at the bottom of each page. The method for choosing GM1 as the best torque model is described in Section 6.5. One mode is shown per page. A legend for the four loss graphs is shown in the lower right corner of each page.

The anomalies seen in mechanical efficiency for motor mode are a result of faulty derived displacement analysis. The lines converge around 3000 RPM at zero displacement, where the models begin to predict negative losses at low displacements.

6.5 Model Selection Method

The best torque and flow methods are chosen by examining the statistics of the fitting process along with intuition. Tables 6.1 and 6.2 are examined which contain the SST

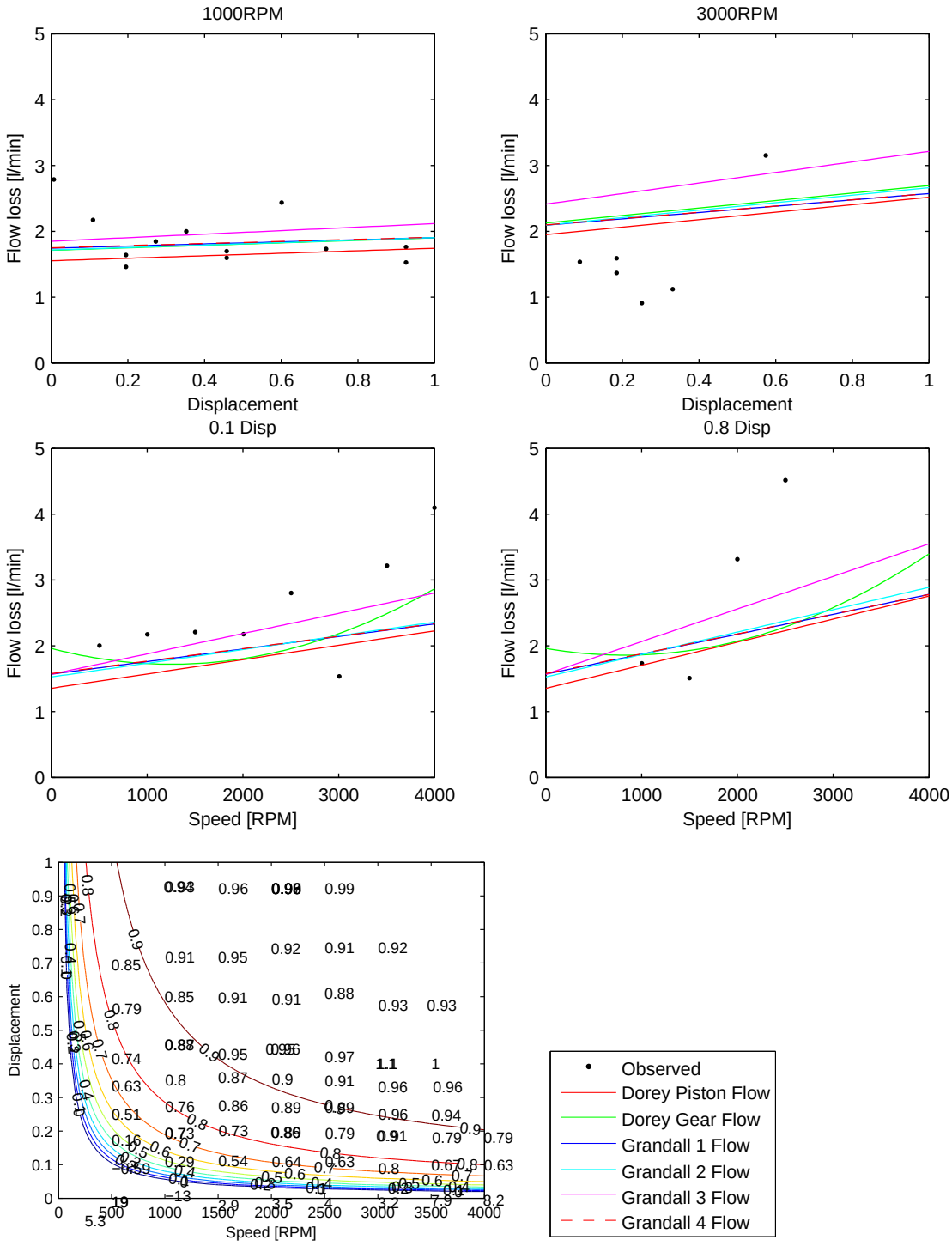


Figure 6.3: Pump CCW flow model results, 13.8 MPa (2000 psi) ΔP. Volumetric efficiency graph utilizes the GM3 flow model.

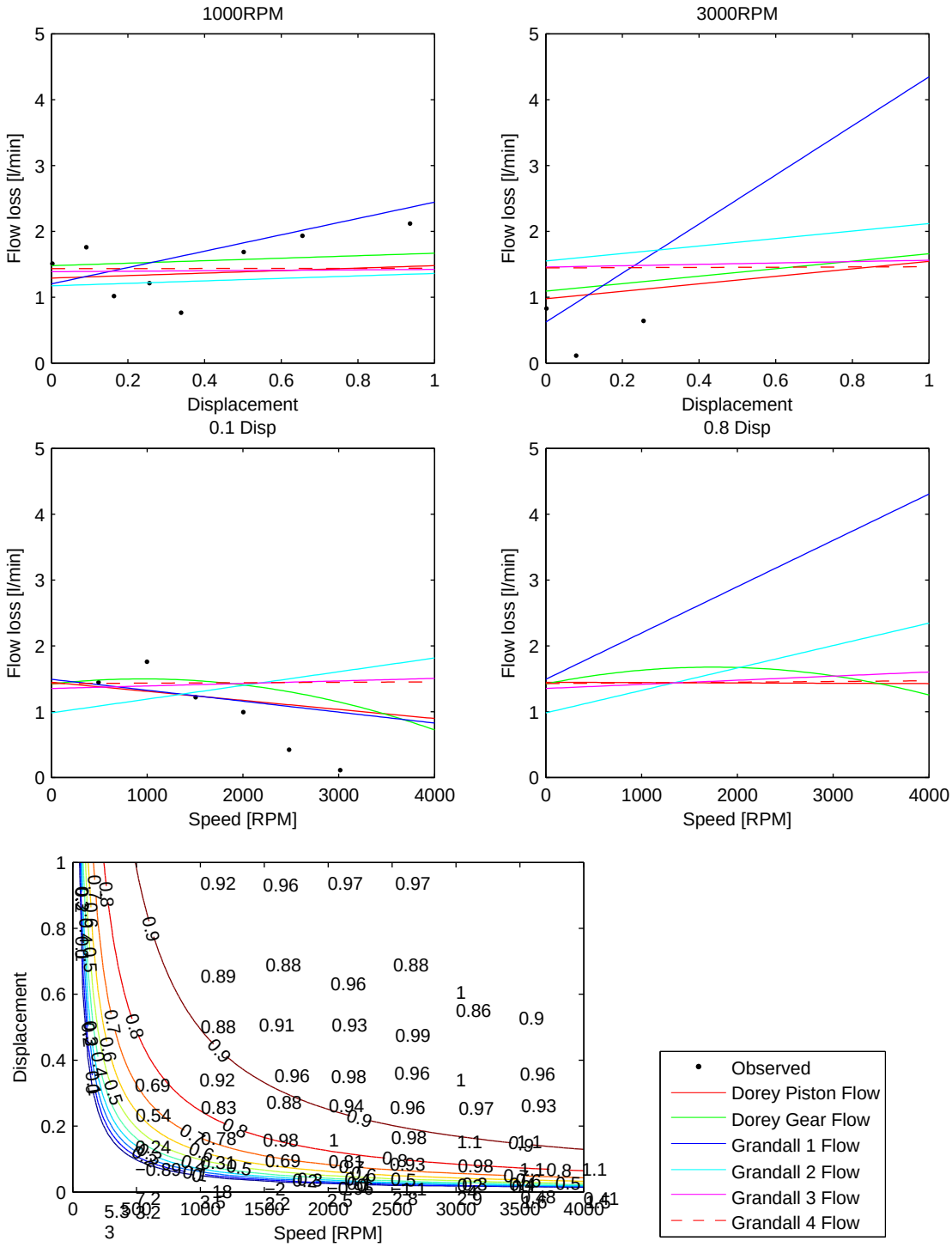


Figure 6.4: Pump CW flow model results, 13.8 MPa (2000 psi) ΔP. Volumetric efficiency graph utilizes the GM3 flow model.

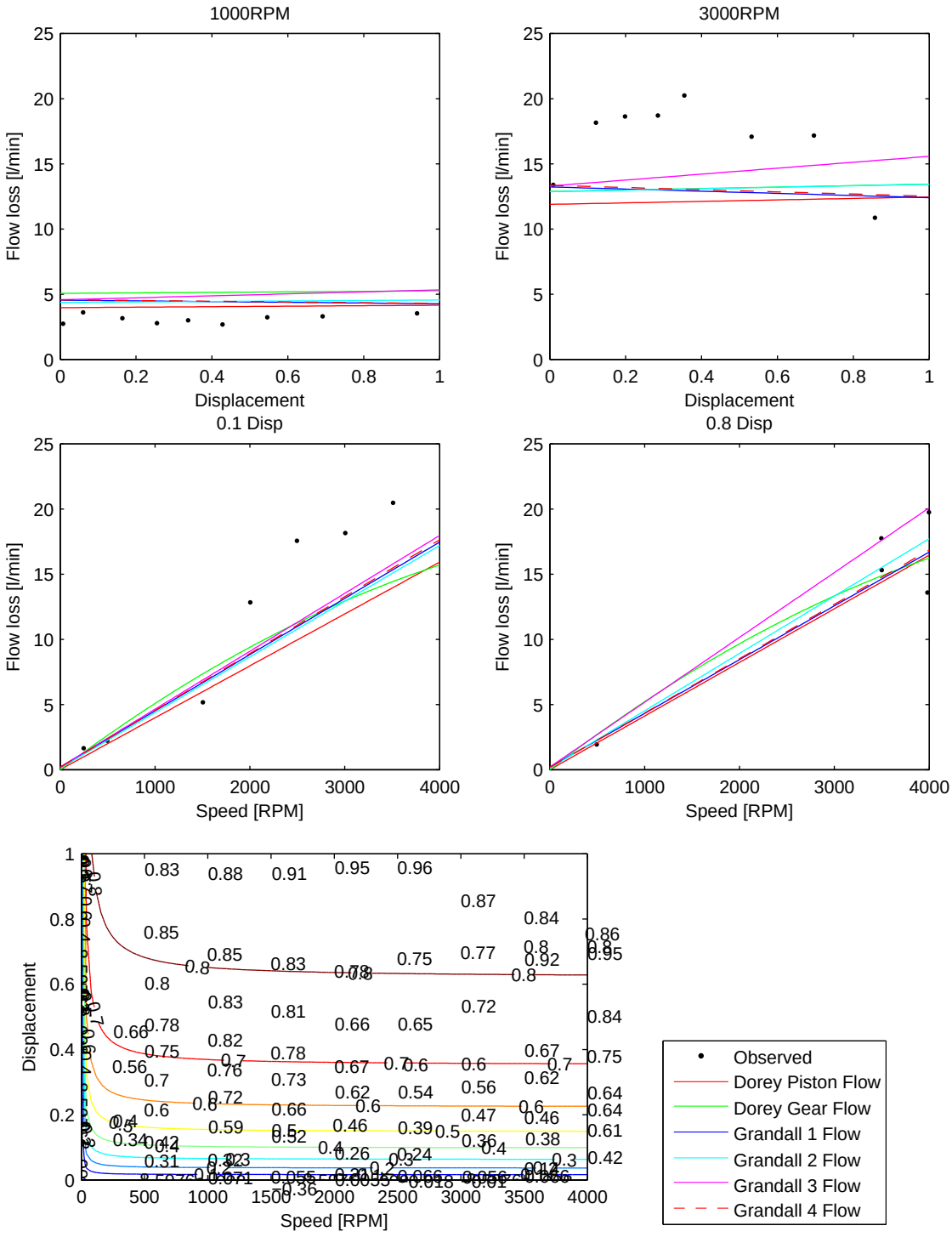


Figure 6.5: Motor CCW flow model results, 13.8 MPa (2000 psi) ΔP. Volumetric efficiency graph utilizes the GM3 flow model.

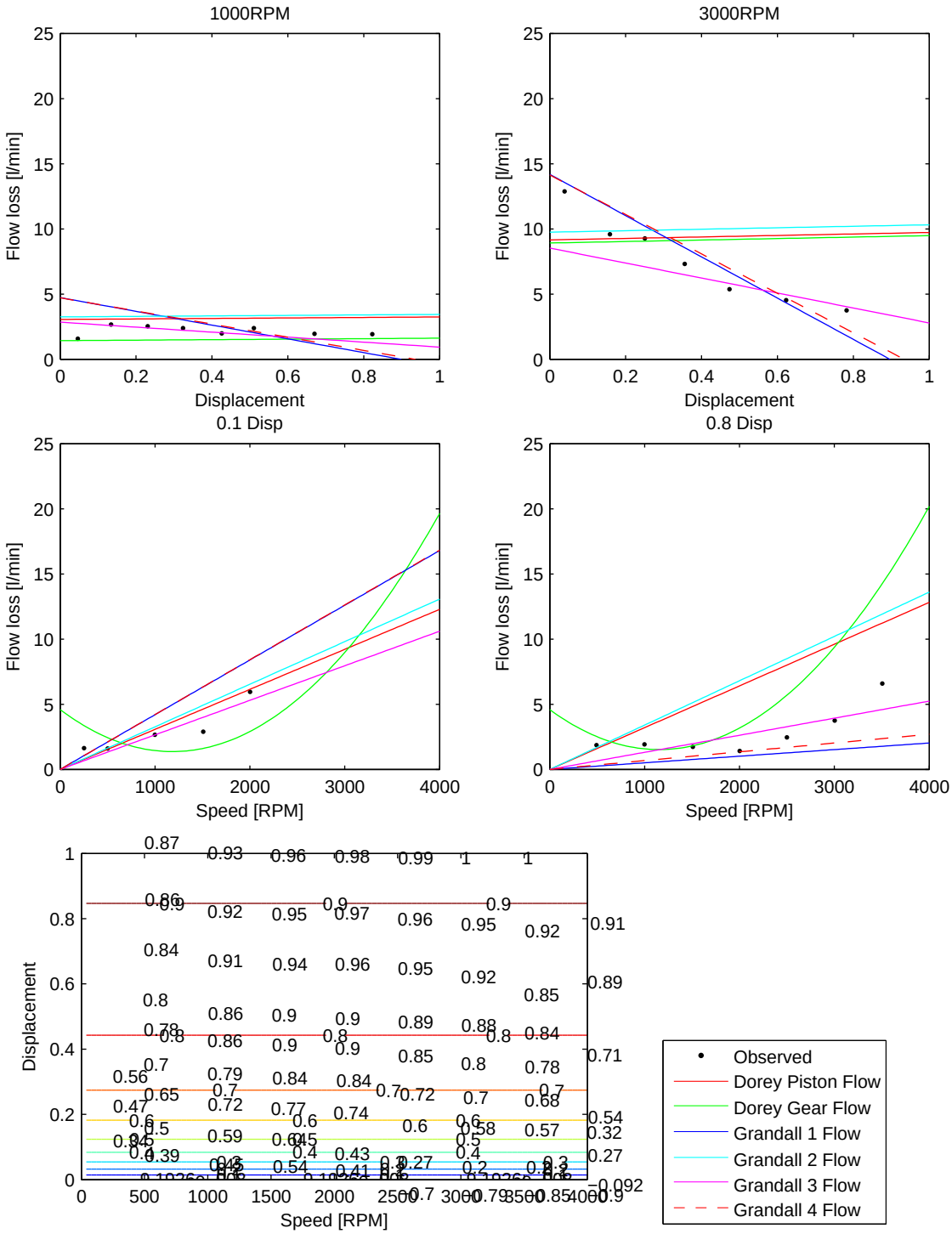


Figure 6.6: Motor CW flow model results, 13.8 MPa (2000psi) ΔP. Volumetric efficiency graph utilizes the GM3 flow model.

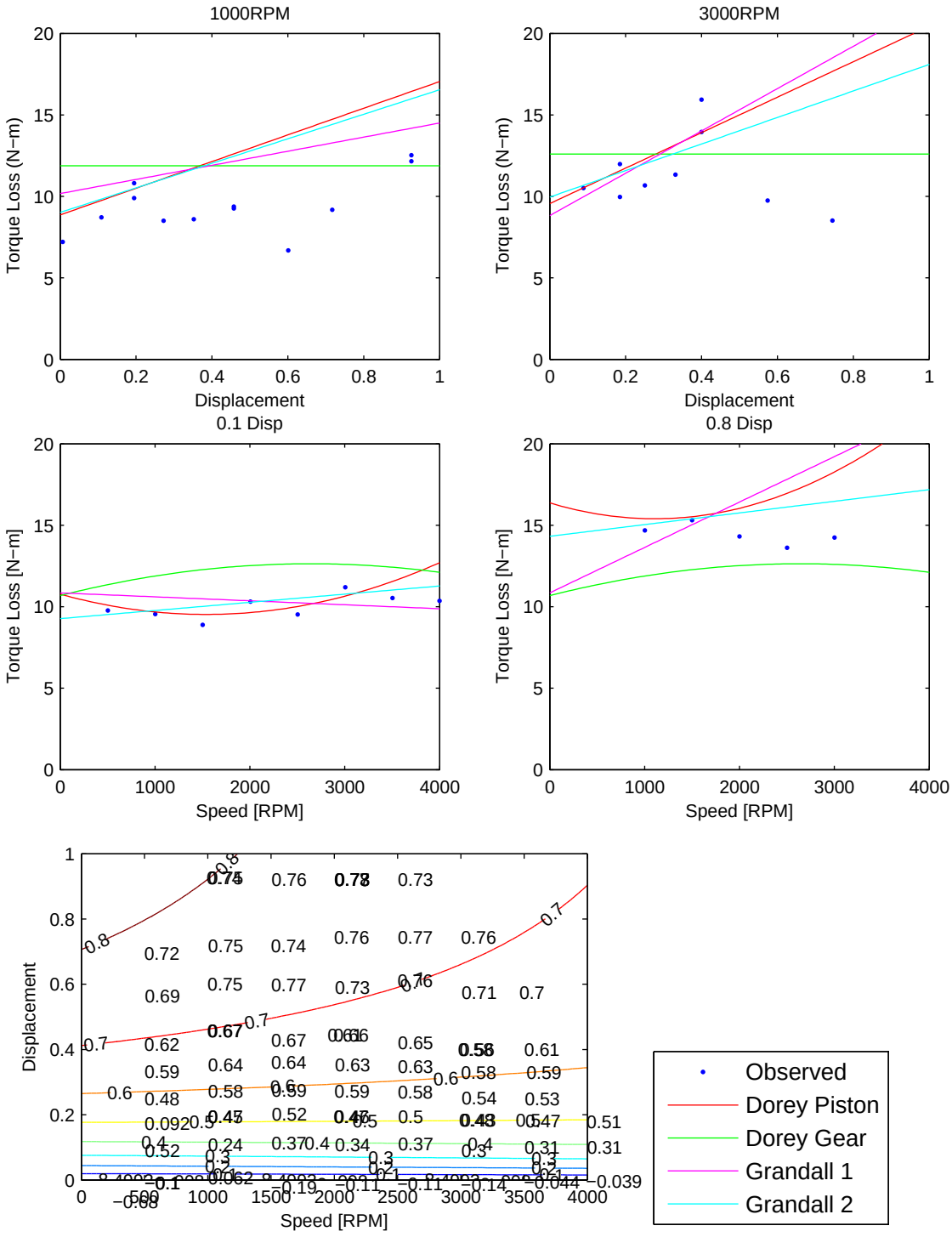


Figure 6.7: Pump CCW torque model results, 13.8 MPa (2000 psi) ΔP. Mechanical efficiency graph utilizes the GM1 torque model.

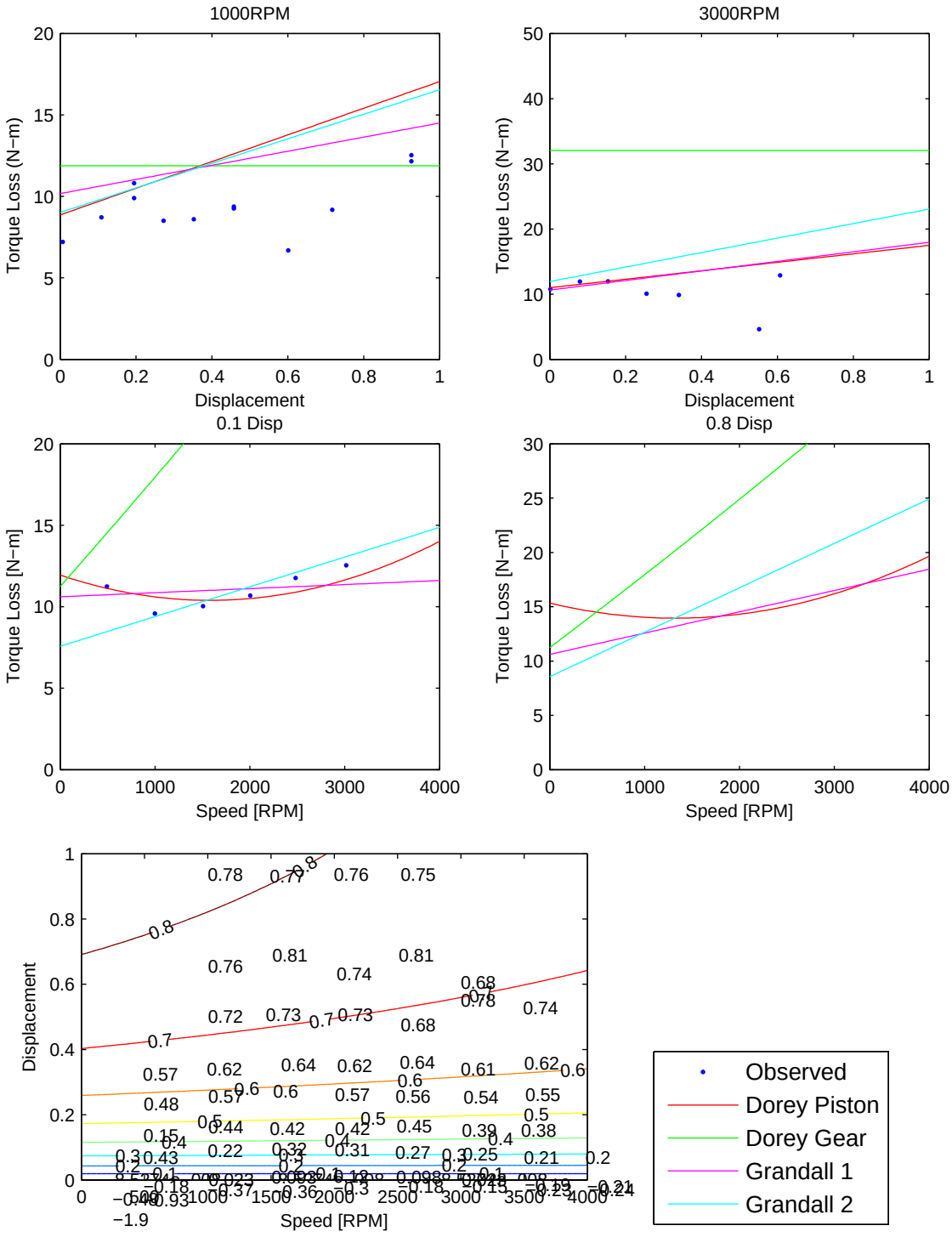


Figure 6.8: Pump CW torque model results, 13.8 MPa (2000 psi) ΔP. Mechanical efficiency graph utilizes the GM1 torque model.

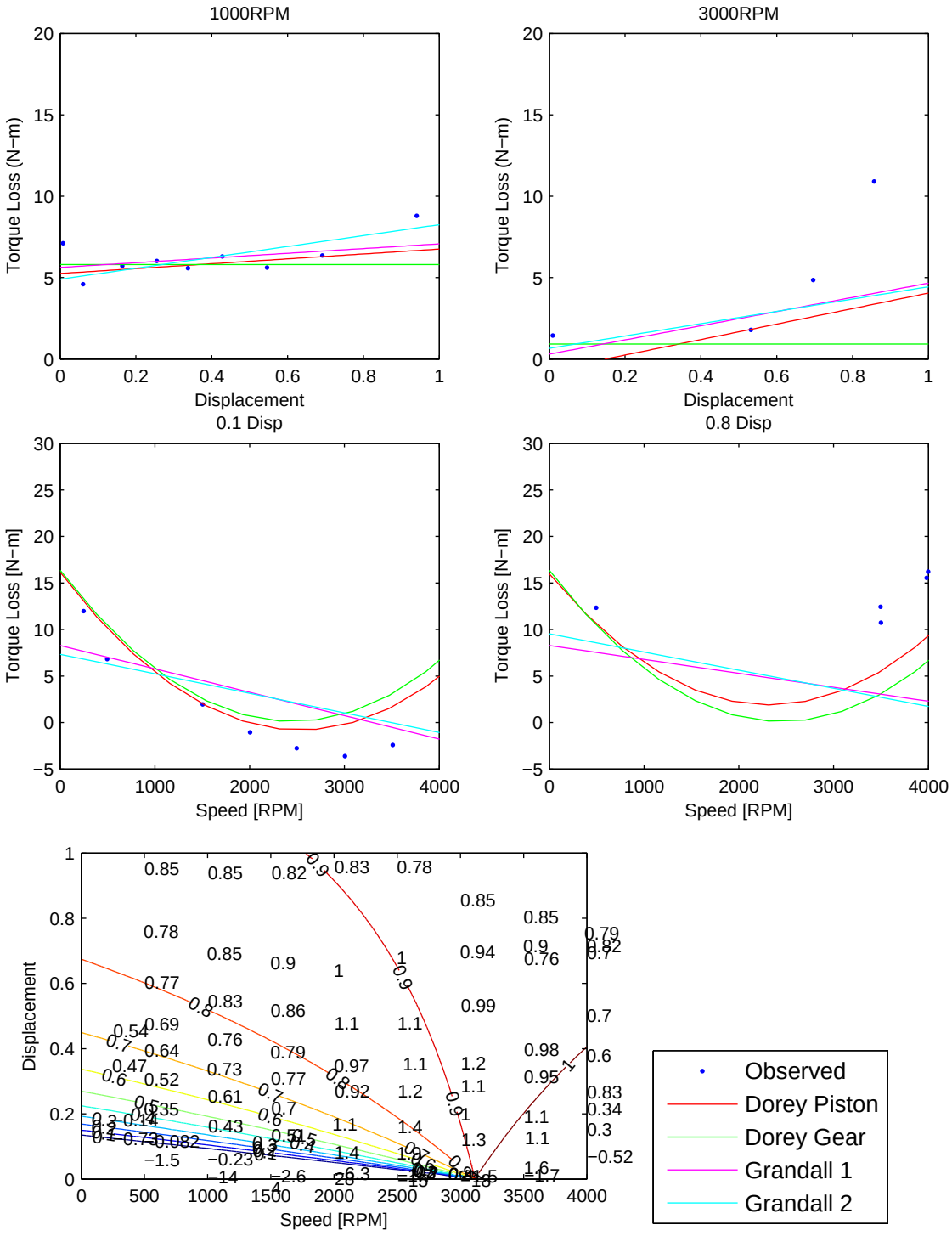


Figure 6.9: Motor CCW torque model results, 13.8 MPa (2000 psi) ΔP. Mechanical efficiency graph utilizes the GM1 torque model.

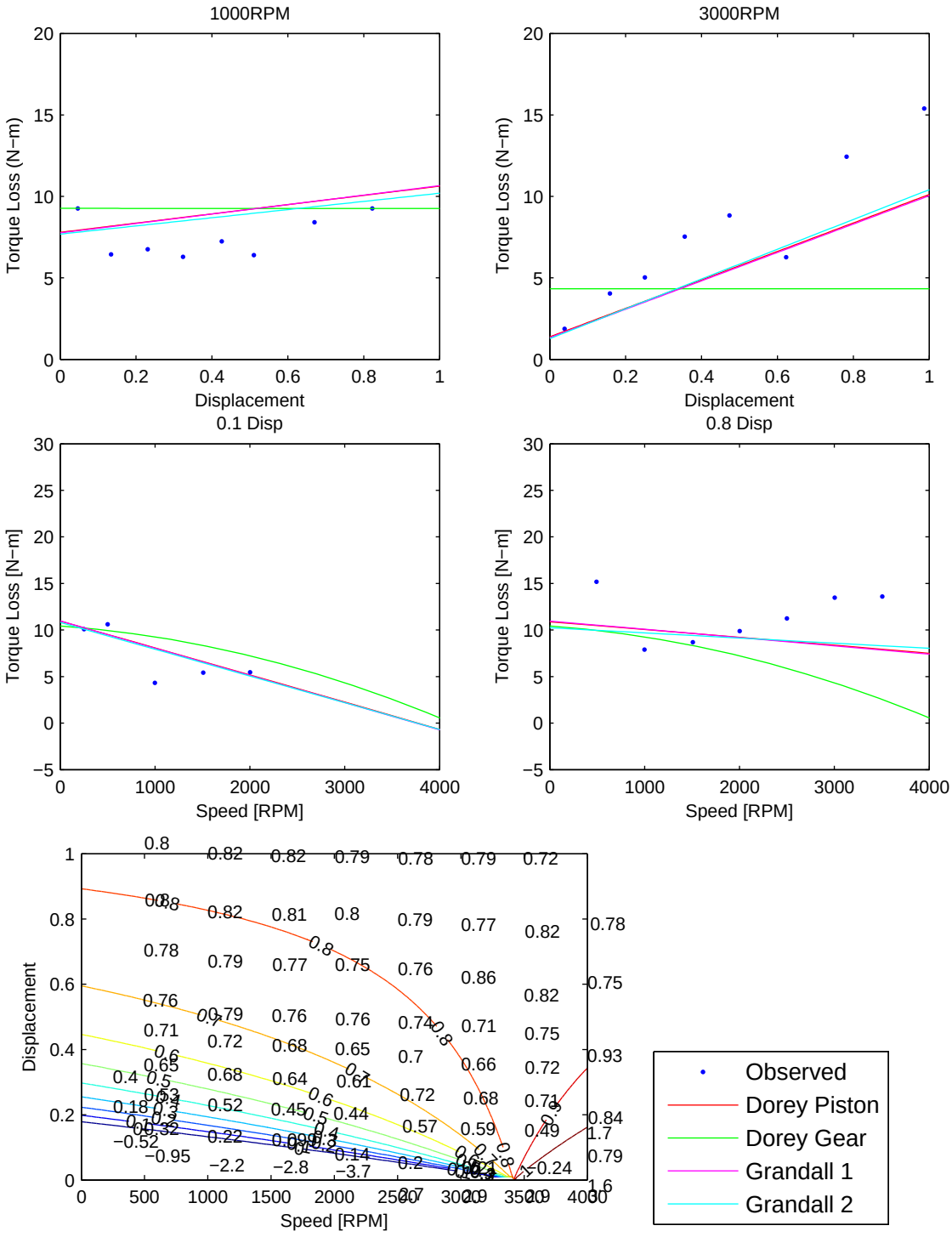


Figure 6.10: Motor CW torque model results, 13.8 MPa (2000 psi) ΔP. Mechanical efficiency graph utilizes the GM1 torque model.

data. A lower value for SST is better. But statistics do not tell the whole story. A model with low SST could be following some of the idiosyncrasies of the data and not a general trend. An example of this is the Dorey gear model seen in Figure 6.10.

For the flow models, GM1 appeared to behave erratically in Figure 6.4. The upper graphs of Figure 6.6 show that GM4 and GM1 both predict negative losses at high displacements. This behavior is erroneous and not acceptable, especially when better alternatives are available. The non-linear behavior of the Dorey gear model appeared to be unnecessary and overly complicated. The data did not show consistent non-linear behavior. The Dorey piston model had significantly higher SST values than GM2 or GM3. The remaining two models, GM2 and GM3, appeared to be equal in merit. A look through all of the efficiency maps was necessary to find the best model. The Motor CW volumetric efficiency graphs provided a slight, yet significant difference. Figure 6.11 shows those graphs. GM3 predicts 90% volumetric efficiency above 0.8 displacement. GM2 only predicts 80% volumetric efficiency above 0.5 displacement. The numbers in the graph provide the volumetric efficiency numbers from the actual data points. These graphs clearly show that GM3 is the best flow model.

The process is similar for the torque models. Both Dorey models exhibit non-linear behavior because of their second degree speed dependendence. This does not appear to be necessary, because the data do not show consistent non-linear behavior. The Dorey gear model shown in Figure 6.8 also shows some strange behavior. GM1 and GM2 appear to predict the data equally well. But the SST data favor GM1 consistently. GM2 is more complex, having seven parameters. GM1 is simpler, having only four parameters. Simpler models are more easily implemented. Therefore, GM1 is the best torque model.

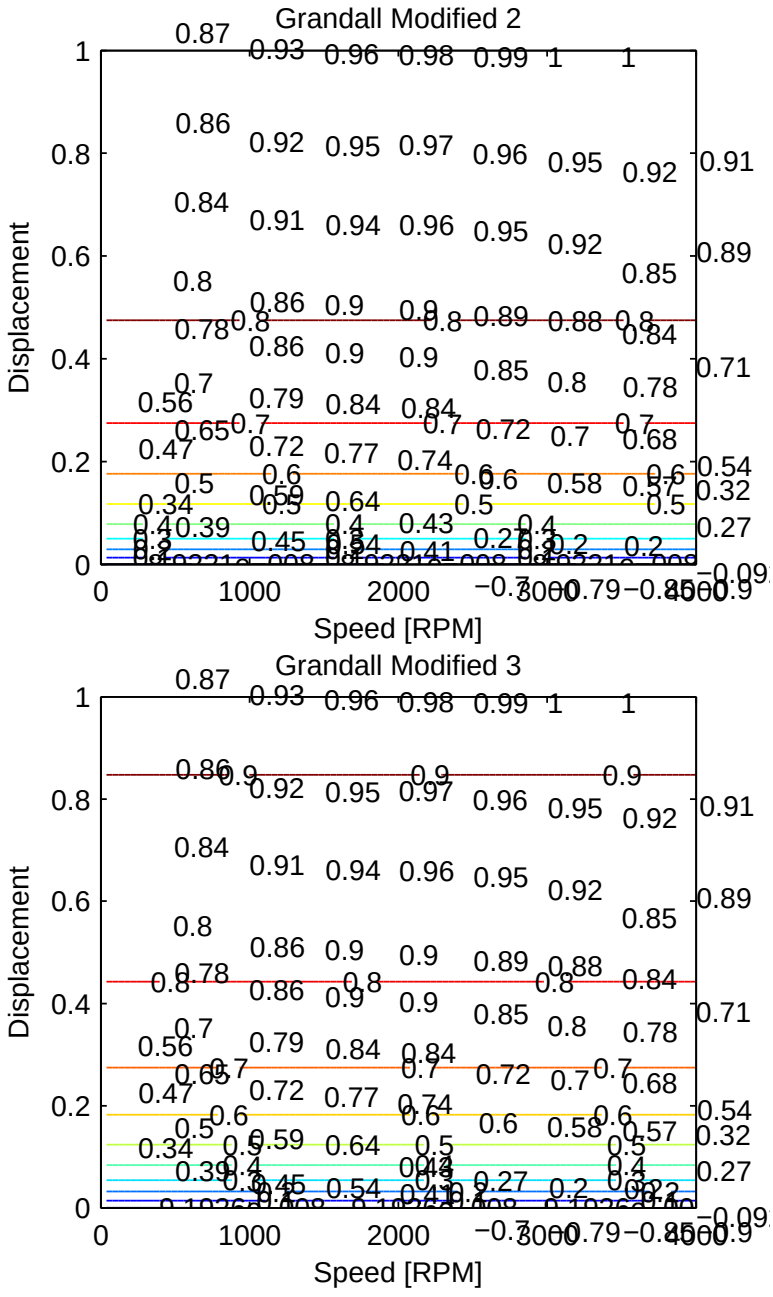


Figure 6.11: Motor CW Volumetric Efficiency Results using two flow models, 13.8 MPa ΔP

6.6 Best Model Surface Plots

Section 6.5 shows surface plots of the GM3 flow model and the GM1 torque model in Figures 6.12 through 6.15. These were previously determined in Section 6.4 to be the best models. Each plot graphs the flow or torque loss vs displacement and speed. The model results are shown as a colored surface. The points in the plot represent the actual data points.

Variations with pressure have not been previously plotted. Concerns existed about the linear or non-linear effect of pressure on torque loss. Figure 6.16 shows torque loss vs. displacement and pressure. Torque loss does not appear to be influenced by pressure when in pump mode. In motor mode, torque loss is reduced with increasing pressure. The reason for this behavior is not known.

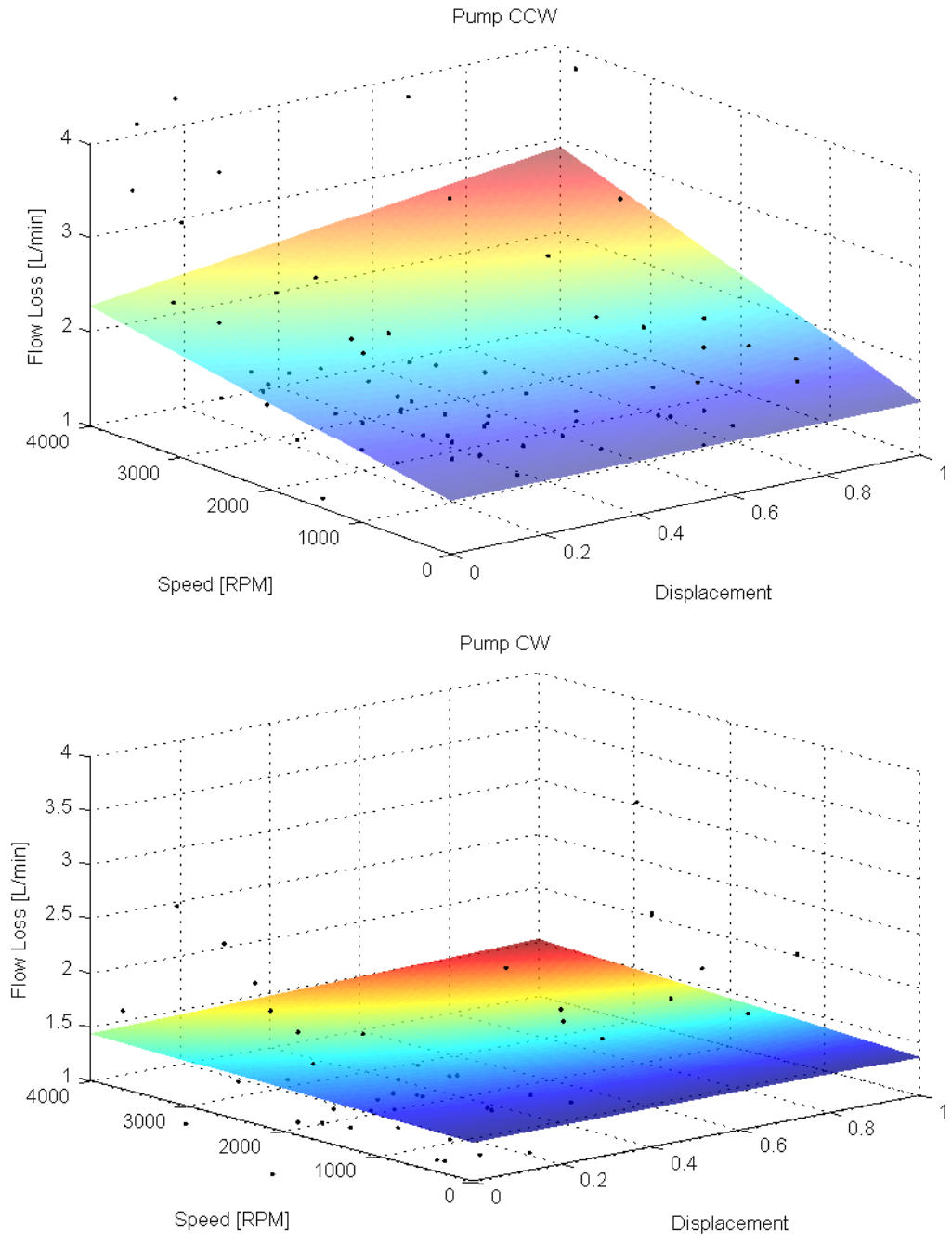


Figure 6.12: Surface plots of GM3 flow model with data points in pump mode, 13.8 MPa ΔP

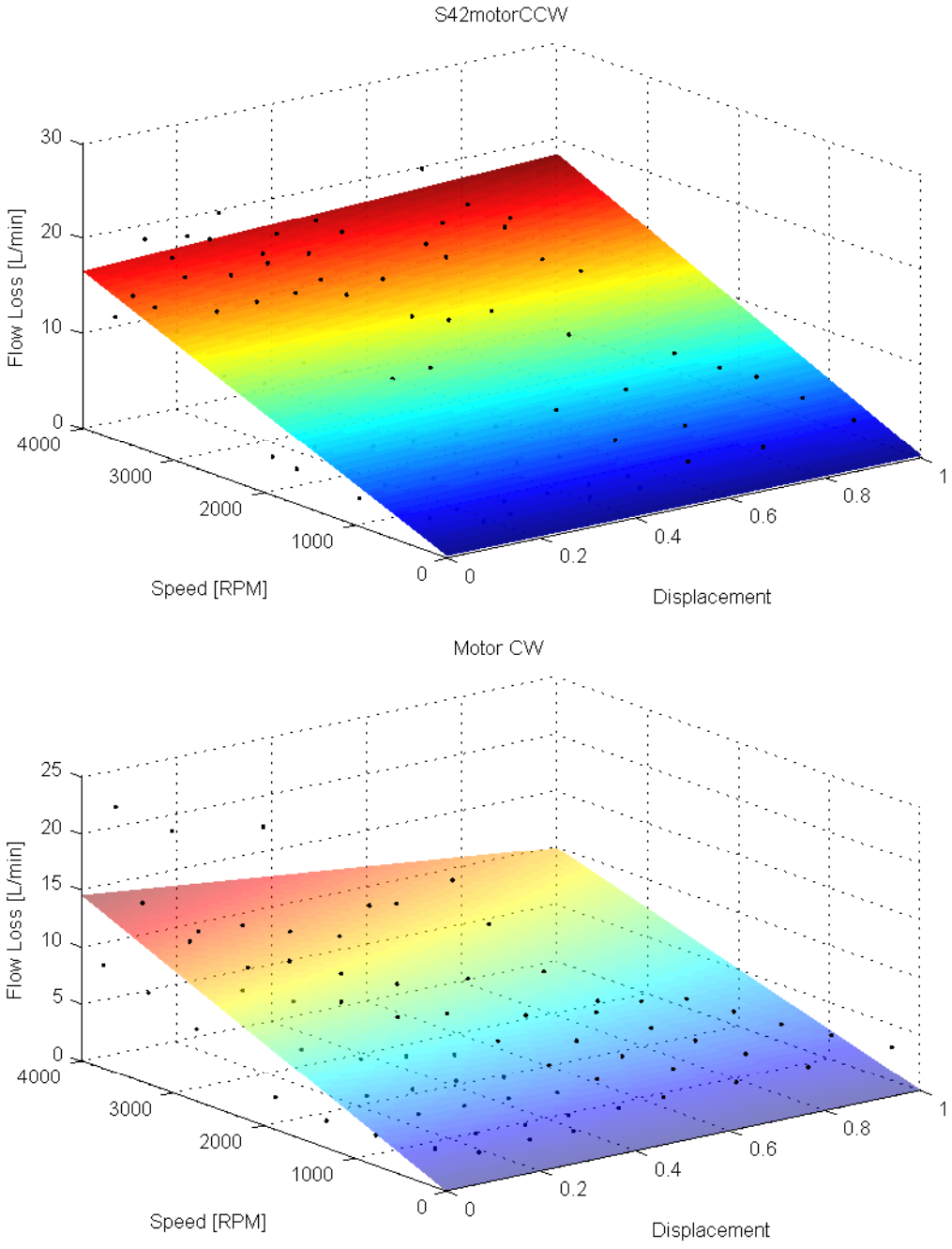


Figure 6.13: Surface plots of GM3 flow model with data points in motor mode, 13.8 MPa ΔP

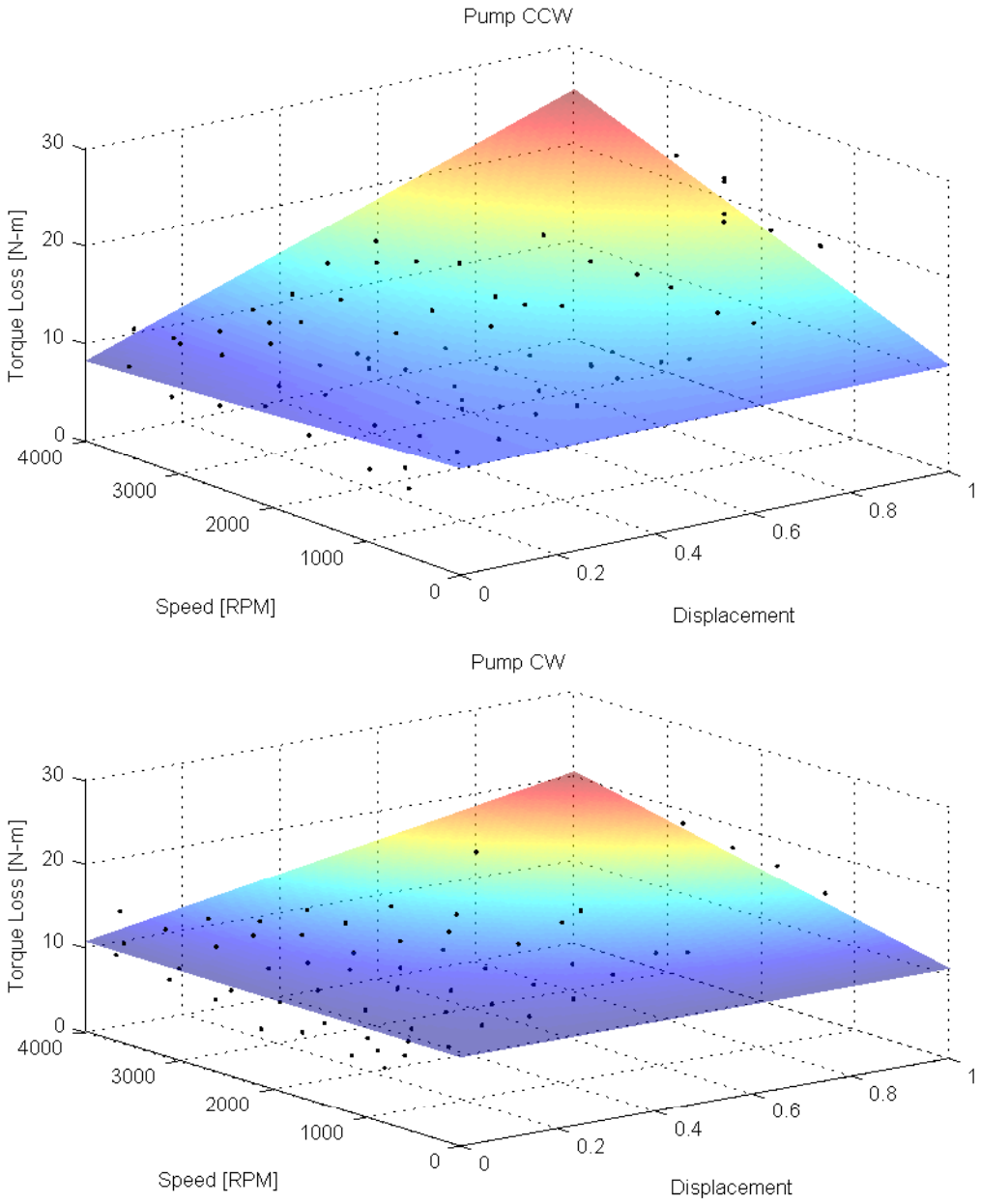


Figure 6.14: Surface plots of GM1 torque model with data points in pump mode, 13.8 MPa ΔP

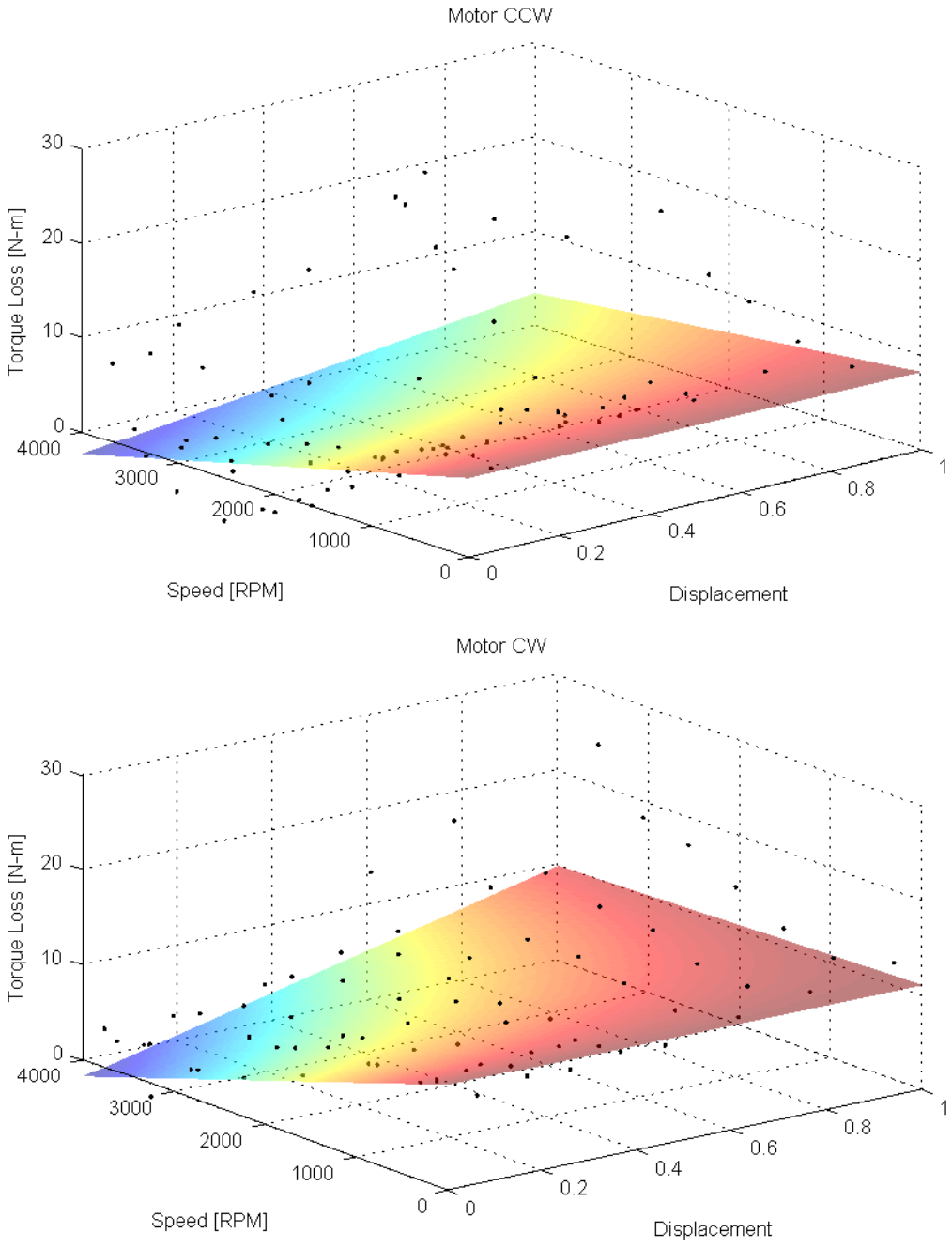


Figure 6.15: Surface plots of GM1 torque model with data points, in motor mode, 13.8 MPa ΔP

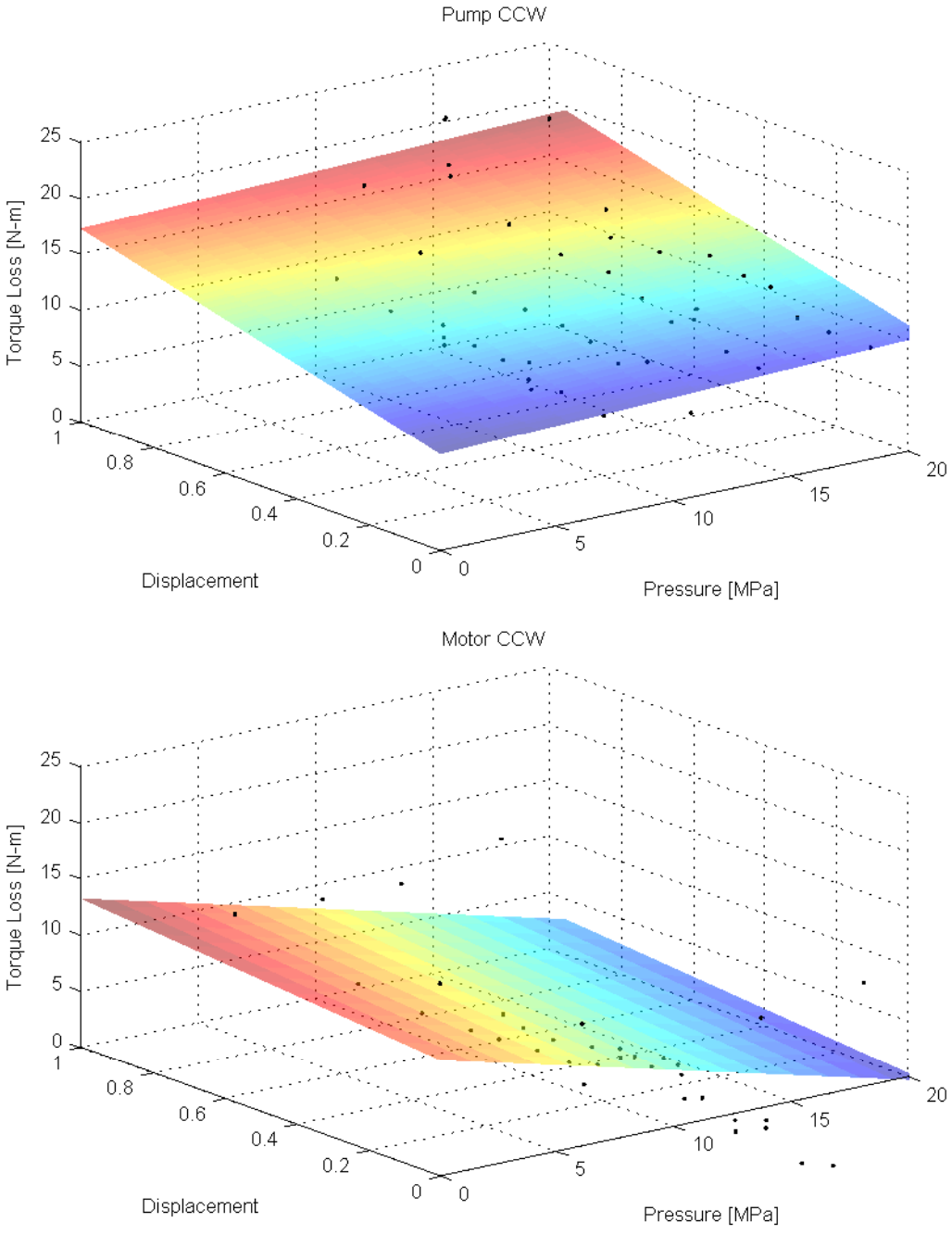


Figure 6.16: Surface plots of GM1 torque model, loss vs. pressure and displacement, at 2000RPM.

Chapter 7

Conclusions

This chapter reviews the content in this thesis and presents conclusions based on the analysis. Section 7.1 reviews the content of the thesis. Section 7.2 includes a summary of the contributions of this research to the literature. Finally, Section 7.3 highlights several recommendations for future work.

7.1 Review

Chapter 1 included background information about the project and a literature review. The literature review provided background in P/M testing, models, and various methods to improve the efficiencies of P/M's. Chapter 2 provided detailed information about the physical parts of the test stand. This included a hydraulic schematic and several electrical block diagrams showing how the sensors were connected to the rest of the system. The Simulink diagram controlling the test stand, the controls algorithm, and the experimental procedure were included in Chapter 3. Chapter 4 described the process of converting the raw data into useful data points, including a discussion of uncertainty. Modeling was

the topic of Chapter 5. This included the ideal model, the formulation of the Dorey [4] model, and improvements to the Dorey model. Chapter 6 contains many graphs showing the results of the modeling found in Chapter 5.

7.2 Contributions

The primary accomplishment of this research is the construction of a P/M efficiency test stand. The stand gathers efficiency data which can then be used to formulate efficiency models. These models can then be used to predict the performance of P/M units in the context of a larger system.

The Dorey [4] model is used in this research. The flow and torque models adequately predict the flow and torque losses. Predicted overall efficiencies correlate reasonably well with the observed data.

Small improvements to the Dorey model were achieved by modifying the flow and torque models. Nevertheless, room for further improvement still exists. Even though the model is not perfect, we can gain knowledge about the performance and efficiency of P/M units at low speeds and low displacements. These are two regimes which have not received a great deal of attention in the past.

The test stand itself will be used in the future to test other P/M units for use on the HHPV and other projects at the University of Minnesota. While different efficiency models may be used in the future, the physical setup and software will be retained and reused.

P/M units are now being introduced in novel applications where hydraulics has not been found in the past. This research feeds directly into the HHPV project at the University of Minnesota. The data and resulting models are used to predict the performance

and efficiency of the HHPV, a hydraulic hybrid vehicle. In hydraulic hybrid vehicles, P/M units are predicted to spend a great deal of time in low displacement and low speed regimes. This research provides new efficiency information at those conditions.

7.3 Recommendations for Future Work

The S42 P/M's do not have displacement sensors which output the position of the swashplate. This causes two problems. First, the P/M unit cannot be precisely and accurately set to a certain displacement. Second, the exact displacement a P/M is operating at is not known. This causes a significant amount of error when calculating the derived volume.

The problem was worst in motor mode. At low and high speeds and moderate to high displacement, the derived volume was close to the theoretical value. At moderate speeds and all displacements, the derived volume was about 30% lower than the theoretical value. At low speeds and low displacements the error was greater than 50%. This was likely caused by an inaccurate displacement control system inside the P/M unit.

P/M manufacturers generally include displacement sensors on their largest units within a series (~200 cc/rev) as a stock item. But displacement sensors are not even an option for smaller units. When testing variable displacement P/M's, an attempt should be made to insert some form of displacement sensor into the case. The lack of displacement sensors has also caused great difficulty in controlling the P/M's on the HHPV.

Other efficiency models exist in the literature. They may be applied to the S42 data. A comparison of the Dorey models, the improvements suggested in this thesis, and other models in the literature may be valuable. The Schlösser model [23, 24] appears to be worth pursuing. The MATLAB programs may be changed easily to reflect the new equa-

tions.

The P/M efficiency test stand is a valuable tool in the creation of performance and efficiency models of hydraulic pumps and motors. The stand measures the efficiency of P/M units at many conditions. Models are then fit to this data. These component models are important for predicting the performance of the entire system, whatever it may be.

Fluid power will continue to expand into new applications. These new applications require component models. These component models are based on data collected using P/M test stands similar to the one presented in this research. Therefore, the test stand built in the scope of this work is likely to have future utility.

Bibliography

[1] ISO 4409. Hydraulic fluid power: Positive-displacement pumps, motors and integral transmissions: Methods of testing and presenting basic steady state performance. Technical Report 4409, ISO, 2007.

[2] ISO 8426. Hydraulic fluid power : Positive displacement pumps and motors: Determination of derived capacity. Technical Report 8426, ISO, 2008.

[3] Jeff W. Dobchuk, Richard T. Burton, Peter N. Nikiforuk, and Paul R. Ukrainetz. Mathematical modeling of a variable displacement axial piston pump. In *Proceedings of the Fluid Power and Technology Division of the ASME*, pages 1–8, 1999.

[4] R. E. Dorey. Modeling of losses in pumps and motors. In *Proceedings of the First Bath International Fluid Workshop*, pages 71–97, 1988.

[5] Akira Hibi and Tsuneo Ichikawa. Torque performance of pressure balanced type vane motor at start and in low speed range. *Bulletin of the JSME*, 19(138), December 1976.

[6] Ken Ichiryu, Masayuki Niikura, and Yasunori Hasegawa. Efficiency estimation of hydraulic pump by thermodynamical method. In *Proceedings of the Third International Symposium on Fluid Power Transmission and Control*, pages 89–93, 1999.

[7] Y Inaguma and A Hibi. Vane pump theory for mechanical efficiency. *Journal of Mechanical Engineering Science*, 219, 2005.

[8] Yu Jinghong, Chen Zhaoneng, and Lu Yuanzhang. The variation of oil effective bulk modulus with pressure in hydraulic systems. *ASME Transactions*, 116:146–150, 1994.

[9] Robert E. Johnson and Noah D. Manring. Modeling a variable displacement pump. In *Proceedings of the 1994 ASME Fluids Engineering Division Summer Meeting*, pages 1–10, 1994.

[10] Mansour A. Karkoub, Osama E. Gad, and Mahmoud G. Rabie. Predicting axial piston pump performance using neural networks. *Mechanism and Machine Theory*, 34:1211–1226, 1999.

[11] Heikki O. J. Kauranne, Asko U. Ellman, Jyrki T. Kajaste, and Matti T. Pietola. Applicability of pump models for varying operational conditions. In *Proceedings of IMECE2003*, pages 45–54, 2003.

[12] Torsten Kohmäscher. *Modellbildung, Analyse und Auslegung hydrostatischer Antriebsstrangkonzepte*. PhD thesis, Aachen University, Aachen: Shaker Verlag, 2008.

[13] Torsten Kohmäscher, Hubertus Murrenhoff, Robert Rahmfeld, and Eckhard Skirde. Improved loss modeling of hydrostatic units: Requirement for precise simulation of mobile working machine drivelines. In *Proceedings of IMECE2007*.

[14] Sang-Yul Lee and Yeh-Sun Hong. Effect of crsin thin film coating on the improvement of the low-speed torque efficiency of a hydraulic piston pump. *Surface and Coatings Technology*, 202:1129–1134, 2007.

[15] William Levine, editor. *Handbook of Networked and Embedded Control Systems*, chapter 18, pages 419–446. Birkhauser Boston, 2005.

[16] Noah D. Manring. Valve-plate design for an axial piston pump operating at low displacements. *ASME Transactions*, 125:200–201, March 2003.

[17] Noah D. Manring. Measuring pump efficiency: Uncertainty considerations. *ASME Transactions*, 127:280–284, December 2005.

[18] D. McCandlish and R. E. Dorey. The mathematical modeling of hydrostatic pumps and motors. *Proceedings of the Institution of Mechanical Engineers*, 198B(10):165–174, 1984.

[19] Herbert E. Merritt. *Hydraulic Control Systems*, page 18. Wiley, 1991.

[20] D. Mikeska and M. Ivantysynova. A precise steady state model of displacement machines for the application in virtual prototyping of power split drives.

[21] G. S. Payne, A. E. Kiprakis, M. Ehsan, W. H. S. Rampen, J. P. Chick, and A. R. Wallace. Efficiency and dynamic performance of digital displacement hydraulic transmission in tidal current energy converters. *Power and Energy*, 221:207–218, 2007.

[22] P. Renvert and W. Weiler. The comparison of various test methods to determine the starting and low speed characteristics of hydraulic motors with regard to their practical use. *Fluidics Quarterly*, 14(3):63–91, 1982.

[23] W. M. J. Schlösser. Mathematical model for displacement pumps and motors. *Hydraulic Power Transmission*, pages 252–257, 269, April 1961.

[24] W. M. J. Schlösser. The overall efficiency of positive-displacement pumps. In *BHRA Fluid Power Symposium*, pages 34–48, 1968.

[25] Ganesh K. Seeniraj and Monika Ivantysynova. Impact of valve plate design on noise, volumetric efficiency, and control effort in an axial piston pump. In *Proceedings of IMECE2006*, pages 77–84, 2006.

[26] D. P. M. Stanzial and G. L. Zarotti. Control optimization of a hydrostatic regenerative system. In *Proceedings of the IASTED Intl. Symposium: Modeling, Identification, and Control*, pages 71–75, 1988.

[27] J. Williamson. Progress towards iso standards for starting and low-speed characteristics of hydraulic motors. In *BHRA Fluid Power Symposium*, pages 49–63, 1981.

[28] W. E. Wilson. Rotary-pump theory. *ASME Transactions*, 68:371–384, May 1946.

[29] G. L. Zarotti. Pump efficiences approximation and modeling. In *BHRA Fluid Power Symposium*, pages 145–164, 1981.

[30] L. G. Zarotti. Regenerative test systems based on hydrostatic units, no. 905052. In *Proceedings of the Society of Automotive Engineers*, pages 375–381, 1990.

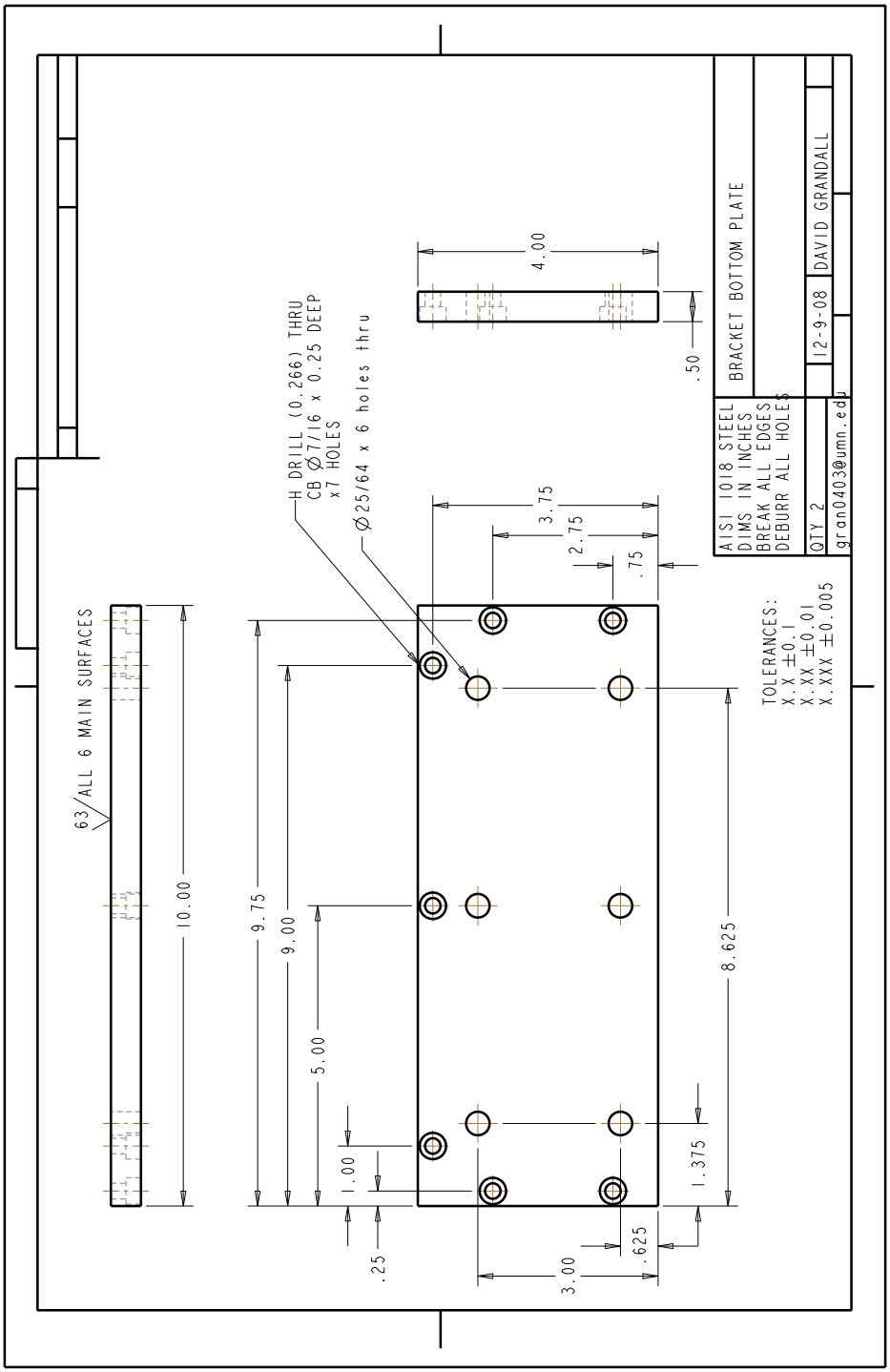
[31] Luca G. Zarotti. Rotary regenerative assemblies: Operation and control. In *Proceedings of the Second Intl. Conference of Fluid Power Transmission and Control*, 1989.

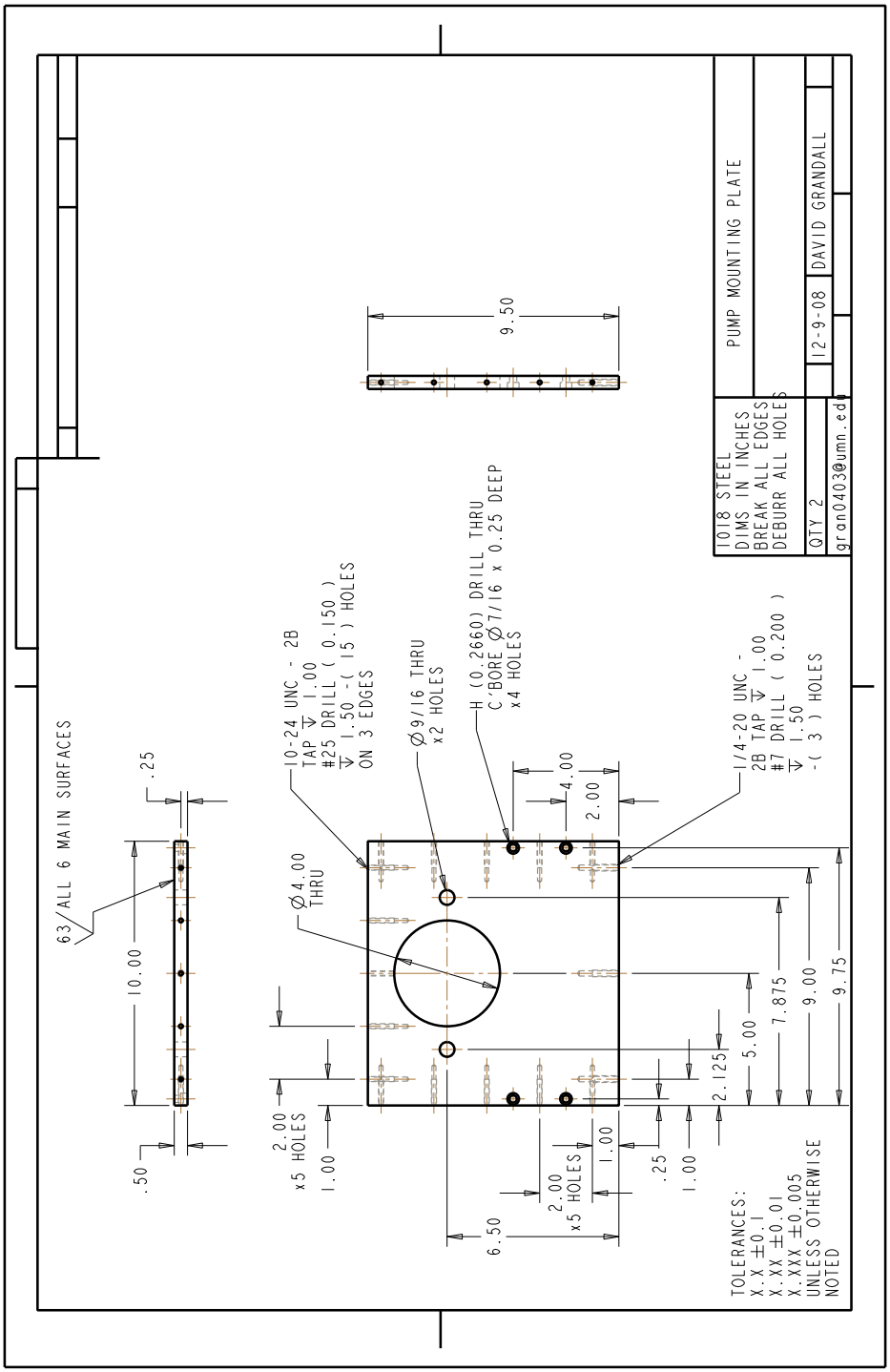
[32] Tadeusz Zloto, Zygmunt Biernacki, and Marek Kurkowski. Computer-based measuring system for determining efficiency of an axial multipiston pump. In *CADSM 2001 Proceedings*, pages 175–178, 2001.

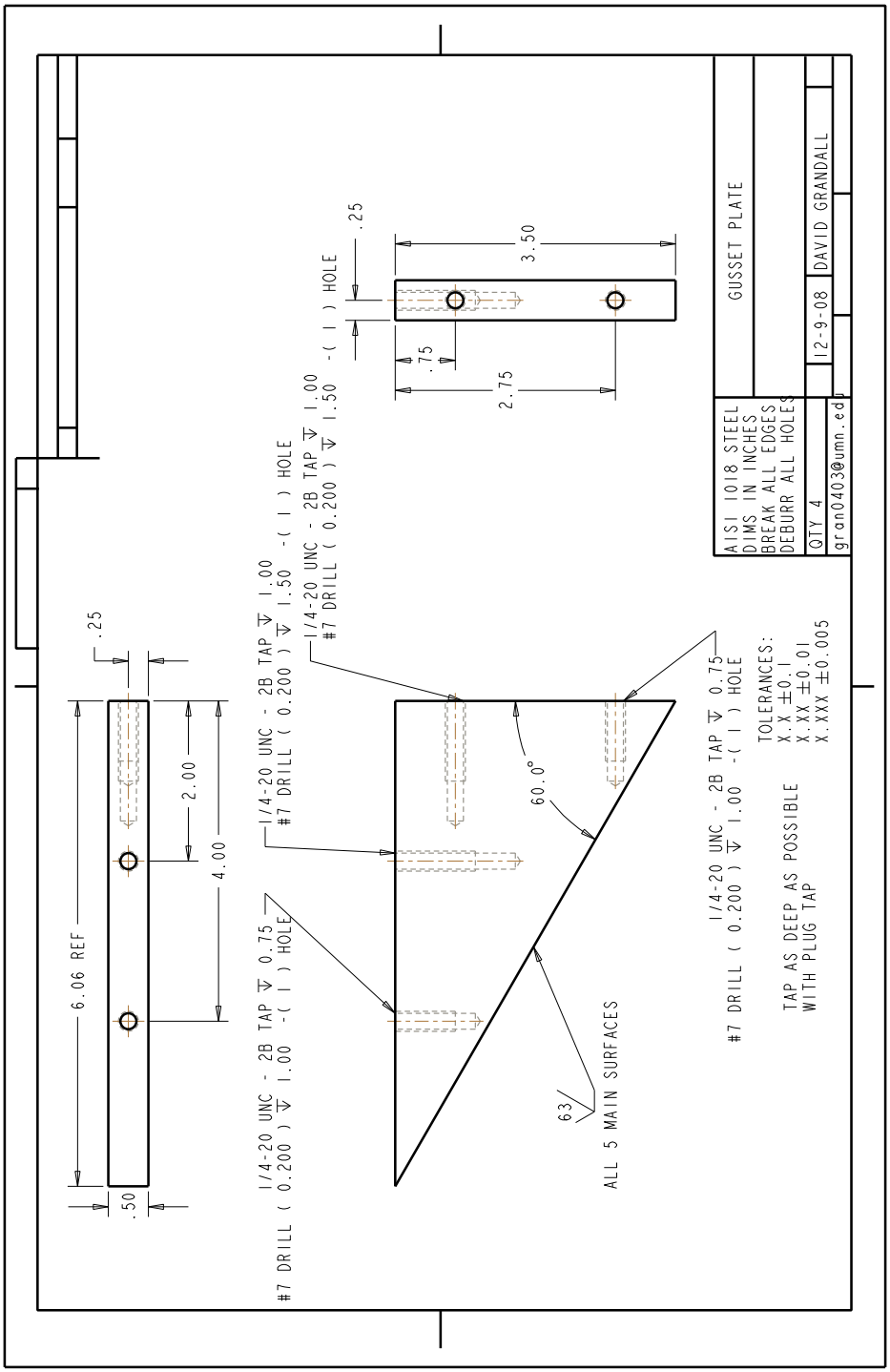
Appendix A

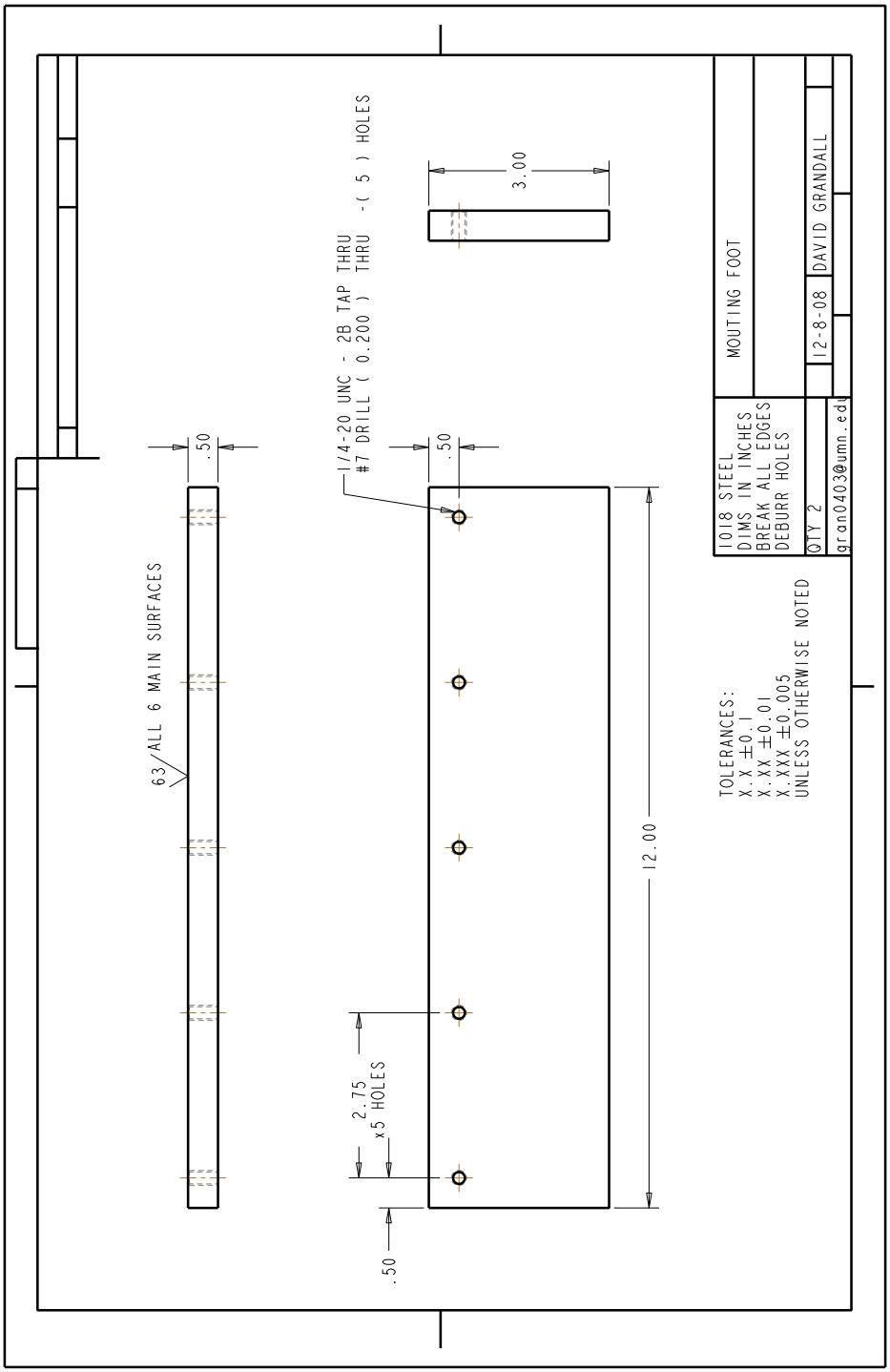
Fixture Shop Drawings

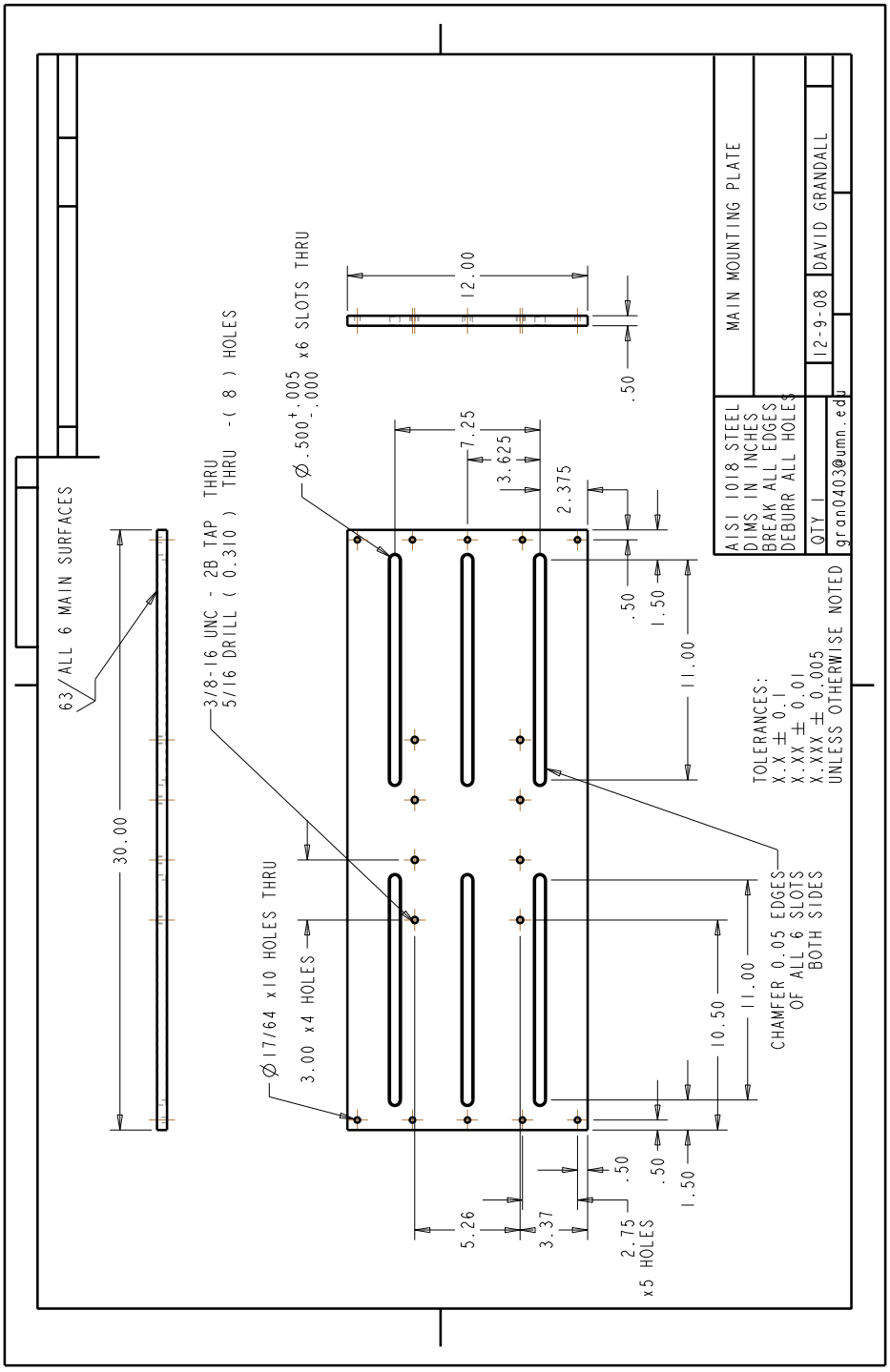
Appendix A contains the shop drawings for individual parts making up the fixture described in Section 2.4. . A color version of the assembly model created in ProlEngineer is found at the back of Appendix A.

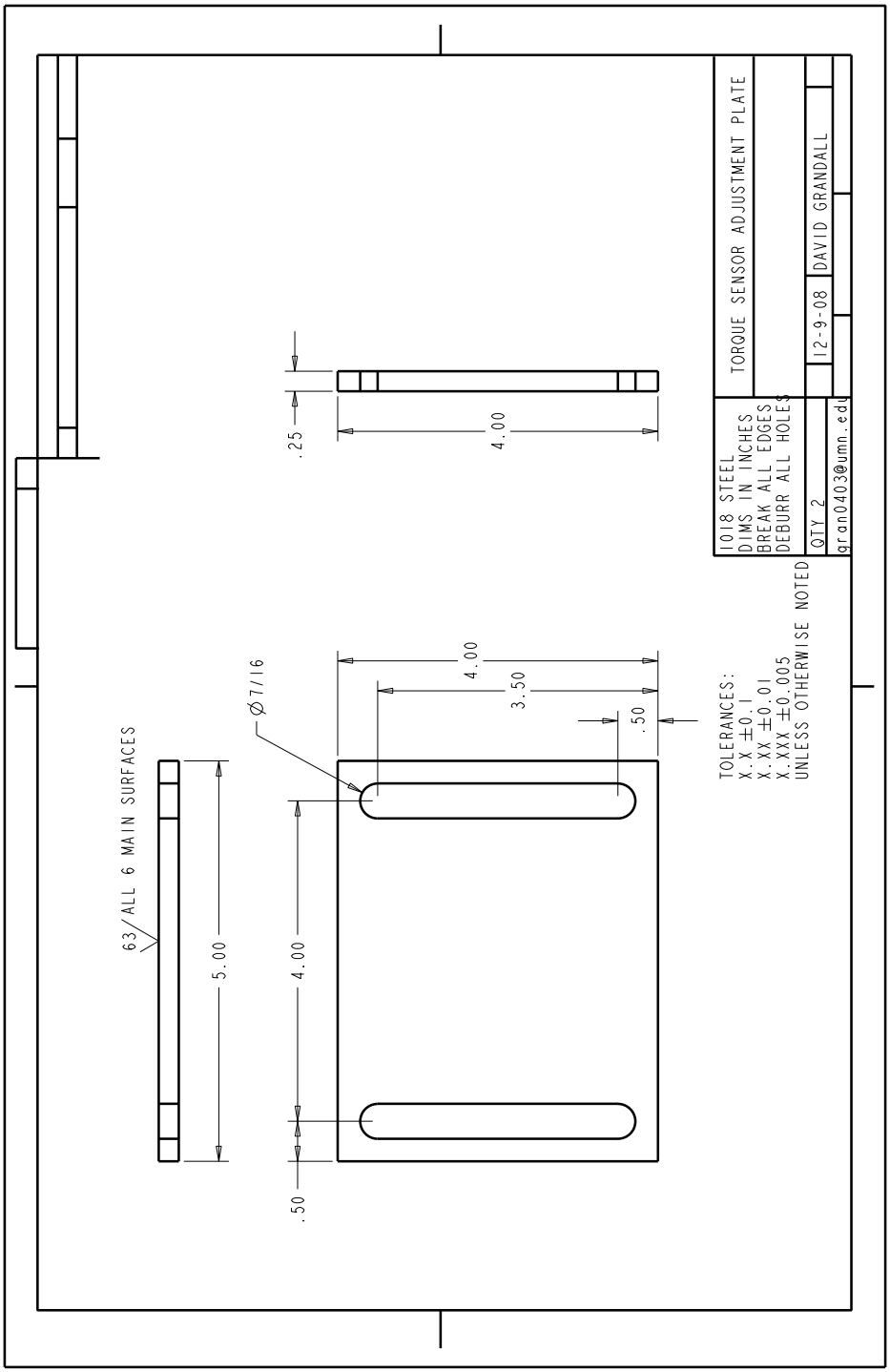

BRACKET BOTTOM PLATE
AISI 1018 STEEL
DIMS IN INCHES
BREAK ALL EDGES
DEBURR ALL HOLES
QTY 2
12-9-08
DAVID GRANDALL
gran0403@umn.edu
TOLERANCES:
X.X ±0.1
X.XX ±0.01
X.XXX ±0.005
63 ALL 6 MAIN SURFACES
H DRILL (0.266) THRU
CB Ø7/16 x 0.25 DEEP
x7 HOLES
Ø25/64 x 6 holes thru
4.00
.50
3.75
2.75
.75
10.00
9.75
9.00
5.00
1.00
.25
8.625
1.375
.625
3.00

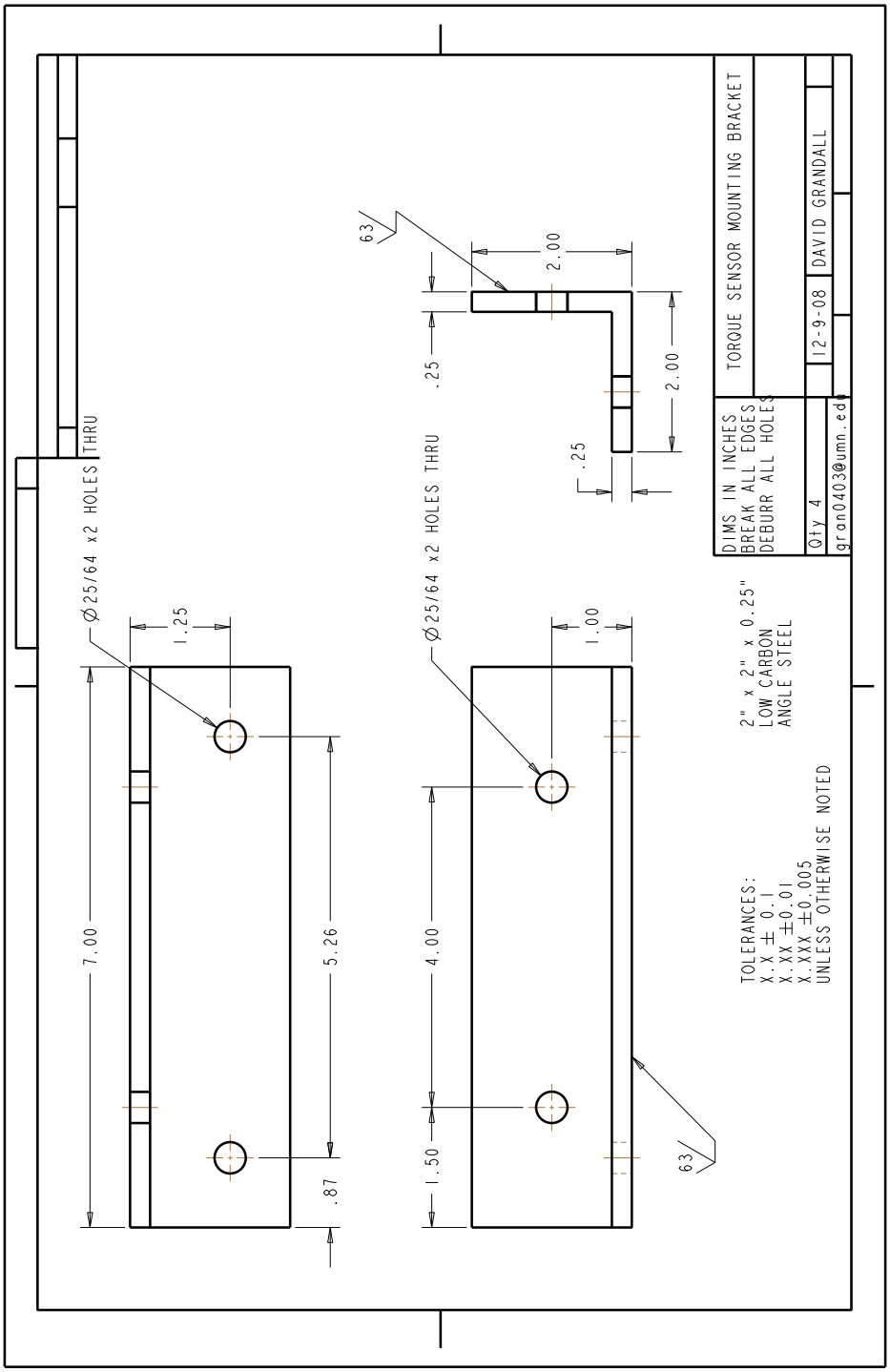

PUMP MOUNTING PLATE
1018 STEEL
DIMS IN INCHES
BREAK ALL EDGES
DEBURR ALL HOLES
QTY 2
12-9-08
DAVID GRANDALL
gran0403@umn.edu
9.50
63/ALL 6 MAIN SURFACES
.25
10.00
.50
10-24 UNC - 2B
TAP ⏚ 1.00
#25 DRILL (0.150)
⏚ 1.50 -(15) HOLES
ON 3 EDGES
Ø9/16 THRU
x2 HOLES
H (0.2660) DRILL THRU
C'BORE Ø7/16 x 0.25 DEEP
x4 HOLES
4.00
2.00
1/4-20 UNC -
2B TAP ⏚ 1.00
#7 DRILL (0.200)
⏚ 1.50
-(3) HOLES
Ø4.00
THRU
2.00
x5 HOLES
1.00
2.00
x5 HOLES
1.00
6.50
.25
1.00
2.125
5.00
7.875
9.00
9.75
TOLERANCES:
X.X ±0.1
X.XX ±0.01
X.XXX ±0.005
UNLESS OTHERWISE
NOTED

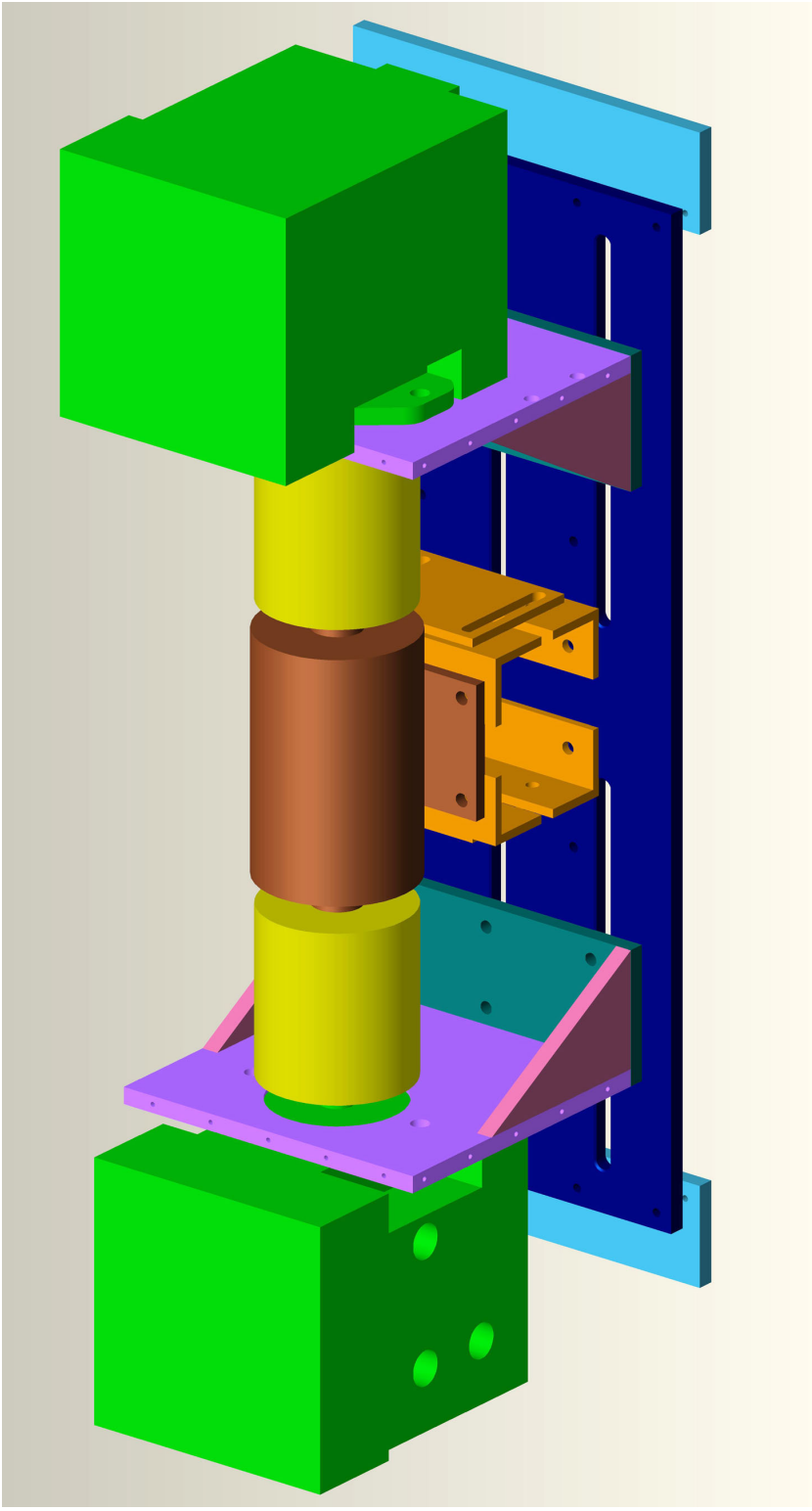

GUSSET PLATE
AISI 1018 STEEL
DIMS IN INCHES
BREAK ALL EDGES
DEBURR ALL HOLES
QTY 4
12-9-08
DAVID GRANDALL
gran0403@umn.edu
TOLERANCES:
X.X ±0.1
X.XX ±0.01
X.XXX ±0.005
TAP AS DEEP AS POSSIBLE
WITH PLUG TAP
#7 DRILL (0.200) ⊽ 1.00 -(1) HOLE
1/4-20 UNC - 2B TAP ⊽ 0.75
#7 DRILL (0.200) ⊽ 1.50 -(1) HOLE
1/4-20 UNC - 2B TAP ⊽ 1.00
#7 DRILL (0.200) ⊽ 1.00 -(1) HOLE
1/4-20 UNC - 2B TAP ⊽ 1.00
ALL 5 MAIN SURFACES
63
60.0°
.25
3.50
.75
2.75
6.06 REF
2.00
4.00
.50


MOUTING FOOT
1018 STEEL
DIMS IN INCHES
BREAK ALL EDGES
DEBURR HOLES
QTY 2
12-8-08
DAVID GRANDALL
gran0403@umn.edu
TOLERANCES:
X.X ±0.1
X.XX ±0.01
X.XXX ±0.005
UNLESS OTHERWISE NOTED
63 ALL 6 MAIN SURFACES
1/4-20 UNC - 2B TAP THRU
#7 DRILL (0.200) THRU -(5) HOLES
3.00
.50
.50
12.00
2.75
x5 HOLES
.50


MAIN MOUNTING PLATE
AISI 1018 STEEL
DIMS IN INCHES
BREAK ALL EDGES
DEBURR ALL HOLES
QTY 1
12-9-08
DAVID GRANDALL
gran0403@umn.edu
63 ALL 6 MAIN SURFACES
3/8-16 UNC - 2B TAP THRU
5/16 DRILL (0.310) THRU -(8) HOLES
Ø.500 +.005 -.000 x6 SLOTS THRU
Ø17/64 x10 HOLES THRU
3.00 x4 HOLES
30.00
12.00
.50
7.25
3.625
2.375
.50
1.50
11.00
10.50
11.00
.50
1.50
5.26
3.37
.50
.50
2.75
x5 HOLES
CHAMFER 0.05 EDGES
OF ALL 6 SLOTS
BOTH SIDES
TOLERANCES:
X.X ± 0.1
X.XX ± 0.01
X.XXX ± 0.005
UNLESS OTHERWISE NOTED


TORQUE SENSOR ADJUSTMENT PLATE
12-9-08
DAVID GRANDALL
1018 STEEL
DIMS IN INCHES
BREAK ALL EDGES
DEBURR ALL HOLES
QTY 2
grand0403@umn.edu
TOLERANCES:
X.X ±0.1
X.XX ±0.01
X.XXX ±0.005
UNLESS OTHERWISE NOTED
.25
4.00
4.00
3.50
.50
Ø7/16
63 ALL 6 MAIN SURFACES
5.00
4.00
.50

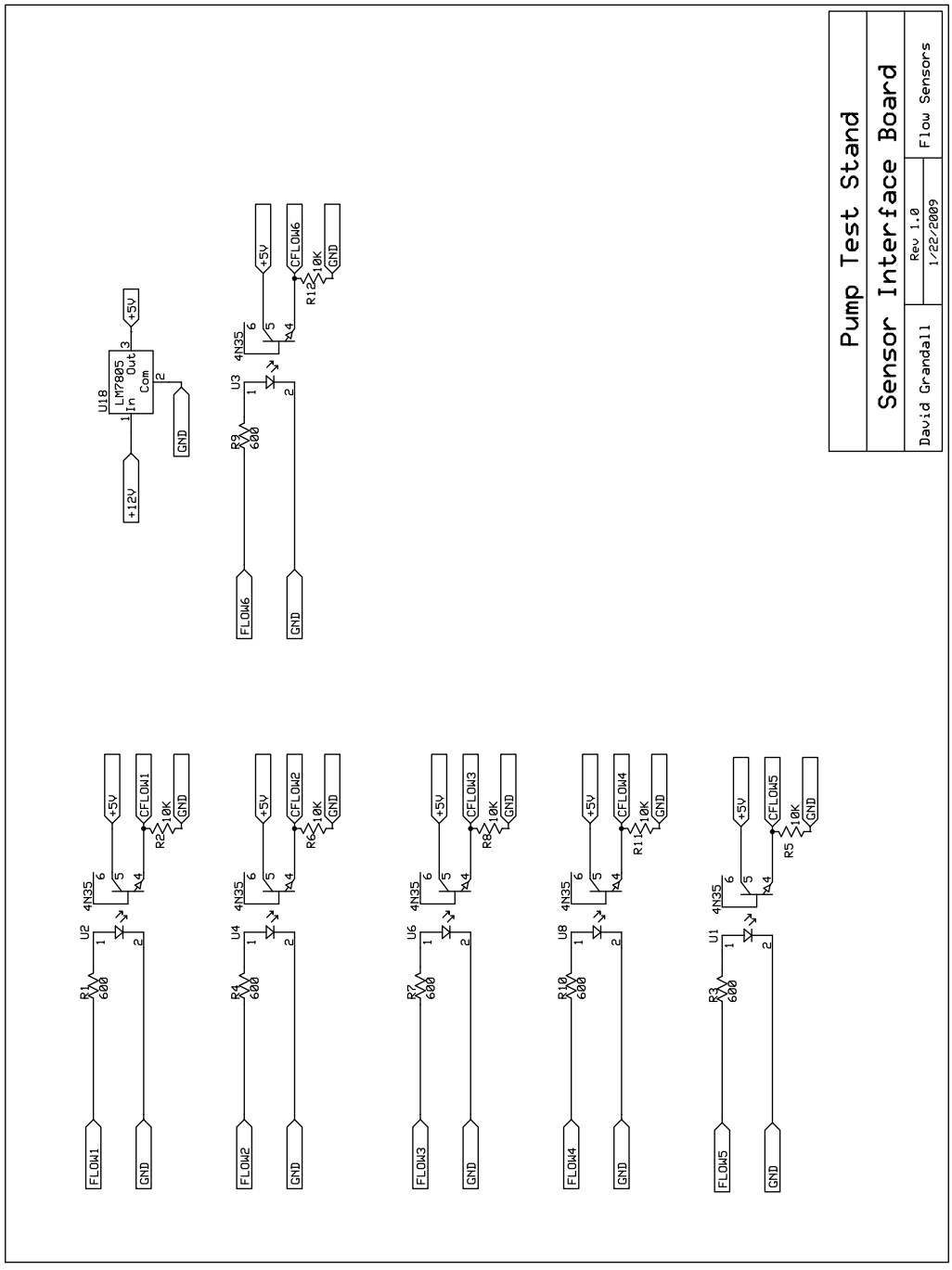

TORQUE SENSOR MOUNTING BRACKET
DIMS IN INCHES
BREAK ALL EDGES
DEBURR ALL HOLES
Qty 4
12-9-08
DAVID GRANDALL
gran0403@umn.edu
2" x 2" x 0.25"
LOW CARBON
ANGLE STEEL
TOLERANCES:
X.X ± 0.1
X.XX ±0.01
X.XXX ±0.005
UNLESS OTHERWISE NOTED
Ø 25/64 x2 HOLES THRU
1.25
7.00
5.26
.87
Ø 25/64 x2 HOLES THRU
1.00
4.00
1.50
2.00
.25
2.00
.25
63
63

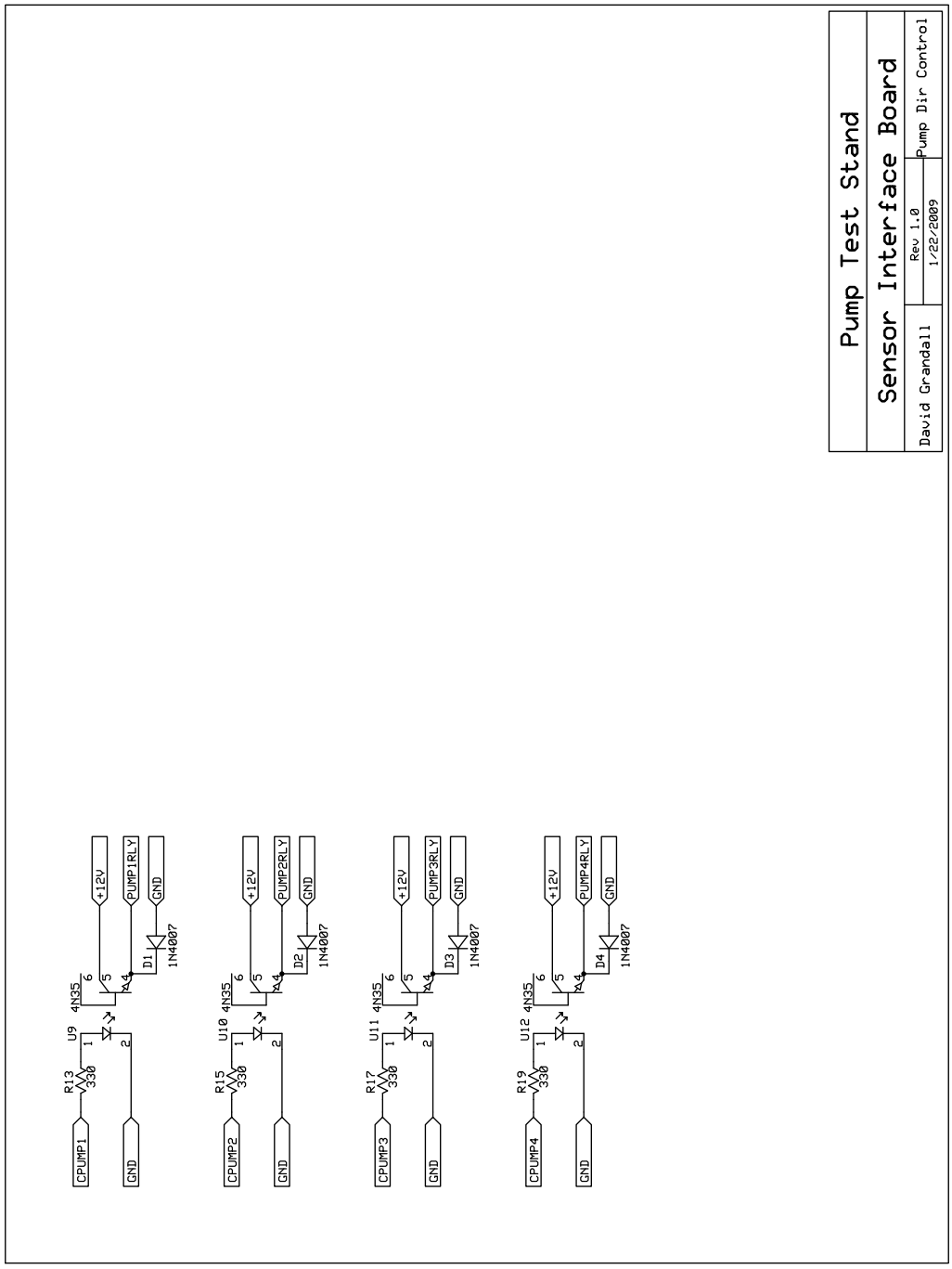


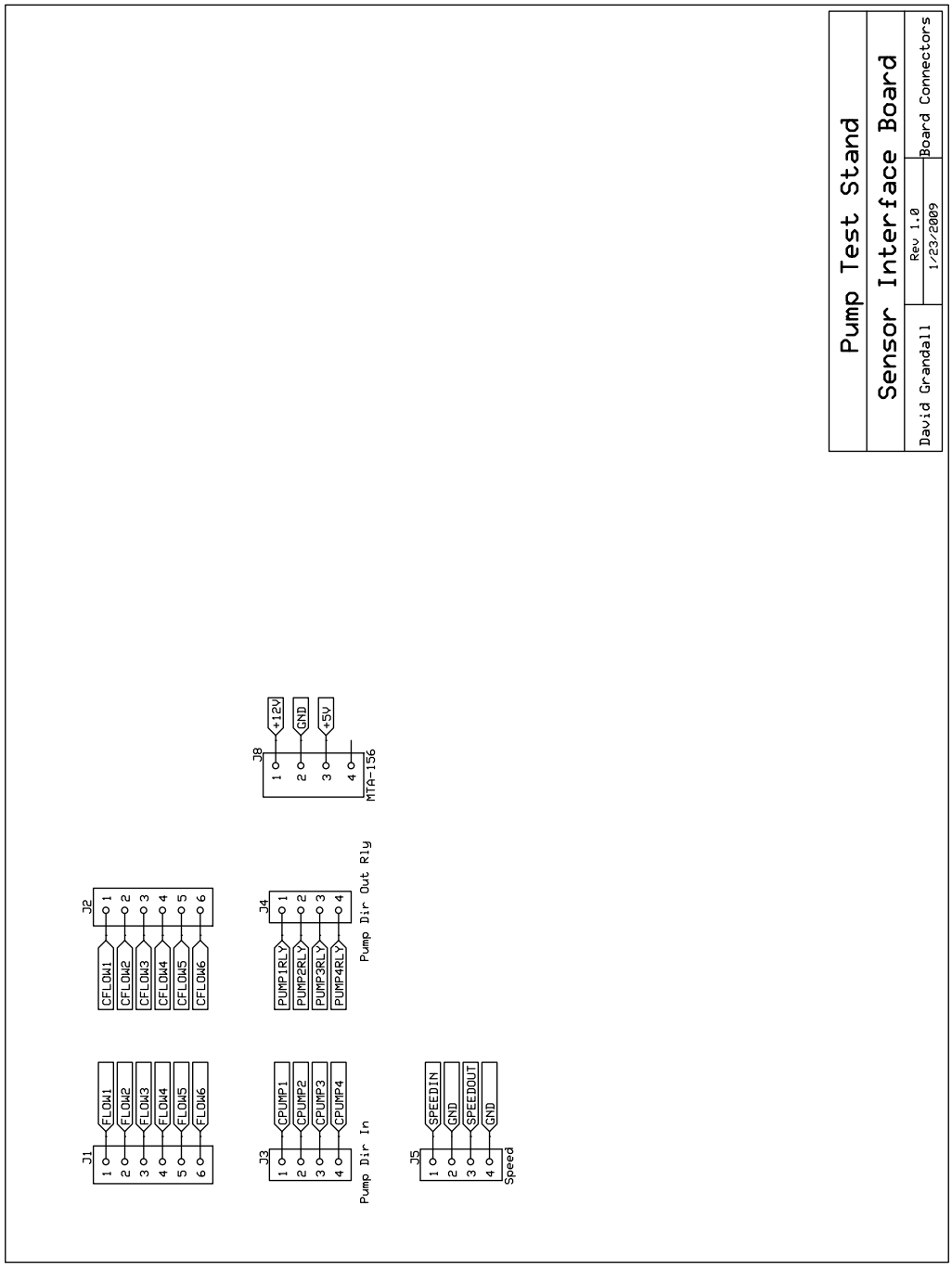
Appendix B

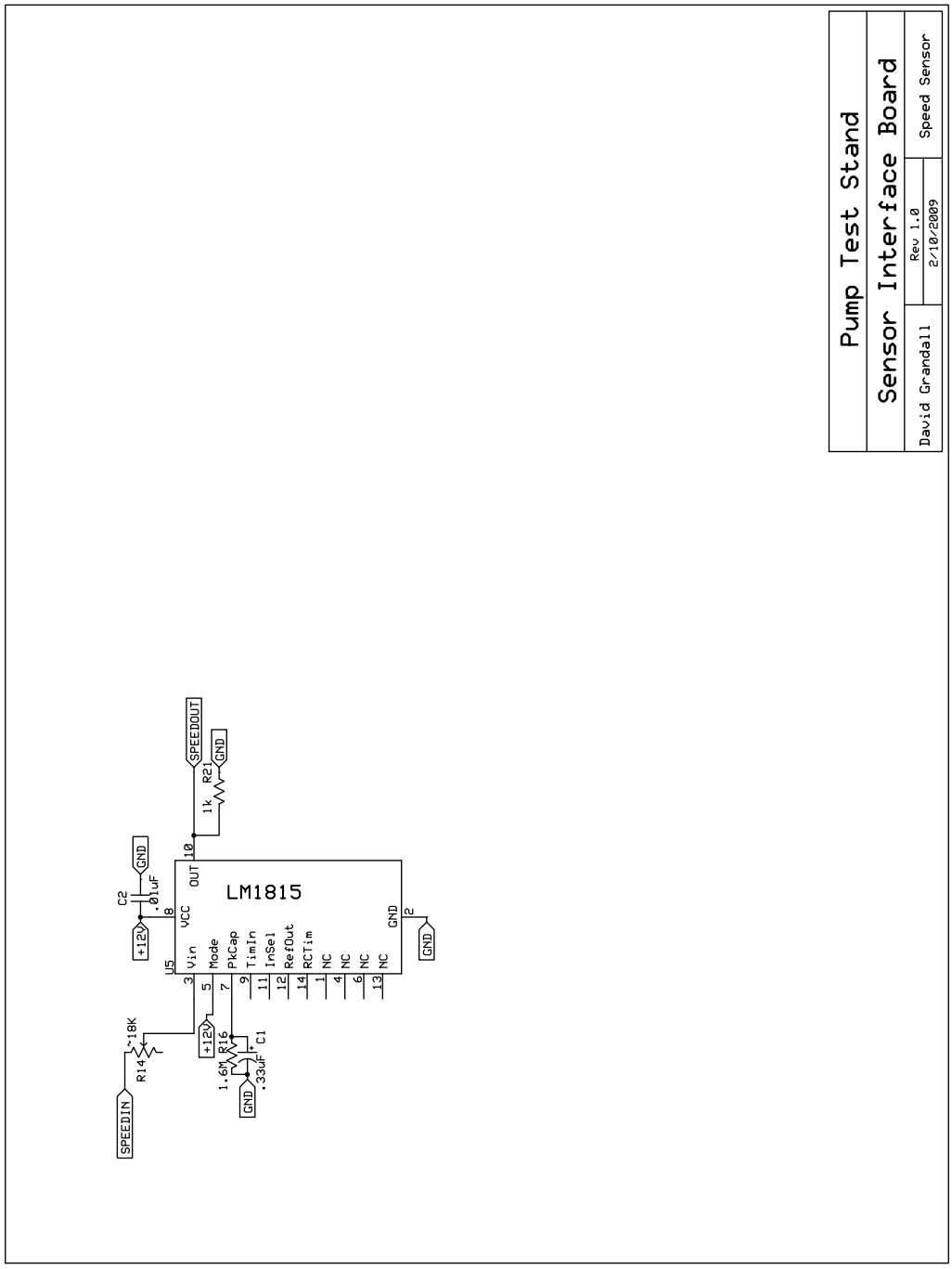
Printed Circuit Board Schematic

Appendix B contains several pages of schematics used to create the custom printed circuit board described in Section 2.5. The schematic and corresponding PCB layout were made in ExpressSCH and ExpressPCB available from http://www.ExpressPCB.com. ExpressPCB fabricated the circuit board. A bill of materials is also included.


Pump Test Stand
Sensor Interface Board
David Grandall
Rev 1.0
1/22/2009
Flow Sensors
U18
LM7805
In
Com
Out
+12V
GND
+5V
U1
U2
U3
U4
U6
U8
4N35
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
600
10K
CFLOW1
CFLOW2
CFLOW3
CFLOW4
CFLOW5
CFLOW6
FLOW1
FLOW2
FLOW3
FLOW4
FLOW5
FLOW6


Pump Test Stand
Sensor Interface Board
David Grandall
Rev 1.0
1/22/2009
Pump Dir Control
U9
4N35
R13
330
CPUMP1
GND
+12V
PUMP1RLY
GND
D1
1N4007
U10
4N35
R15
330
CPUMP2
GND
+12V
PUMP2RLY
GND
D2
1N4007
U11
4N35
R17
330
CPUMP3
GND
+12V
PUMP3RLY
GND
D3
1N4007
U12
4N35
R19
330
CPUMP4
GND
+12V
PUMP4RLY
GND
D4
1N4007


Pump Test Stand
Sensor Interface Board
David Grandall
Rev 1.0
1/23/2009
Board Connectors
J1
FLOW1
FLOW2
FLOW3
FLOW4
FLOW5
FLOW6
J2
CFLOW1
CFLOW2
CFLOW3
CFLOW4
CFLOW5
CFLOW6
J3
CPUMP1
CPUMP2
CPUMP3
CPUMP4
Pump Dir In
J4
PUMP1RLY
PUMP2RLY
PUMP3RLY
PUMP4RLY
Pump Dir Out Rly
J5
SPEEDIN
GND
SPEEDOUT
GND
Speed
J8
+12V
GND
+5V
MTA-156


Pump Test Stand
Sensor Interface Board
David Grandall
Rev 1.0
2/10/2009
Speed Sensor
LM1815
U5
Vin
Mode
PkCap
TimIn
InSel
RefOut
RCTim
NC
NC
NC
NC
VCC
OUT
GND
GND
SPEEDOUT
SPEEDIN
+12V
R1
1k
R2
10K
R14
1.6M
R16
.33uF
C1
.01uF
C2

ID	Description	Part Number	Quantity
R1, R3, R4, R7, R9, R10	600 ohm resistor		6
R2, R5, R6, R8, R11, R12	10k resistor		6
R13, R15, R17, R19	330 ohm resistor		4
R14	20k variable resistor		1
R16	1.6M resistor		1
R21	1k resistor		1
C1	0.33uF electrolytic capacitor		1
C2	0.01uF capacitor		1
U1, U2, U3, U4, U6, U8, U9, U10, U11, U12	Optoisolator	4N35	10
U18	5V voltage regulator	LM7805	1
U5	Adapter/amplifier	LM1815	1
D1, D2, D3, D4	Diode	1N4007	4
J1, J2, J3, J4, J5	0.1” header pins		24 pins
J8	0.156” 4-pin header	MTA-156	1

Table B.1: Bill of materials for printed circuit board

Appendix C

MATLAB Code Listing

Appendix C contains a code listing of the most important MATLAB script files used with the test stand.

`PumpStandCondenseData_PumpCCW.m` performs all of the data reduction described in Section 4.1. `PumpStandCalculateDisplacement.m` calculates the derived volume as explained in Section 4.2.

`PumpStandModelPumpCCWFlow.m` and `PSPumpFlowModel.m` are used together to fit the model to the data. `PSPumpFlowModel.m` contains the equations for the model. `PumpStandModelPumpCCWFlow.m` formats the data and runs the optimization.

```
  1 function [ptdataCond, ptdataCond500, ptdataCond500Avg, ptdataCalc] =↙
PumpStandCondenseData_PumpCCW(ptdata)
  2
  3 pumpmode = 1; % 1 for pump mode, 0 for motor  (affects calculations.)
  4
  5 %% PumpStandCondenseData
  6
  7 % David Grandall, April 17, 2009; heavily modded 5-6-09
  8
  9 % This program separates the useful data (savecmd not equal to zero) from
 10 % the input ptdata struct and outputs it as ptdataCond
 11
 12 % The second half of the program goes through ptdataCond struct and
 13 % pulls the first 5 seconds (first 500 data points) out of each test
 14 % outputs as ptdataCond500
 15
 16
 17 %% Determine length of ptdata, how many "useful" points exist
 18 ptdataLength = length(ptdata);
 19 ptdataUseful = 0;
 20 for i= 1:ptdataLength
 21     if ptdata(i).savecmd ~=0
 22         ptdataUseful = ptdataUseful + 1;
 23     end
 24 end
 25
 26 %% Define new struct with existing fieldnames (Justin Lapp)
 27 % save names of all the fields of PdataMasterPump to names
 28 names=fieldnames(ptdata);
 29 % Define blank string
 30 nstring=[];
 31 % Put existing nstring 'next fieldname', 0,  populating nstring with the
 32 % argument for =struct(' nstring........)
 33 for i = 1:length(names)
 34     nstring=[nstring '''' char(names(i)) ''',0,'];
 35 end
 36 eval(['ptdataCond(' num2str(1) ':' num2str(ptdataUseful) ')=struct(' nstring(1:↙
length(nstring)-1) ');']);
 37
 38 %% Populate PdataCondPump with datapoints with savecmd not equal to zero.
 39 i = 1; %#ok<NASGU> % i is the index running through ptdata
 40 j = 1; % j is the index running through ptdataCond
 41
 42 for i = 1:ptdataLength
 43     if ptdata(i).savecmd ~= 0
 44         ptdataCond(j).speed = ptdata(i).speed;
 45         ptdataCond(j).flowhigh = ptdata(i).flowhigh;
 46         ptdataCond(j).flowcase = ptdata(i).flowcase;
 47         ptdataCond(j).flowcharge = ptdata(i).flowcharge;
 48         ptdataCond(j).flowlow = ptdata(i).flowlow;
 49         ptdataCond(j).time = ptdata(i).time;
```

```
 50          ptdataCond(j).p1disp = ptdata(i).p1disp;
 51          ptdataCond(j).p2disp = ptdata(i).p2disp;
 52          ptdataCond(j).p3disp = ptdata(i).p3disp;
 53          ptdataCond(j).p4disp = ptdata(i).p4disp;
 54          ptdataCond(j).presshi = ptdata(i).presshi;
 55          ptdataCond(j).presscase = ptdata(i).presscase;
 56          ptdataCond(j).presslow = ptdata(i).presslow;
 57          ptdataCond(j).presshiwall = ptdata(i).presshiwall;
 58          ptdataCond(j).presscharge = ptdata(i).presscharge;
 59          ptdataCond(j).torque = ptdata(i).torque;
 60          ptdataCond(j).tempa = ptdata(i).tempa;
 61          ptdataCond(j).tempcase = ptdata(i).tempcase;
 62          ptdataCond(j).p1dir = ptdata(i).p1dir;
 63          ptdataCond(j).savecmd = ptdata(i).savecmd;
 64          ptdataCond(j).refpress = ptdata(i).refpress;
 65          ptdataCond(j).refspeed = ptdata(i).refspeed;
 66          ptdataCond(j).refp1cmd = ptdata(i).refp1cmd;
 67          ptdataCond(j).flowhicounts = ptdata(i).flowhicounts;
 68          ptdataCond(j).flowcasecounts = ptdata(i).flowcasecounts;
 69          ptdataCond(j).flowchargecounts = ptdata(i).flowchargecounts;
 70          ptdataCond(j).flowlowcounts = ptdata(i).flowlowcounts;
 71          ptdataCond(j).datetime = ptdata(i).datetime;
 72          j = j+1;
 73      end
 74 end
 75
 76
 77 %% This half of the program extracts the first 5 seconds (500 data points)
 78 %% out of each test and puts the result in ptdataCond
 79 % The ptdataCond struct still contains the data from the first half of
 80 % the program.  There are no "empty" data points in PdataCondPump; all
 81 % contain data.
 82
 83 %% Define new struct (ptdataCond500) with existing fieldnames (Justin Lapp), same↙
length as
 84 %% ptdataCond
 85
 86 %Find the number of data poitns required for the new ptdataCond500
 87 %struct.
 88 k = 500; %The first point of ptdataCond "counts" but will not be caught by the↙
algorithm.
 89
 90 for i = 2:(ptdataUseful-150)
 91     %%If the set point values for two consecutive data points are
 92     %%different, treat the second (i) as a new "test point".  Also, check
 93     %%100 points in to make sure it's not a hoax
 94     if (((ptdataCond(i).refpress ~= ptdataCond(i-1).refpress) || ...
 95             (ptdataCond(i).refspeed ~= ptdataCond(i-1).refspeed)) || ...
 96             ((ptdataCond(i).refp1cmd ~= ptdataCond(i-1).refp1cmd))) && ...
 97             (ptdataCond(i+100).refspeed == ptdataCond(i).refspeed)
 98         %Copy 500 values
```

```
 99         k = k + 500;
100     end
101 end
102 
103 ptdataCond500Length = k;
104 
105 %% save names of all the fields of ptdataCond to names
106 names=fieldnames(ptdataCond);
107 % Define blank string
108 nstring=[];
109 % Put existing nstring 'next fieldname', 0,  populating nstring with the
110 % argument for =struct(' nstring........)
111 for i = 1:length(names)
112     nstring=[nstring '''' char(names(i)) ''',0,'];
113 end
114 eval(['ptdataCond500(' num2str(1) ':' num2str(ptdataCond500Length) ')=struct('↙
nstring(1:length(nstring)-1) ');']);
115 
116 %% Go through ptdataCond to find the beginning of each test section, copy
117 %the first 500 data points to ptdataCond500.  The first "test point" in
118 %PdataCondPump is valid, that's why there's a for loop before the main
119 %loop.
120 k = 1;
121 for i=1:500
122     ptdataCond500(k).speed = ptdataCond(i).speed; %#ok<*AGROW>
123     ptdataCond500(k).flowhigh = ptdataCond(i).flowhigh;
124     ptdataCond500(k).flowcase = ptdataCond(i).flowcase;
125     ptdataCond500(k).flowcharge = ptdataCond(i).flowcharge;
126     ptdataCond500(k).flowlow = ptdataCond(i).flowlow;
127     ptdataCond500(k).time = ptdataCond(i).time;
128     ptdataCond500(k).p1disp = ptdataCond(i).p1disp;
129     ptdataCond500(k).p2disp = ptdataCond(i).p2disp;
130     ptdataCond500(k).p3disp = ptdataCond(i).p3disp;
131     ptdataCond500(k).p4disp = ptdataCond(i).p4disp;
132     ptdataCond500(k).presshi = ptdataCond(i).presshi;
133     ptdataCond500(k).presscase = ptdataCond(i).presscase;
134     ptdataCond500(k).presslow = ptdataCond(i).presslow;
135     ptdataCond500(k).presshiwall = ptdataCond(i).presshiwall;
136     ptdataCond500(k).presscharge = ptdataCond(i).presscharge;
137     ptdataCond500(k).torque = ptdataCond(i).torque;
138     ptdataCond500(k).tempa = ptdataCond(i).tempa;
139     ptdataCond500(k).tempcase = ptdataCond(i).tempcase;
140     ptdataCond500(k).p1dir = ptdataCond(i).p1dir;
141     ptdataCond500(k).savecmd = ptdataCond(i).savecmd;
142     ptdataCond500(k).refpress = ptdataCond(i).refpress;
143     ptdataCond500(k).refspeed = ptdataCond(i).refspeed;
144     ptdataCond500(k).refp1cmd = ptdataCond(i).refp1cmd;
145     ptdataCond500(k).flowhicounts = ptdataCond(i).flowhicounts;
146     ptdataCond500(k).flowcasecounts = ptdataCond(i).flowcasecounts;
147     ptdataCond500(k).flowchargecounts = ptdataCond(i).flowchargecounts;
148     ptdataCond500(k).flowlowcounts = ptdataCond(i).flowlowcounts;
```

```
149     ptdataCond500(k).datetime = ptdataCond(i).datetime;
150     k = k+1;
151 end
152
153 i = 2; %#ok<NASGU> %Goes through PdataCondPump incrementally to find new test poitns
154
155 for i = 2:(ptdataUseful-150)
156     %% If the set point values for two consecutive data points are
157     %% different, treat the second (i) as a new "test point".  Also, check
158     %% 100 points in to make sure it's not a hoax
159     if (((ptdataCond(i).refpress ~= ptdataCond(i-1).refpress) || ...
160             (ptdataCond(i).refspeed ~= ptdataCond(i-1).refspeed)) || ...
161             ((ptdataCond(i).refp1cmd ~= ptdataCond(i-1).refp1cmd))) && ...
162             (ptdataCond(i+100).refspeed == ptdataCond(i).refspeed)
163         %Copy 500 values
164         for j=i:(i+500-1)
165             ptdataCond500(k).speed = ptdataCond(j).speed;
166             ptdataCond500(k).flowhigh = ptdataCond(j).flowhigh;
167             ptdataCond500(k).flowcase = ptdataCond(j).flowcase;
168             ptdataCond500(k).flowcharge = ptdataCond(j).flowcharge;
169             ptdataCond500(k).flowlow = ptdataCond(j).flowlow;
170             ptdataCond500(k).time = ptdataCond(j).time;
171             ptdataCond500(k).p1disp = ptdataCond(j).p1disp;
172             ptdataCond500(k).p2disp = ptdataCond(j).p2disp;
173             ptdataCond500(k).p3disp = ptdataCond(j).p3disp;
174             ptdataCond500(k).p4disp = ptdataCond(j).p4disp;
175             ptdataCond500(k).presshi = ptdataCond(j).presshi;
176             ptdataCond500(k).presscase = ptdataCond(j).presscase;
177             ptdataCond500(k).presslow = ptdataCond(j).presslow;
178             ptdataCond500(k).presshiwall = ptdataCond(j).presshiwall;
179             ptdataCond500(k).presscharge = ptdataCond(j).presscharge;
180             ptdataCond500(k).torque = ptdataCond(j).torque;
181             ptdataCond500(k).tempa = ptdataCond(j).tempa;
182             ptdataCond500(k).tempcase = ptdataCond(j).tempcase;
183             ptdataCond500(k).p1dir = ptdataCond(j).p1dir;
184             ptdataCond500(k).savecmd = ptdataCond(j).savecmd;
185             ptdataCond500(k).refpress = ptdataCond(j).refpress;
186             ptdataCond500(k).refspeed = ptdataCond(j).refspeed;
187             ptdataCond500(k).refp1cmd = ptdataCond(j).refp1cmd;
188             ptdataCond500(k).flowhicounts = ptdataCond(j).flowhicounts;
189             ptdataCond500(k).flowcasecounts = ptdataCond(j).flowcasecounts;
190             ptdataCond500(k).flowchargecounts = ptdataCond(j).flowchargecounts;
191             ptdataCond500(k).flowlowcounts = ptdataCond(j).flowlowcounts;
192             ptdataCond500(k).datetime = ptdataCond(j).datetime;
193             k = k+1;
194         end
195     end
196 end
197
198 %% Save PdataCondPump
199 %
```

```
200 % filename = 'PdataExt500Pump.mat';
201 % cd('Pump_Stand_Results')
202 % save(filename, 'PdataExt500Pump', '-v6');
203 % cd ..
204
205 clear ptdataLength ptdataUseful ptdata i j k names nstring
206
207 %% This section of code takes an average over each 500 set of points.
208 %Find the number of test points (500 data points per test point)
209 numtestpoints = fix(ptdataCond500Length / 500);  %rounds down to integer.
210
211 %% Create new struct, ptdataCond500Avg
212 names=fieldnames(ptdataCond);
213 % Define blank string
214 nstring=[];
215 % Put existing nstring 'next fieldname', 0,  populating nstring with the
216 % argument for =struct(' nstring........)
217 for i = 1:length(names)
218     nstring=[nstring '''' char(names(i)) ''',0,'];
219 end
220 eval(['ptdataCond500Avg(' num2str(1) ':' num2str(numtestpoints) ')=struct(' nstring↙
(1:length(nstring)-1) ');']);
221
222 %Take averages, save to PdataExt500AvgPump
223 for i = 1:numtestpoints
224     ptdataCond500Avg(i).speed = mean([ptdataCond500(((i-1)*500+1):i*500).speed]);
225     ptdataCond500Avg(i).flowhigh = mean([ptdataCond500(((i-1)*500+1):i*500).↙
flowhigh]);
226     ptdataCond500Avg(i).flowcase = mean([ptdataCond500(((i-1)*500+1):i*500).↙
flowcase]);
227     ptdataCond500Avg(i).flowcharge = mean([ptdataCond500(((i-1)*500+1):i*500).↙
flowcharge]);
228     ptdataCond500Avg(i).flowlow = mean([ptdataCond500(((i-1)*500+1):i*500).↙
flowlow]);
229     ptdataCond500Avg(i).time = mean([ptdataCond500(((i-1)*500+1):i*500).time]);
230     ptdataCond500Avg(i).p1disp = mean([ptdataCond500(((i-1)*500+1):i*500).p1disp]);
231     if abs(ptdataCond500Avg(i).p1disp) < 0.2;
232         ptdatasign = sign(ptdataCond500Avg(i).p1disp);
233         ptdataCond500Avg(i).p1disp = ptdatasign*0.2;
234     end
235     if abs(ptdataCond500Avg(i).p1disp) > 0.8;
236         ptdatasign = sign(ptdataCond500Avg(i).p1disp);
237         ptdataCond500Avg(i).p1disp = ptdatasign*0.8;
238     end
239     ptdataCond500Avg(i).p2disp = mean([ptdataCond500(((i-1)*500+1):i*500).p2disp]);
240     ptdataCond500Avg(i).p3disp = mean([ptdataCond500(((i-1)*500+1):i*500).p3disp]);
241     ptdataCond500Avg(i).p4disp = mean([ptdataCond500(((i-1)*500+1):i*500).p4disp]);
242     ptdataCond500Avg(i).presshi = mean([ptdataCond500(((i-1)*500+1):i*500).↙
presshi]);
243     ptdataCond500Avg(i).presscase = mean([ptdataCond500(((i-1)*500+1):i*500).↙
presscase]);
```

```
244     ptdataCond500Avg(i).presslow = mean([ptdataCond500(((i-1)*500+1):i*500).↙
presslow]);
245     ptdataCond500Avg(i).presshiwall = mean([ptdataCond500(((i-1)*500+1):i*500).↙
presshiwall]);
246     ptdataCond500Avg(i).presscharge = mean([ptdataCond500(((i-1)*500+1):i*500).↙
presscharge]);
247     ptdataCond500Avg(i).torque = mean([ptdataCond500(((i-1)*500+1):i*500).torque]);
248     ptdataCond500Avg(i).tempa = mean([ptdataCond500(((i-1)*500+1):i*500).tempa]);
249     ptdataCond500Avg(i).tempcase = mean([ptdataCond500(((i-1)*500+1):i*500).↙
tempcase]);
250     ptdataCond500Avg(i).p1dir = mean([ptdataCond500(((i-1)*500+1):i*500).p1dir]);
251     ptdataCond500Avg(i).savecmd = mean([ptdataCond500(((i-1)*500+1):i*500).↙
savecmd]);
252     ptdataCond500Avg(i).refpress = mean([ptdataCond500(((i-1)*500+1):i*500).↙
refpress]);
253     ptdataCond500Avg(i).refspeed = mean([ptdataCond500(((i-1)*500+1):i*500).↙
refspeed]);
254     ptdataCond500Avg(i).refp1cmd = mean([ptdataCond500(((i-1)*500+1):i*500).↙
refp1cmd]);
255     %flowcasecounts, flowlowcounts, are all "negative".
256     ptdataCond500Avg(i).flowhicounts = (1)*(ptdataCond500(i*500).flowhicounts) - (1)↙
*(ptdataCond500((i-1)*500+1).flowhicounts); %Positive flow comes out A
257     ptdataCond500Avg(i).flowcasecounts = (-1)*(ptdataCond500(i*500).flowcasecounts)↙
- (-1)*(ptdataCond500((i-1)*500+1).flowcasecounts); %Positive flow comes out case drain
258     ptdataCond500Avg(i).flowchargecounts = (ptdataCond500(i*500).flowchargecounts) -↙
(ptdataCond500((i-1)*500+1).flowchargecounts); %Positive flow goes into charge
259     ptdataCond500Avg(i).flowlowcounts = (-1)*(ptdataCond500(i*500).flowlowcounts) -↙
(-1)*(ptdataCond500((i-1)*500+1).flowlowcounts); %Positive flow goes into B
260     ptdataCond500Avg(i).datetime = ptdataCond500((i-1)*500+1).datetime;
261
262 end
263
264 %% Save PdataExt500AvgPump
265
266 % filename = 'PdataExt500AvgPump.mat';
267 % cd('Pump_Stand_Results')
268 % save(filename, 'PdataExt500AvgPump', '-v6');
269 % cd ..
270
271 clear numtestpoints nstring names i filename ptdataCond500Length
272
273 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
274
275 %% PumpStandCalculateDataPump
276 %
277 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
278 % This program calculates the deltaP, flowrate, losses, efficiency etc... for each
279 % point in ptdataCond500Avg.  The data is stored the struct ptdataCalc
280
281
282 %% Create yet another struct, PdataCalcPump
```

```
283 ptdataCalc = ptdataCond500Avg;
284
285 %% Make and Calculate New columns for different units
286 [junk calclength] = size(ptdataCalc);
287 %calclength is the number of test points in ptdataCalc.
288
289 % A serious attempt is made here to make fields that are in appropriate SI
290 % units with those in psi at the end of the name
291
292 %The default short field name is in the appropriate SI.  flowhigh is in lpm
293 % S-D Series 42 28cc pump is 28.00 cc/rev
294
295 for i = 1:calclength
296     % Flows, use flowxxxxxcounts to calculate flow in gpm & lpm
297     ptdataCalc(i).flowhighgpm = (ptdataCalc(i).flowhicounts/1715.608/5*60);
298     ptdataCalc(i).flowhigh = (ptdataCalc(i).flowhicounts/453/5*60); %q_vp2e (pump),↙
qvm1e(motor)
299     ptdataCalc(i).flowcasegpm = (ptdataCalc(i).flowcasecounts/6465.651/5*60);
300     ptdataCalc(i).flowcase = (ptdataCalc(i).flowcasecounts/1708/5*60); %q_vd
301     ptdataCalc(i).flowchargegpm = (ptdataCalc(i).flowchargecounts/16118.63/5*60);
302     ptdataCalc(i).flowcharge = (ptdataCalc(i).flowchargecounts/4259/5*60);
303     ptdataCalc(i).flowlowgpm = (ptdataCalc(i).flowlowcounts/1723.651/5*60);
304     ptdataCalc(i).flowlow = (ptdataCalc(i).flowlowcounts/455/5*60);
305     % Pressures, create deltaP, change units.  Base unit is bar
306     ptdataCalc(i).deltaPpsi = ptdataCalc(i).presshi - ptdataCalc(i).presslow;
307     ptdataCalc(i).deltaP = ptdataCalc(i).deltaPpsi * 0.0689475728;
308     ptdataCalc(i).presshipsi = ptdataCalc(i).presshi;
309     ptdataCalc(i).presshi = ptdataCalc(i).presshi * 0.0689475728;
310     ptdataCalc(i).presscasepsi = ptdataCalc(i).presscase;
311     ptdataCalc(i).presscase = ptdataCalc(i).presscase * 0.0689475728;
312     ptdataCalc(i).presslowpsi = ptdataCalc(i).presslow;
313     ptdataCalc(i).presslow = ptdataCalc(i).presslow * 0.0689475728;
314     ptdataCalc(i).presshiwallpsi = ptdataCalc(i).presshiwall;
315     ptdataCalc(i).presshiwall = ptdataCalc(i).presshiwall * 0.0689475728;
316     ptdataCalc(i).presschargepsi = ptdataCalc(i).presscharge;
317     ptdataCalc(i).presscharge = ptdataCalc(i).presscharge * 0.0689475728;
318     ptdataCalc(i).refpresspsi = ptdataCalc(i).refpress;
319     ptdataCalc(i).refpress = ptdataCalc(i).refpress * 0.0689475728;
320
321 end
322
323
324 if pumpmode
325
326     %% Calculation of efficiency, etc...WRITTEN FOR PUMP MODE
327     for i = 1:calclength
328         %Power/losses in kW
329         ptdataCalc(i).mechpower = 2 * pi * ptdataCalc(i).speed * ptdataCalc(i).↙
torque / 60000;
330         ptdataCalc(i).hydpower = ptdataCalc(i).flowhigh * ptdataCalc(i).presshi /↙
600;
```

```
331         ptdataCalc(i).hydpowereffective = (ptdataCalc(i).flowhigh * ptdataCalc(i).↙
presshi) / 600 - (ptdataCalc(i).flowlow * ptdataCalc(i).presslow) / 600;
332
333         % Overall, LOSSES in KW
334         ptdataCalc(i).effoverall = ...
335             (((ptdataCalc(i).flowhigh/1000 * ptdataCalc(i).presshi * 100000) -↙
(ptdataCalc(i).flowlow/1000 * ptdataCalc(i).presslow * 100000)) ...
336             / (2 * pi * ptdataCalc(i).speed * ptdataCalc(i).torque));
337         ptdataCalc(i).lossoverall = ...
338             ((2 * pi * ptdataCalc(i).speed/60 * ptdataCalc(i).torque) - ...
339             ((ptdataCalc(i).flowhigh/1000/60 * ptdataCalc(i).presshi * 100000) -↙
(ptdataCalc(i).flowlow/1000/60 * ptdataCalc(i).presslow * 100000)))/1000;
340
341         % Volumetric
342         ptdataCalc(i).effvol = (ptdataCalc(i).flowhigh/60) / (1.666*(abs(ptdataCalc↙
(i).refp1cmd) - .2) * 28/1000 * ptdataCalc(i).speed/60);
343         ptdataCalc(i).lossvol = ((1.666*(abs(ptdataCalc(i).refp1cmd)-.2) * 28/1000 *↙
ptdataCalc(i).speed) - ptdataCalc(i).flowhigh)  ...
344             / 60 / 1000 * (ptdataCalc(i).presshi * 100000) / 1000;
345
346         % Mechanical
347         ptdataCalc(i).effmech = ptdataCalc(i).effoverall / ptdataCalc(i).effvol;
348         ptdataCalc(i).lossmech = ptdataCalc(i).lossoverall - ptdataCalc(i).lossvol;
349
350         % Direction
351         ptdataCalc(i).p1dir = round(ptdataCalc(i).p1dir);
352
353     end
354
355 end
356
357 % %% Save PdataCalcPump
358 % filename = 'PumpStandCalculateDataPump.mat';
359 % cd('Pump_Stand_Results')
360 % save(filename, 'PdataCalcPump', '-v6');
361 % cd ..
```

```
% PumpStandCalculateDisplacement

% David Grandall, June 22 2009

% This program helps find the actual displacement (by linear extrapolation)
% of each of the displacements of the pumps.  Info goes into the struct
% field "derdisp".


%% Find the test points at a certain displacement & speed
% Variable under "test":

switchstring = 'S42motorCW';
ptdata = PumpStandLoadFile(switchstring);
derdispstruct = PumpStandLoadFile('derdispstruct');

%Define speed & displacement vectors to run through
speedvector = [250 500 1000 1500 2000 2500 3000 3500 4000];
displacementvector = [0 0.1 0.2 0.3 0.4 0.5 0.65 0.8 1];

for j = 1:length(displacementvector)
    for k = 1:length(speedvector)
        plotdisplacement = displacementvector(j);
        plotspeed = speedvector(k);

        %plotdisplacement = .65;
        %plotspeed = 1000;
        fulldisplacement = 28; %cc/rev

        plotdispup = plotdisplacement + 0.01; %uppder bound
        plotdisplow = plotdisplacement - 0.01; %lower bound

        plotspeedup = plotspeed + 10; %upper bound
        plotspeedlow = plotspeed - 10; %lower bound
        %
        refp1disp = abs([ptdata.refp1disp]);
        refspeed = abs([ptdata.refspeed]);
        flowhigh = [ptdata.flowhigh];
        speed = abs([ptdata.speed]);
        deltaP = [ptdata.deltaP];

        %ind = find((p1disp > plotdisplow & p1disp < plotdispup));
        ind = find((refp1disp > plotdisplow & refp1disp < plotdispup) & ...
            (refspeed > plotspeedlow & refspeed < plotspeedup));
        flowvect = 0;
        speedvect = 0;
        deltaPvect = 0;

        for i = 1:length(ind)
            flowvect(i) = flowhigh(ind(i));
            speedvect(i) = speed(ind(i));
```

```
 52              deltaPvect(i) = [ptdata(ind(i)).deltaP];
 53          end
 54
 55          % flowhigh (lpm) to cc/rev:
 56          flowvect = flowvect / 60; %now liters / sec
 57          flowvect = flowvect * 1000; %now cc/sec
 58          flowvect = flowvect ./(speedvect./60);
 59
 60          % Polynomial fit
 61          % [p,S] = polyfit(x,y,n)
 62          [polydata,S] = polyfit(deltaPvect,flowvect,1);
 63          %Plot polynomial fit
 64          if 1
 65
 66              figure;
 67              hold on
 68              X = 0:1:200;
 69              Y = polyval(polydata,X);
 70              plot(X,Y,'-b')
 71
 72              plot(deltaPvect,flowvect,'b.')
 73              xlabel('deltaP (bar)')
 74              ylabel('Effective Displacement (cc/rev)')
 75              title(['Determining Effective Displacement, ' num2str(plotdisplacement)↙
' des disp, ' num2str(plotspeed) 'RPM ' switchstring])
 76              xlim([0 200]);
 77              ylim([-10 30]);
 78              interceptstring = num2str(polydata(2));
 79              text(5,-5,['intercept (cc):  ' interceptstring]);
 80              interceptstring = num2str(plotdisplacement);
 81              text(5,-7.5,['command displacement (.2-.8): ' interceptstring]);
 82
 83              hold off
 84          end
 85
 86
 87          %% Create two vectors acting as labels for the displacementmatrix
 88
 89
 90          %% Save intercept data to data points involved
 91
 92          % for i = 1:length(ind)
 93          %    ptdata(ind(i)).derdisp = polydata(2)/28;
 94          %end
 95
 96          % find where to put in the value in derdispmatrix
 97          indspeed = find(speedvector == plotspeed);
 98          inddisp = find(displacementvector == plotdisplacement);
 99          % find whether or not displacement needs to be multiplied by negative 1
100          % Displacement coming out of above algorithm will "usually" be positive.
101          switch switchstring
```

```
102             case 'S42pumpCCW' %1st quadrant of graph
103                 dispsign = 1;
104                 derdispstruct.(switchstring).sign(inddisp,indspeed) = dispsign;
105                 derdispstruct.(switchstring).derdisp(inddisp,indspeed) = polydata(2)↙
/fulldisplacement;
106             case 'S42motorCW' %2nd quadrant of graph
107                 dispsign = 1;
108                 derdispstruct.(switchstring).sign(inddisp,indspeed) = dispsign;
109                 derdispstruct.(switchstring).derdisp(inddisp,indspeed) = polydata(2)↙
/fulldisplacement;
110             case 'S42pumpCW' %3rd quadrant of graph
111                 dispsign = -1;
112                 derdispstruct.(switchstring).sign(inddisp,indspeed) = dispsign;
113                 derdispstruct.(switchstring).derdisp(inddisp,indspeed) = polydata(2)↙
/fulldisplacement;
114             case 'S42motorCCW' %4th quadrant of graph
115                 dispsign = -1;
116                 derdispstruct.(switchstring).sign(inddisp,indspeed) = dispsign;
117                 derdispstruct.(switchstring).derdisp(inddisp,indspeed) = polydata(2)↙
/fulldisplacement;
118         end
119
120         pause
121     end
122 end
123
124 PumpStandSaveFile(switchstring,ptdata);
125 %PumpStandSaveFile('derdispstruct',derdispstruct);
126 clear S X Y deltaP deltaPvect flowhigh flowvect foldername i ind refp1disp p1disp↙
plotdisplacement plotdisplacementarchive ...
127     plotdisplow plotdispup plotspeed plotspeedlow plotspeedup refspeed speed↙
speedvect ptdata switchstring interceptstring polydata...
128     displacementvector speedvector indspeed inddisp fulldisplacement dispsign j k
```

```
%% PumpStandModelPumpCCWFlow

% This program fits a model (Dorey, 1988) to one quadrant of P/M data.  The
% MATLAB function LSQCURVEFIT is used extensively.  Only flow is modeled
% here.

% August 17, 2009

%% Which quadrant
clear
switchstring = 'S42pumpCCW';

ptdata = PumpStandLoadFile(switchstring);

%% Dorey Piston
if 0
    %% Load fields into matrix to pass to function
    % Find data points that ARE part of the main data set (derdisp does not
    % equal -2)
    ind = find([ptdata.derdisp] > -1.98);
    % Prepare fields for insertion into matrix to pass
    for i = 1:length(ind)
        speed(i) = ptdata(ind(i)).speed * (2*pi) /60; %convert from RPM to rad/s
        disp(i) = ptdata(ind(i)).derdisp;
        deltaP(i) = ptdata(ind(i)).deltaP / 10 * 1E6; %convert from bar to Pa
        visc(i) = ((ptdata(ind(i)).tempcase + ptdata(ind(i)).tempa) /2 ) * -0.000538 + 0.05949;↙
%Visc PLACEHOLDER
        flowhigh(i) = ptdata(ind(i)).flowhigh / 60 / 1000 * 1e6;   %convert fr lpm cc/sec.  Goes↙
straight to lsqcurvefit ...
        % This allows the optimization to work with values of flowhigh that are
        % at least greater than 1, instead of many decimal places.
    end

    %Set bounds for a0 (initial guess)
    abcdef_lb = -100;
    abcdef_ub = 100;
    Vr_lb = 0;
    Vr_ub = 0.2;
    n_lb = -5;
    n_ub = 5;
    %Number of initial guesses/runs
    numruns = 20;
    options = optimset('TolFun',1E-8,'TolX',1E-8);
    data = [speed; disp; deltaP; visc;]; %Each variable is a row in the matrix.  Columns are↙
points.
    i = 1;
    for i = 1:numruns
        a0 = [(abcdef_ub - abcdef_lb)*rand(2,1) + abcdef_lb; (Vr_ub - Vr_lb)*rand(1,1);];
        [a(:,i),resnorm(i),residual(i,:)] = lsqcurvefit(@PSPumpFlowModel,a0,data,flowhigh,↙
[abcdef_lb abcdef_lb 0],[abcdef_ub abcdef_ub Vr_ub],options);
        goodness(i) = sum(residual(i,:).^2);
    end

end

%% Dorey Gear
if 0
```

```
 54     %% Load fields into matrix to pass to function
 55     % Find data points that ARE part of the main data set (derdisp does not
 56     % equal -2)
 57     ind = find([ptdata.derdisp] > -1.98);
 58     % Prepare fields for insertion into matrix to pass
 59     for i = 1:length(ind)
 60         speed(i) = ptdata(ind(i)).speed * (2*pi) /60; %convert from RPM to rad/s
 61         disp(i) = ptdata(ind(i)).derdisp;
 62         deltaP(i) = ptdata(ind(i)).deltaP / 10 * 1E6; %convert from bar to Pa
 63         visc(i) = ((ptdata(ind(i)).tempcase + ptdata(ind(i)).tempa) /2 ) * -0.000538 + 0.05949;↙
%Visc PLACEHOLDER
 64         flowhigh(i) = ptdata(ind(i)).flowhigh / 60 / 1000 * 1e6;   %convert fr lpm cc/sec.  Goes↙
straight to lsqcurvefit ...
 65         % This allows the optimization to work with values of flowhigh that are
 66         % at least greater than 1, instead of many decimal places.
 67     end
 68
 69     %Set bounds for a0 (initial guess)
 70     abc_lb = -100;
 71     abc_ub = 100;
 72     Vr_lb = 0;
 73     Vr_ub = 0.2;
 74     n_lb = -5;
 75     n_ub = 5;
 76     %Number of initial guesses/runs
 77     numruns = 30;
 78     options = optimset('TolFun',1E-8,'TolX',1E-8);
 79     data = [speed; disp; deltaP; visc;]; %Each variable is a row in the matrix.  Columns are↙
points.
 80     i = 1;
 81     for i = 1:numruns
 82         a0 = [(abc_ub - abc_lb)*rand(3,1) + abc_lb; (n_ub - n_lb)*rand(1,1) + n_lb;(Vr_ub -↙
Vr_lb)*rand(1,1);];
 83         [a(:,i),resnorm(i),residual(i,:)] = lsqcurvefit(@PSPumpFlowModelGear,a0,data,flowhigh,↙
[abc_lb abc_lb abc_lb n_lb 0],[abc_ub abc_ub abc_ub n_ub Vr_ub],options);
 84         goodness(i) = sum(residual(i,:).^2);
 85     end
 86
 87 end
 88
 89
```

```
 1 function Flow = PSPumpFlowModel(avector,data)
 2 % Evaluating function for Pump Flow Model (Dorey)
 3
 4 %August 17, 2009; DRG
 5
 6 %Initializing constants
 7 D = 28E-6/(2*pi); %Convert from cc/rev to m3/rev
 8 C_s = 1.046E-8; %Slip coefficient.  Always positive, negative is taken care of with P vs M.
 9 P_low = 1723689.32; %Pa, converted from 250psi (low pressure)
10 speed_max = 4000 * (2*pi) /60; %convert fr RPM to rad/s
11 B = 1.700E9; %Bulk modulus, Pa
12 %V_r = 0.1; % Clearance volume/Total volume at x=1 (don't actually know);
13
14
15 %Pull field data from "data" matrix
16 speed = data(1,:);
17 disp = data(2,:);
18 deltaP = data(3,:);
19 visc = data(4,:);
20 a = avector(1);
21 b = avector(2);
22 V_r = avector(3);
23
24 %Actual function (has to be .* everything)
25
26 % Flow = ((speed .* D .* disp) - ...
27 %     (C_s .* (press/P_low) .* (a+b.*(speed/speed_max)).* (press .* D)./visc) - ...
28 % (press .* speed .* disp ./ B) .* (V_r + (1+disp)./2)) * 1e6;
29
30 %% Dorey Flow Model for Piston Pump "The Simple Model"
31 Flow = ((speed .* D .* disp) - ...
32      (C_s .* (deltaP/P_low) .* (a+b.*(speed/speed_max)).* (deltaP .* D)./visc) - ...
33  (deltaP .* speed .* D ./ B) .* (V_r + (1+disp)./2)) * 1e6;
34
35
```