# AN EXPERIMENTAL EXAMINATION OF A PROGRESSING CAVITY PUMP OPERATING AT VERY HIGH GAS VOLUME FRACTIONS

A Thesis

by

MICHAEL W. GLIER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2011

Major Subject: Mechanical Engineering

# AN EXPERIMENTAL EXAMINATION OF A PROGRESSING CAVITY PUMP OPERATING AT VERY HIGH GAS VOLUME FRACTIONS

A Thesis

by

MICHAEL W. GLIER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

| | |
|---|---|
| Chair of Committee, | Gerald Morrison |
| Committee Members, | Gioia Falcone |
| | Devesh Ranjan |
| Head of Department | Dennis O'Neal |

May 2011

Major Subject: Mechanical Engineering

# ABSTRACT

An Experimental Examination of a Progressing Cavity Pump Operating at Very High Gas Volume Fractions. (May 2011)

Michael W. Glier, B.S., Texas A&M University

Chair of Advisory Committee: Dr. Gerald Morrison

The progressing cavity pump is a type of positive displacement pump that is capable of moving nearly any fluid. This type of pump transports fluids in a series of discrete cavities formed by the helical geometries of its rigid rotor and elastomeric stator. With appropriate materials for the rotor and stator, this pump can move combinations of liquids, suspended solids, and gasses equally well. Because of its versatility, the progressing cavity pump is widely used in the oil industry to transport mixtures of oil, water, and sediment; this investigation was prompted by a desire to extend the use of progressing cavity pumps to wet gas pumping applications.

One of the progressing cavity pump's limitations is that the friction between the rotor and stator can generate enough heat to damage the rotor if the pump is not lubricated and cooled by the process fluid. Conventional wisdom dictates that this type of pump will overheat if it pumps only gas, with no liquid in the process fluid. If a progressing cavity pump is used to boost the output from a wet gas well, it could potentially be damaged if the well's output is too dry for an extended period of time. This project seeks to determine how a progressing cavity pump behaves when operating at gas volume fractions between 0.90 and 0.98.

A progressing cavity pump manufactured by seepex, model no. BN 130-12, is tested at half and full speed using air-water mixtures with gas volume fractions of 0.90, 0.92, 0.94, 0.96, and 0.98. The pump's inlet and outlet conditions are controlled to produce suction pressures of 15, 30, and 45 psi and outlet pressures 0, 30, 60, 90, 120, and 150 psi higher than the inlet pressure. A series of thermocouples, pressure transducers, and turbine flow meters measures the pump's inlet and outlet conditions, the flow rates of water and air entering the pump, and pressures and temperatures at four positions within the pump's stator.

Over all test conditions, the maximum recorded temperature of the pump stator did not exceed the maximum safe rubber temperature specified by the manufacturer. The pump's flow rate is

independent of both the fluid's gas volume fraction and the pressure difference across the pump, but it increases slightly with the pump's suction pressure. The pump's mechanical load, however, is dependent only on the pressure difference across the pump and increases linearly with that parameter. Pressure measurements within the stator demonstrated that the leakage between the pump's cavities increases with the fluids gas volume fraction, indicating that liquid inside the pump improves its sealing capability. However, those same measurements failed to detect any appreciable leakage between the two pressure taps nearest the pump's inlet. This last observation suggests that the pump could be shortened by as much as 25% without losing any performance in the range of tested conditions; shortening the pump should increase its efficiency by decreasing its frictional mechanical load.

# DEDICATION

As with all my efforts, I dedicate this to the greater glory of God.

# ACKNOWLEDGEMENTS

A great many people invested their time and effort into this investigation; without their assistance this project would not yet have reached completion. I can offer them only my most sincere gratitude. First and foremost, I thank Dr. Gerald Morrison for his guidance and instruction, but most of all for the patience he displayed as I wrote this thesis. I also thank Dr. Gioia Falcone and Dr. Devesh Ranjan for lending me their time and expertise as members of my committee. I am deeply indebted to Mr. Eddie Denk of the Turbomachinery Laboratory, not only for his advice and assistance throughout the creation my test apparatus, but also for his incredible efforts to make the Turbolab such a marvelously equipped testing facility. I must also thank my friends and coworkers, Shankar Narayanan, Becky Hollkamp, and Ryan Kroupa, for both the countless hours of work they have put into this project and for putting up with my music. Finally, I thank Dr. Stuart Scott and Dr. Jun Xu at Shell for making this research possible.

# NOMENCLATURE

| | | |
|---|---|---|
| $GVF = \frac{Q_{air}}{Q}$ | — | Gas volume fraction |
| $L$ | — | Mechanical power supplied to the pump |
| $P_{air}$ | — | The pressure in the air supply line, upstream of the flow meter |
| $P_{ax}^{i}$ | — | The pressure in the pump stator at axial position *i* |
| $P_d$ | — | The pressure at the pump discharge |
| $P_s$ | — | The pressure at the pump suction |
| $Q = Q_{air} + Q_{water}$ | — | The total volumetric flow rate of the fluid entering the pump |
| $Q_{air}$ | — | The volumetric flow rate of the air at the pump suction |
| $Q_{air\ in}$ | — | The volumetric flow rate of the air passing through the flow meter |
| $Q_{water}$ | — | The volumetric flow rate of the water at the pump suction |
| $T_{air}$ | — | The temperature in the air supply line, upstream of the flow meter |
| $T_{ax}^{i}$ | — | The temperature in the pump stator at axial position *i* |
| $T_d$ | — | The temperature at the pump discharge |
| $T_s$ | — | The temperature at the pump suction |
| $\Delta P$ | — | The difference between $P_d$ and $P_s$ |
| $\eta$ | — | The nominal efficiency of the pump |
| $\omega$ | — | The rotational speed of the pump |

# TABLE OF CONTENTS

Page

Page

# LIST OF FIGURES

Page

Page

# LIST OF TABLES

Page

# INTRODUCTION

The progressing cavity pump (PCP) is a type of positive displacement pump capable of moving nearly any fluid. Liquids of nearly any viscosity, liquid-gas mixtures, and even liquids with large solid particles in suspension can all be pumped equally well with a PCP [1]. Designed by the French engineer René Moineau in the 1930s, the progressing cavity pump is built around the interaction of a helical metal stator with a solid rubber stator formed into a double internal helix. The geometry of the rotor and stator creates cavities that are completely sealed by the stator pressing against the rotor, as seen in Figure 1. As the rotor turns, the cavities within the pump move down its length, carrying fluid from the suction to the discharge.

FIGURE 1: CUTAWAY VIEW OF A TYPICAL PROGRESSING CAVITY PUMP'S ROTOR AND STATOR

PCPs have been used for oil production since the 1980s [1]; they are more commonly used for pumping abrasive mixtures of oil or water and sediment than for liquid-gas mixtures. Nevertheless, there have been successful implementations of PCPs in liquid-gas multiphase applications. In 1987, Robbins & Myers installed a PCP to boost the multiphase output of three

This thesis follows the model of the Journal of Fluids Engineering.

wells [2]. A subsequent installation was less successful and prompted Robbins & Myers to initiate a formal test program to measure the performance of one of their Series 2000 pumps with gas volume fractions in excess of 0.80. According to Mirza and Wild, these tests determined that the maximum stator temperature is "a function of pump speed, gas volume fraction, discharge pressure, and the pressure ratio (inlet to outlet pressure)." With the insight gained from these tests, a progressing cavity pump was modified to operate reliably at gas volume fractions as high as 0.99.

Moineau, Vetter, Paluchowski, Robello, Saveth, Gamboa, Olivet, and Espin have all worked to develop models for PCP performance [3-6]. Moineau proposed using the Hagen-Poisuelle equation to model the pump's internal slip and thereby estimate the pump's delivered pressure [3]. This model assumes laminar, viscous, and incompressible flow, but as the Hagen-Poisselle equation governs fluid behavior in a cylindrical pipe, Moineau's model can only be considered a first order approximation. Much later, Vetter and Paluchowski developed a model to predict a PCP's net positive suction head [4] while Robello and Saveth used incompressible flow analysis to derive equations for a pump performance based on its geometry [6]. Gamboa, Olivet, and Espin developled a model for the pump's slip assuming constant, rectangular, internal clearances, as well as laminar, viscous, incompressible flow [5]. Their experiments demonstrated that their model is only appropriate for PCPs with a non-deformable (*e.g.* steel or bronze) stator and high viscosity process fluids [5]. All these analyses apply only to liquid flows, not the two-phase flows of this investigation.

Most recently, in 2005, Bratu of PCM Pompes performed an extensive analysis of an industrial PCP running liquid-gas flows and developed a model that correlated well to his experimental results [7]. Bratu's model accounts for the gas compression within the pump, the deformation of the elastomeric stator due to pressure difference between the pump's cavities, and the friction caused by the deformation. Not only does this model correlate with the pressure distribution inside the pump, but it also predicts the internal temperature. Bratu's experiments did not examine any *GVF*s above 0.90.

This project was commissioned by Shell and seepex to help determine the suitability of a PCP for wet gas pumping applications. The production rate and lifespan of a wet gas well can be significantly increased by using a multiphase pump. The pressure boost provided by wet gas pumps can help transport the output of several wells to a common collection facility where

liquids and gasses can be split in a single large separator. Moreover, a multiphase pump attached to a well head can reduce the pressure in the well bore producing both higher flow and higher profits from the well. Indeed, if the bore pressure in a "dead" well is lowered enough, a wet gas pump could even return the well to production.

All these advantages can also be achieved by first separating the multiphase well output into liquids and gasses and boosting the respective fluids' pressure with pumps and compressors. However, a single multiphase pump may be less expensive than such a collection of machinery and a single transport pipeline is certainly less expensive than two parallel pipelines, one for each phase. The smaller footprint of a single multiphase pump is particularly desirable in sea floor applications where space is limited and expensive. Employing a single machine in place of several can reduce both cost and the risk of a mechanical failure.

A PCP is a reasonable choice to pump the multiphase flows encountered in wet gas production, but only a series of tests with multiphase flows can determine how well a PCP will perform when pumping wet gas. The possibility that a PCP might overheat if a well's output becomes too dry is a particular concern for this application.

Since a PCP's rotor and stator rub continuously, they must be always be lubricated and cooled when operating the pump. In a PCP, the process fluid is used to lubricate and cool the pump; running only gas through such a pump can generate enough heat to destroy the rubber stator. Because the liquid-gas ratio is generally not controlled in wet gas pumping applications, a wet gas pump might need to pump nearly dry gases for an extended period.

To better understand the likely behavior of a PCP operating in such conditions, a PCP manufactured by seepex is tested pumping air-water mixtures of gas volume fractions ranging from 0.90 to 0.98. The pump is operated with its suction pressure held at 15, 30, and 45 psi. Its average discharge pressure is also controlled to generate a pressure difference across the pump ranging from 0 to 150 psi. At each test condition, thermocouples, pressure transducers, and flow meters measure the properties of the process fluid as it enters and exits the pump. Additionally, thermocouples and pressure transducers installed in the pump's stator sample the temperature and pressure of the fluid within a cavity as it traveled along the length of the pump.

This document presents the results of this investigation and the conclusions that can be drawn from it. To begin, the experimental apparatus is described, including both the systems for supplying air and water to the pump and the equipment used to measure and record test data. This is followed by an account of the procedure used to evaluate the performance of the pump and an explanation of how the test data were processed. Next, the results of the tests are presented; these include the pump's flow rate, the product of its flow rate and the difference between the discharge and suction pressure, the maximum recorded temperatures in the stator, the pressures recorded along the length of the stator, the mechanical power used by the pump, and the pump's nominal efficiency. These results are examined and their implications are discussed. Finally, the conclusions of the investigation are summarized and some areas for further investigation are suggested.

# EXPERIMENTAL APPARATUS

The test platform for this investigation is a complex system that is best explained as a composition of eight subsystems: the progressing cavity pump (PCP), the variable frequency drive (VFD), the data acquisition system (DAQ), the air supply, the air control, the water supply, the water control, and the discharge control. Figure 2 provides a system level overview of the test platform, showing all subsystems and the connections between them. The heart of the test bed is the PCP itself. The VFD powers the PCP and sends a signal proportional to the power used to the DAQ. The Turbomachinery Laboratory's compressed air supply provides air for the pump, while the air control system regulates the air pressure entering the PCP and sends information on the air temperature, pressure, and flow rate to the DAQ. Water is recirculated, flowing from the water supply system to the water control system. The water mixes with air in the PCP, which forces the mixture through the discharge control system. The mixture is piped from the discharge control back to the water supply system where the air and water separate. The water control system sends temperature and flow rate data to the DAQ; the discharge control system sends temperature and pressure data. The PCP is equipped with several temperature and pressure sensors that send data to the DAQ during testing. The DAQ not only collects all the incoming information but also sends control signals to the flow-control valves in the air, water, and discharge control systems.

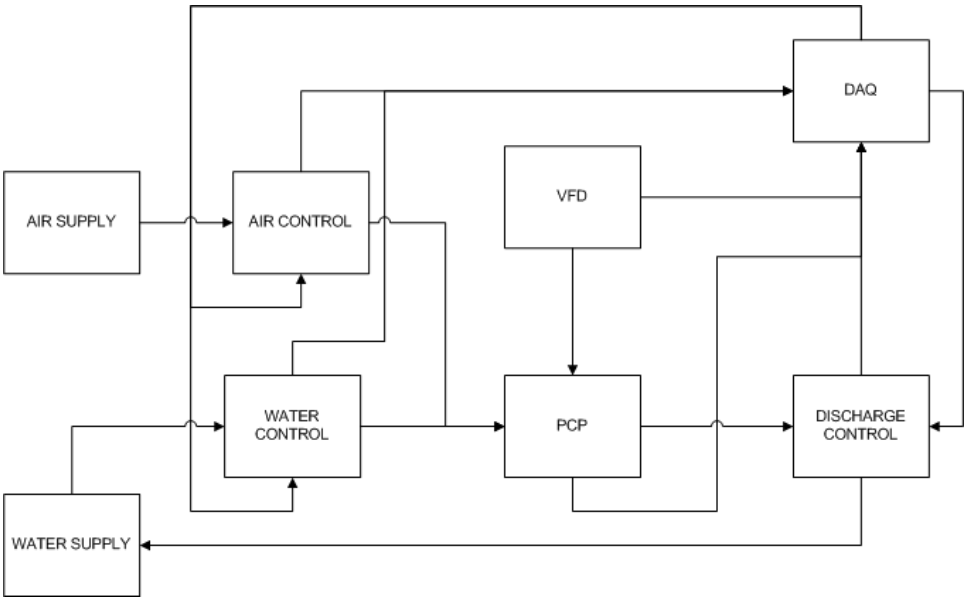


FIGURE 2: SYSTEM LEVEL SCHEMATIC OF THE TEST PLATFORM

This section presents each of these subsystem in detail, beginning with the PCP itself, then the air and water supply systems, the air and water control systems, the discharge control, the VFD, and finally the instrumentation and DAQ. Schematics of the subsystems are provided, as well as photographs of some equipment and the specifications for all equipment and instrumentation.

## PROGRESSING CAVITY PUMP

The pump under investigation is a seepex progressing cavity pump, model no. BN 130-12. This pump is a stand-alone unit with motor, gearbox, and pump assembled as one unit, as seen in Figure 3. The motor on the pump is a three-phase, 100 hp induction motor made by Siemens, model no. 1LG4 280-4AA69-Z. It is rated to draw 132 amps at 460 volts and is wired in a delta configuration. The rated speed for this motor is 1785 rpm and the power factor is 0.86. The motor is directly coupled to a gearbox that gives a 5 to 68 gear reduction.

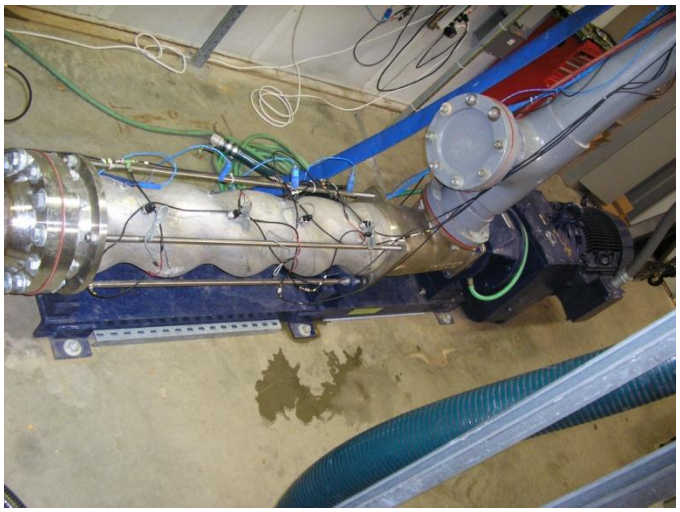

FIGURE 3: PHOTOGRAPH OF THE SEEPEX PCP, MODEL BN130-12, WITH ATTACHED INSTRUMENTATION

The motor and gearbox are separated from the process fluid by a mechanical seal. This seal is typically cooled by the process fluid, but is designed with a separate cooling circuit if additional cooling is necessary; this circuit does not allow the coolant to mix with the process fluid. Because the process fluid might not adequately cool the seal at the high *GVF*s being investigated, the seal is cooled with water from a separate water line – not from the test platform's water supply. The seal is rated to withstand pressures well above the maximum 45 psi inlet pressure being investigated.

The rotor and stator together form four stages over a length of 49 inches. The pressure within the stages of the pump is measured by four pressure transducers located axially along the stator equidistant from one another, as seen in Figure 3. Thermocouples are also placed axially to measure the stator's temperature near its inner surface. The frictional heating of the stator, coupled with the lack of lubrication and cooling from the high *GVF* process fluid, could overheat the rubber stator and destroy the pump, but the thermocouples allow continuous monitoring of the rubber temperature during testing.

The air and water entering the PCP are mixed at the suction side of the pump and the mixture's temperature and pressure measured. This mixing and measurement is done in a vertical stack atop the pump's suction flange. As seen in Figure 4, the air and water are fed into the stack

through hoses near the top. A thermocouple and a pressure transducer are mounted on a flange at the top of the stack to measure the mixture's temperature and pressure. A pressure-relief valve prevents the suction pressure from exceeding 50 psi. At the bottom of the stack, an eight-inch flange allows alternate sources of air or water to be supplied to the pump. This alternate input is included to allow higher water flows into the pump than the water controls system can accommodate.

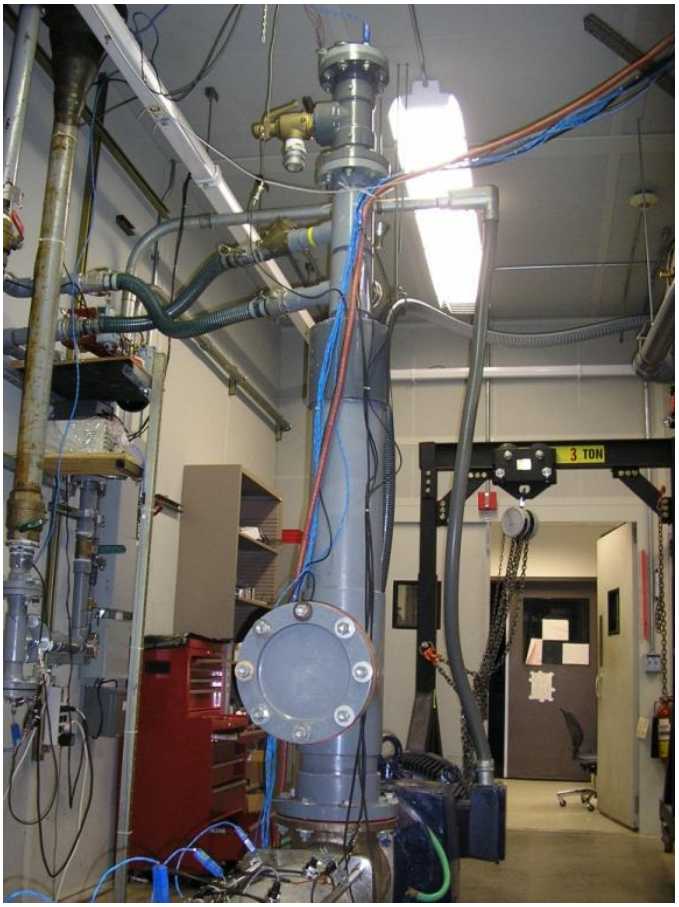


FIGURE 4: PHOTOGRAPH OF THE SUCTION STACK

AIR SUPPLY

The compressed air used for testing is drawn from one branch of the Turbomachinery Laboratory's compressed air system. The branch supplies air compressed up to 120 psi with a dew point of -40 °F.

All the control valves are pneumatically actuated, as is the back pressure regulator. Air for these devices is drawn from a separate branch of the air supply that is typically used for pneumatic tools and other mechanisms that require only small amounts of compressed air.

WATER SUPPLY

Figure 5 provides a schematic of the water supply system. The only practical way to continuously supply the necessary 64 gpm of water to the PCP for the duration of a test is to recirculate the water. All pipes in this system are either PVC or CPVC; the pipe size and schedule varies.

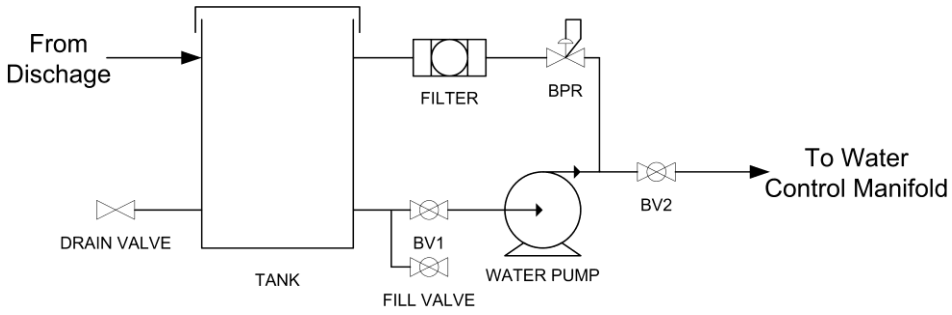


FIGURE 5: SCHEMATIC OF THE WATER SUPPLY SYSTEM

Water is held in a 5000-gallon fiberglass tank; the tank is ten feet in diameter and eight feet six inches at the maximum water level. The tank has a 2-inch, a 4-inch, and an 8-inch NPT opening at the bottom of the tank as well as a 2-inch and an 8-inch opening at the top. The top of the tank has a two-and-a-half foot diameter access hole that is covered with a wire mesh screen keep out debris while allowing air to vent. The tank also has two-inch diameter air vent at the top.

Prior to testing, the tank is filled from a hose connected to the fill valve, a ½ inch ball valve, at the bottom of the tank, as indicated Figure 5. The valve between the tank and the pump, BV1, is a two-inch ball valve which is closed when filling the. The drain valve is a four-inch gate valve attached to the eight-inch opening at the bottom of the tank. The four-inch opening at the bottom of the tank is capped at all times.

The water pump that moves water from the tank to the PCP is a centrifugal pump made by Ingersol-Rand, type H-HC, size 2X1-5X9H. The pump is rated to operate at 3550 rpm and supply 150 gpm with 347 total head feet. The motor driving the pump is a three-phase induction motor rated for 35 hp at 3520 rpm.

The water pump's discharge tees into two branches. One branch travels to the water control system through BV2 while the other feeds back into the tank through a back pressure regulator (labeled BPR in Figure 5). The back pressure regulator is adjusted to hold the water pressure exiting the water supply system at approximately 80 psi; higher pressures are not needed to test the PCP and could damage the pipes.

Water exiting the back pressure regulator passes through a filter before reentering the tank at the two-inch opening at the tank's top. The filter reduces the chances of any sediment or debris reaching one of the flow meters in the water control system.

## AIR CONTROL

The air control system regulates the air flow into the pump to maintain a constant pressure at the pump inlet. While the air supply provides compressed air at up to 120 psi, the PCP only needs up to 45 psi. Figure 6 provides a schematic of the air control system. All the pipes and fittings in this system are two-inch, schedule 80 CPVC, excepting the ball valve, BV3, which is a 4-inch brass valve.

The air flow is regulated by an electro-pneumatic control valve, marked as CV1 in the figure, made by Masoneilan. The valve is a Camflex II valve, model number 35-3512. A Daniel Mini Gas Turbine Meter, serial number 94-050167, measures the air flow rate upstream of the control valve and is denoted as FM1. This turbine meter can measure up to 6000 ACFH at up to 1440 psi. The meter is positioned on a vertical stretch of pipe with more than 20 inches of straight pipe

upstream and 10 downstream to allow the flow to develop. The meter also has three flow-straightening vanes. A thermocouple and a pressure transducer, labeled as TE1 and PE1, respectively, are mounted on a flange upstream of the turbine meter. The ball valve, BV3, can shut of the air supply when the PCP is not being tested.

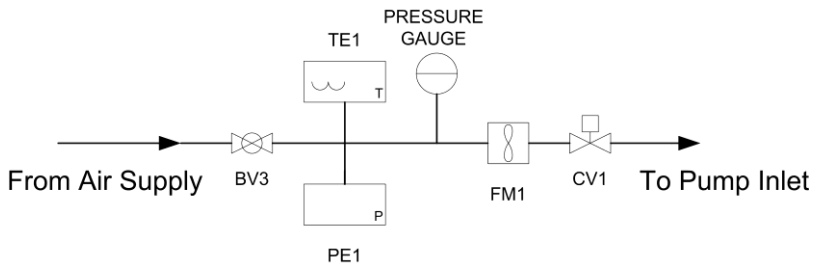


FIGURE 6: AIR CONTROL SYSTEM SCHEMATIC

WATER CONTROL

The water control system regulates the flow of water entering the pump. Water is piped from the water supply system to the control system, a schematic of which is shown in Figure 7. As the water enters the test cell, a four-inch dial pressure gauge displays the water pressure. This gauge provides a visual confirmation of the incoming water pressure during testing. The water temperature is measured by a thermocouple before splitting into three branches. Because the tests require accurate measurement and control of the flow rate between 64 to 6 gpm, it is impractical to use a single flow meter and control valve for all tests. Each branch accommodates a different range of flow rates and can be fully isolated using the ball valves, BV4 through BV9. The top branch can accommodate flow rates above 25 gpm, the middle branch between 5 and 50 gpm, and the bottom branch below 5 gpm. Note that all the ball valves in this system are brass, not CPVC.

The control valve in the top branch, denoted as CV2 in Figure 7, is a two-inch electro-pneumatically actuated control valve made by Masoneilan, model number 35-35212. The flow meter, FM2, is mounted approximately 20 inches downstream of the valve. The meter is manufactured by Daniel Industries, model number 1503-10, and can accommodate flow rates

between 25 and 225 gpm. The pipes and fittings in this branch are two-inch diameter, schedule 80, CPVC.

The middle branch uses a similar two-inch Masoneilan control valve, CV3, model number 30-30223. The flow meter is also located 20 inches downstream of the valve. The flow meter, FM3, is rated for flow rates between 5 and 50 gpm and is manufactured by Omega Engineering, model number FTB-1425. The pipes and fittings in this branch are schedule 80 CPVC – two-inches in diameter upstream of the ball valve and one-inch downstream.

The bottom branch can use one of two control valves, CV4 or CV5, to control the smallest flow rates. However, the bottom branch is never used in this investigation because the water flow rate is always greater than 5 gpm.

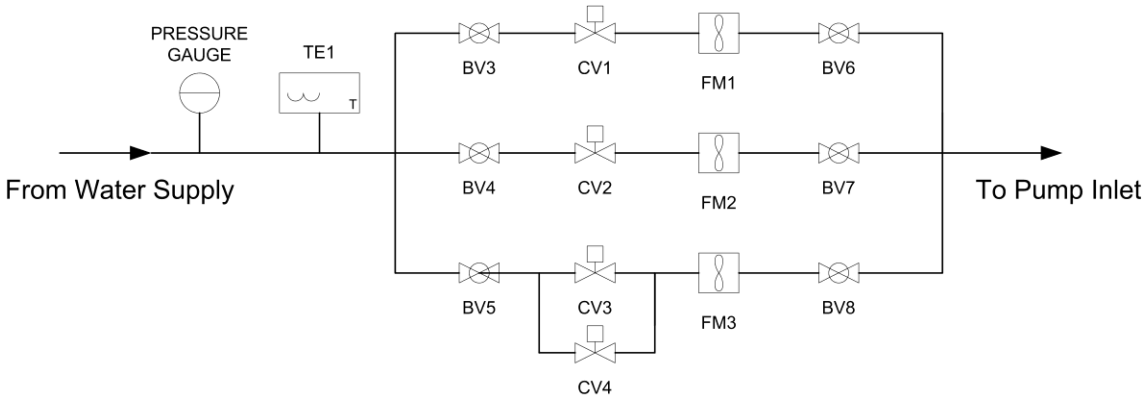


FIGURE 7: SCHEMATIC OF THE WATER FLOW CONTROL SYSTEM

DISCHARGE CONTROL

The discharge control system, pictured in Figure 8, regulates the pressure of the fluid exiting the pump. A stainless steel reducing bell reduces the eight-inch pump exit to a three-inch line. A thermocouple and a pressure transducer are mounted on the tee to gather pressure and temperature data at the pump discharge. A dial pressure gauge and safety relief valve also branch off the tee. The pressure gauge provides a visual confirmation of the pump's discharge pressure while the valve prevents the pump's discharge pressure from exceeding 200 psi.

A three-inch diameter Masoneilan control valve, model number 35-35212, throttles the pump's discharge to control the pressure at the pump exit. Downstream of the valve, the pipe expands back to an eight-inch line of schedule 80 CPVC; the increased pipe size decreases the bulk flow velocity within the pipe. The eight-inch line exits the test cell and carries the fluid to the water tank where the air and water separate.

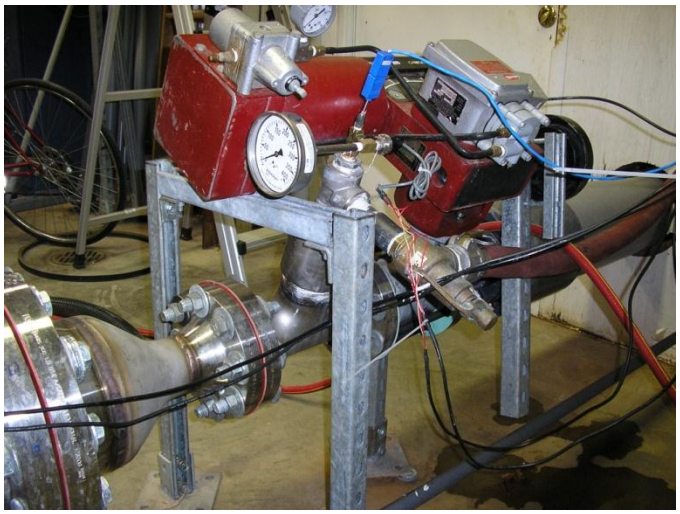

FIGURE 8: PHOTOGRAPH OF DISCHARGE MANIFOLD

## VARIABLE FREQUENCY DRIVE

Because this investigation requires that the PCP operate at half-speed, the pump must be powered by a variable frequency drive (VFD). A Dynamatic VFD, model number AF-520008-0480, supplies power to the PCP. The VFD is capable of driving a 200 hp three-phase motor at 480V. The VFD is configured to match the pump motor specifications; Table 1 summarizes the VFD settings for this investigation – all other settings are the VFD's factory defaults. While the power and speed settings are self-explanatory, the slip setting accounts for the inevitable mechanical slip in the motor and sets the actual running speed to 1785 rpm as specified on the motor's nameplate. The Volts/Hz setting is the motors rated voltage divided by its rated running frequency at that voltage, *i.e.* 480 V at 60 Hz. The ramp time specifies the time between the VFD starting the motor and the motor reaching full speed. The VFD's output frequency is also specified in its settings, either to 60 Hz for full-speed tests or 30 Hz for half-speed.

TABLE 1: SPECIFIED SETTINGS ON THE VARIABLE FREQUENCY DRIVE

| Power (HP) | Speed (RPM) | Slip (%) | Volts/Hz (V-s) | Ramp Time (s) |
|---|---|---|---|---|
| 100 | 1800 | 0.833 | 8 | 3 |

The VFD is located outside the test cell, but its output feeds into a fused disconnect mounted beside the PCP. A remote control that starts and stops the pump is wired to the VFD so that the pump can be operated without leaving the test cell. A signal output from the VFD that gives a voltage proportional to the power supplied to the pump connects to the DAQ.

## INSTRUMENTATION

The test platform uses thermocouples, pressure transducers, and flow meters to measure the PCP's performance at the various test conditions. All thermocouples employed in this investigation are grounded T-type thermocouples with a stainless steel sheath that are accurate to within ±0.5 °C (Omega Engineering, model number HTQSS). The thermocouples installed in the pump stator are all 1/16 inch diameter while the others are 1/8 inch.

The pressure transducers are also manufactured by Omega engineering. The transducers are from Omega's PX481A series. The transducer at the pump inlet is rated for pressures from 0-60 psig, those on the pump stator and air control system from 0-200 psig, and the one at the pump's outlet from 0-300 psig. These transducers are accurate to within .6 psi for the lower pressure transducers and 2 psi for the higher.

The specifications for the three flow meters used in this investigation have been presented in the preceding discussions of the air and water control systems.

All the thermocouples and pressure transducers employed in this investigation were purchased factory-calibrated with NIST traceable calibration. Nevertheless, several pressure transducers were checked using a deadweight tester to confirm their accuracy. The water flow meters were calibrated by weighing the amount of water that passed through them in a short time interval measured by a common digital stopwatch. The k-factor for each flow meter was computed by

recording the meter's output at ten different flow rates, measuring the actual flow rate by dividing the volume of water dispensed by the dispensing time, and performing a linear regression between the two data sets.

## DATA ACQUISITION SYSTEM

The thermocouples, pressure transducers, and VFD signal all connect to a series of National Instruments (NI) I/O modules housed in an NI DAQ chassis. The thermocouples are connected to a NI-9213 module: a 16-channel thermocouple input module with built in cold-junction compensation. This module has 24-bit analog-digital conversion and samples each channel, sequentially, 75 times per second (up to 1200 samples per second total).

The voltage signals from the pressure transducers and VFD are read by an NI-9205 module. This module reads up to 16 differential voltage inputs with configurable voltage ranges of ±200 mV, ±1 V, ±5 V, or ±10 V. The module has 16-bit analog-digital conversion and samples channels sequentially, taking up to 250000 samples per second across all channels.

The signals actuating each control valve are also produced by an NI module, the NI-9265. The NI-9265 is a 4-channel 4-20 mA analog output module with 16-bit digital-analog conversion. The channels are updated simultaneously 100,000 times per second.

The NI modules are designed to plug into an NI chassis, the cDAQ-9172. This chassis holds up to eight NI modules and sends the outputs from the module to a computer via a USB connection.

The flow meters produce a frequency signal rather than a voltage like the other instruments. Two frequency conditioners from Omega Engineering, model number iFPX-W, convert the signals from the each flow meter into a digital signal that can be read by any computer over a LAN connection. The conditioners can read signals between 1 Hz and 100 kHz and have a frequency resolution of $10^{-10}$ Hz. The conditioners output the average frequency over every one-second interval.

All data collected in this investigation are recorded on a PC running National Instruments (NI) LabVIEW version 10.0. The PC runs on Windows Vista and has an Intel Core2 6300 processor and 2 Gb of RAM. A Virtual Instrument (VI) in LabVIEW collects and records all

measurements during the experiment. The VI also generates the control signals for the control valves and provides a graphical interface for both observing the system measurements in real-time and controlling the PCP's suction pressure, discharge pressure, and gas volume fraction. Figure 9 and Figure 10 show screenshots of the VI interface.

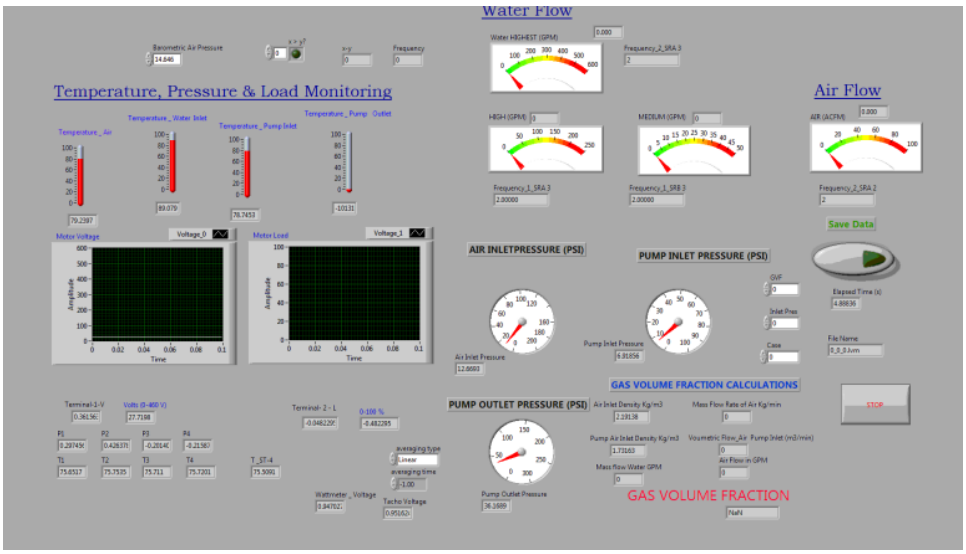


FIGURE 9: SCREENSHOT OF THE VI INTERFACE FOR FLOW CONTROL AND MONITORING

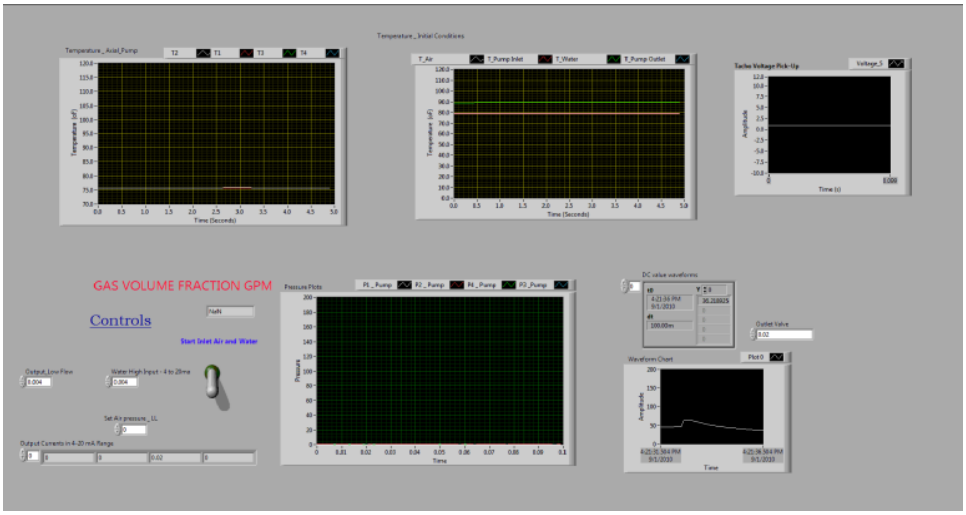


FIGURE 10: SCREENSHOT OF VI INTERFACE FOR MONITORING STATOR AND OUTPUT CONDITIONS

# TEST PROCEDURE

All testing on the PCP follows a procedure developed to protect the pump and the supporting equipment and collect test data accurately and quickly. The procedure is broken into three chronological steps: the startup, the testing, and the shut down.

STARTUP

To begin a test on the PCP, all the support systems are first checked and readied before the pump is turned on. Because the control valves and back-pressure regulator (BPR) are all pneumatically actuated, their air supply is turned on first. Next, all the valves in the system are closed, whether manually, as for the ball valves, or electrically, for the control valves. The valve between the water tank and the water supply pump is then opened and the water supply pump started. After checking that the water pump is generating pressure and that the back-pressure regulator is maintaining the desired water pressure in the test line, the ball valve downstream of the pump (BV2 in Figure 5) is opened slowly; opening the valve too quickly hammers the pipes and could damage the system. Depending on the branch of the water control system to be used, either BV4 and BV7 or BV5 and BV8 are then opened.

Once the water supply is prepared, the ball valve in the air control system, BV3, is opened and the dial pressure gauge checked to ensure the system was supplying adequate pressure for testing. The seal cooling line is opened and the drain on the pump suction is confirmed closed. The pump's electrical disconnect is then closed and the VFD turned on. The settings on the VFD are checked and the frequency set for the anticipated tests.

After all the ancillary equipment is ready, the LabVIEW VI is set to close all control valves and started. The discharge valve is fully opened through the VI interface.

Since the pump cannot sustain vacuum pressures on its suction, the air and water must be flowing into the PCP before starting. However, if air and water are allowed into the suction stack for too long while the pump isn't running, they will overpressurize the suction stack and open the pressure relief valve. Consequently, the pump is started from the remote within seconds of opening the air and water control valves in LabVIEW.

Once the PCP starts, the discharge valve must be throttled to generate pressure at the discharge. Running the pump with lower pressure at the discharge than at the suction (*i.e.* $\Delta P<0$) may damage the pump.

## TESTING

After the pump is running stably, all dial gauges are checked to confirm the VI measurements and ensure that the pump is operating as expected. The desired suction pressure, $P_s$, is set in the VI; this setting is fed into a PID control within the VI that automatically adjusts the air control valve to maintain the desired pressure. The water control valve is adjusted to produce the desired *GVF*. Because of the constant fluctuations in the discharge pressure as each stage reaches the end of the pump, the discharge valve must be adjusted manually. The VI displays the average $\Delta P$ over each one-second interval. The control signal to the valve is manually adjusted to bring this average as close as possible to the desired $\Delta P$. The pump is then run at that condition for several minutes until the internal pump temperatures, displayed on a graph in the VI, no longer appear to change. After the pump reaches steady state, the name for the data file is specified, and the save button in the VI is clicked; five to ten seconds of data are recorded. Once a test condition is recorded the $\Delta P$ is increased by 30 psi. After the pump reaches a steady state, the data for this new test condition is saved. This process is repeated until $\Delta P$ has reached 150 psi.

After all six values of $\Delta P$ are recorded for a single $P_s$ and *GVF*, $\Delta P$ is reduced and $P_s$ is raised by 15 psi to the next test condition. $\Delta P$ is reset to zero, and the testing sequence repeats. After all combinations of $P_s$ and $\Delta P$ are tested and recorded for a single *GVF*, the water control valve is adjusted to increase the *GVF* to the next test condition. This process is repeated until all configurations have been tested at one speed. Table 2 summarizes the all the test conditions for a single speed.

TABLE 2: *GVF*, $P_S$, AND $\Delta P$ FOR ALL TEST CONDITIONS

| ***GVF*** | $P_s$ | $\Delta P$ | | | | | |
|---|---|---|---|---|---|---|---|
| 90 | 15 | 0 | 30 | 60 | 90 | 120 | 150 |
| | 30 | 0 | 30 | 60 | 90 | 120 | 150 |
| | 45 | 0 | 30 | 60 | 90 | 120 | 150 |
| 92 | 15 | 0 | 30 | 60 | 90 | 120 | 150 |
| | 30 | 0 | 30 | 60 | 90 | 120 | 150 |
| | 45 | 0 | 30 | 60 | 90 | 120 | 150 |
| 94 | 15 | 0 | 30 | 60 | 90 | 120 | 150 |
| | 30 | 0 | 30 | 60 | 90 | 120 | 150 |
| | 45 | 0 | 30 | 60 | 90 | 120 | 150 |
| 96 | 15 | 0 | 30 | 60 | 90 | 120 | 150 |
| | 30 | 0 | 30 | 60 | 90 | 120 | 150 |
| | 45 | 0 | 30 | 60 | 90 | 120 | 150 |
| 98 | 15 | 0 | 30 | 60 | 90 | 120 | 150 |
| | 30 | 0 | 30 | 60 | 90 | 120 | 150 |
| | 45 | 0 | 30 | 60 | 90 | 120 | 150 |

When performing tests on the pump, there are several points of which the operator must be mindful. The axial temperatures in the pump must always be monitored for two reasons. The first and most obvious is to ensure that the pump does not overheat (*i.e.* the rubber stator temperature must be below 160 °F). The highest temperature within the stator is always found at the probe nearest the discharge, $T_{ax}^4$. Secondly, the temperatures should be monitored after changing the test condition to determine when internal pump temperatures become constant, indicating that the pump has reached a steady-state condition. Data is only collected after the pump appears to have reached its steady-state condition for that configuration. The suction and discharge pressures should also be monitored to ensure they do not go above 50 psi and 250 psi, respectively. If the pressure exceeds those limits, the pressure relief valves will trigger. Suction pressure can climb if LabVIEW stops running for some reason – the PID control for the air control valve is needed to hold the suction pressure constant. The discharge pressure only exceeds its limits when the discharge control valve is closed too far.

## SHUTDOWN

Shutting down the PCP is nearly the reverse of the startup procedure. First the pump is turned of at the remote and the air and water control valves shut off in the VI immediately thereafter. The

air supply ball valve is closed, along with ball valves on the water manifold. The suction stack of the pump is drained using the drain valve at the bottom of the pump. The water supply ball valve is closed and the water supply pump turned off. Finally the tank valve is closed, the air to the control valves turned off and the seal cooling water valve closed.

# DATA PROCESSING

All the collected data was processed using Matlab – the full program is available in the appendix. Data saved by LabVIEW are stored in tab-delimited text files that can be directly imported into Matlab. Each column in the file contains the data from one instrument. The data is averaged over the entire sample time for each run to evaluate the mean value of that measurement. The standard deviation of the data is also calculated to find the uncertainty of the measurement.

Several values of interest cannot be measured directly and must be calculated. To find the flow rate of the air entering the pump, the flow rate through the air flow meter is first computed by the equation,

$$Q_{air\ in} = \frac{AFM}{5.3138} \quad \text{(gpm)} \tag{1}$$

The air flow meter produces a voltage pulse for every turbine revolution. The frequency of its output, denoted *AFM*, is directly proportional to the flow rate. Once the volumetric flow rate going through the air flow meter is known, the air flow rate entering the pump can be computed by the equation,

$$Q_{air} = Q_{air\ in}\frac{(P_{air}+14.7)(T_s+459.67)}{(P_s+14.7)(T_{air}+459.67)} \quad \text{(gpm)} \tag{2}$$

Knowing that the mass flow rate of the air passing through the meter and into the pump is the same, the relation between the volumetric flow rates at those locations is derived by applying the Ideal Gas Law. Note that all pressures are measured in psig and all temperatures in °F.

Finding the water flow rate entering the pump is simpler because water is incompressible (under the conditions of this investigation); the flow rate into the pump is the flow rate through the water flow meter. As for the air flow meter, the water flow rate is proportional to the frequency output of the meter. The total water flow rate is computed to be,

$$Q_{water} = \frac{WFM_a}{15.1833} + \frac{WMF_b}{1.9417} \quad \text{(gpm)} \tag{3}$$

Since two flow meters are employed during testing, the water flow rate is computed as the sum of the flow rate through each meter. The frequency output of the medium-flow meter is denoted

$WFM_a$ while that through high-flow meter is denoted $WFM_b$. Only one flow meter is operated during any single test, so one of the flow meter outputs is always zero; always adding the flow rates instead of only using the rate of the operating meter simplifies the flow rate computations.

Once the flow rates of air and water entering the pump are calculated, the process fluid's total flow rate and the *GVF* can be calculated as the sum of $Q_{air}$ and $Q_{water}$ and the ratio of $Q_{air}$ to $Q$, respectively.

$$Q = Q_{air} + Q_{water} \quad \text{(gpm)} \tag{4}$$

$$GVF = \frac{Q_{air}}{Q} \quad (\%) \tag{5}$$

The pressure difference across the pump is simply the discharge pressure, $P_d$, less the suction pressure, $P_s$.

$$\Delta P = P_d - P_s \quad \text{(psi)} \tag{6}$$

The power supplied to the pump is measured using a voltage output from the VFD that is directly proportional to the motor's electric load. The VFD's documentation does not explain how the output corresponds to the motor load, so a relation was determined empirically. The pump is operated at five different running conditions at each speed; the conditions are selected to sample across the full range of operating conditions. While the pump is operating, two multimeters measure the voltage and current at the pump's disconnect. The voltage is measured between two legs of the three-phase and the current is measured indirectly by measuring the voltage across a current transformer that encircles one of the power lines. The voltage and current measurements allow the motor's power output to be calculated by the equation,

$$Power = \sqrt{3}(Voltage)(Current)0.86 \quad \text{(W)} \tag{7}$$

where 0.86 is the power factor for the motor and $\sqrt{3}$ is present because of the three-phase power. Given the VFD's output at each test condition, the calculated power delivered by the motor, and the fact that the two are linearly related, a simple linear regression finds the relation between the power and the VFD output. Figure 11 shows the collected data points and the regression for both full and half speeds, along with the equations and $R^2$ values for each regression.

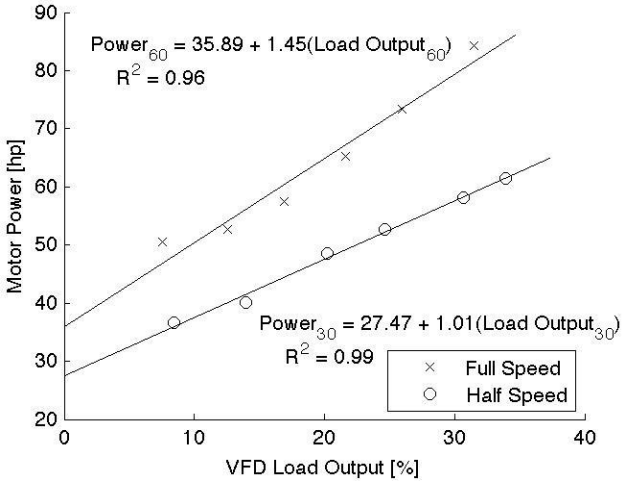


FIGURE 11: COLLECTED POWER DATA AND REGRESSION LINES FOR FULL AND HALF SPEEDS

The efficiency of a pump is, by definition, the ratio of the power it adds to the process fluid (*i.e.* hydraulic power) to the power supplied to the pump. Because of the two-phase nature of the process fluid, the hydraulic power is not only difficult to accurately measure in any circumstance, but also impossible to determine from the data collected in this investigation. However, a nominal efficiency is calculated by treating the process fluid as incompressible. The nominal hydraulic power then becomes $Q\Delta P$, and the nominal efficiency is calculated as,

$$\eta = \frac{Q\Delta P}{L} \quad (\%) \tag{8}$$

The following equations, derived using the Kline-McClintock method, determine the uncertainty of these computed values based on the respective uncertainty of the measured quantities. The uncertainty of any measured quantity is two standard deviations of the measurement data for each test condition – a 95% confidence interval. Note that $U_i$ denotes the uncertainty of the quantity $i$.

$$U_{\Delta P} = \sqrt{U_{P_s}^2 + U_{P_d}^2} \quad \text{(psi)} \tag{9}$$

$$U_{Q_{water}} = \sqrt{\left(U_{WFM_a}\frac{60}{911}\right)^2 + \left(U_{WFM_b}\frac{60}{116.5}\right)^2} \quad \text{(gpm)} \tag{10}$$

$$U_{Q_{air\,in}} = U_{AFM}\frac{60}{2385*0.133681} \quad \text{(gpm)} \tag{11}$$

$$U_{Q_{air}} = \sqrt{a+b+c+d+e} \quad \text{(gpm)} \tag{12}$$

$$a = \left(U_{Q_{air\,in}}\frac{(P_{air}+14.7)(T_s+459.67)}{(P_s+14.7)(T_{air}+459.67)}\right)^2 \quad \text{(gpm)} \tag{13}$$

$$b = \left(U_{P_{air}}Q_{air\,in}\frac{(T_s+459.67)}{(P_s+14.7)(T_{air}+459.67)}\right)^2 \quad \text{(gpm)} \tag{14}$$

$$c = \left(U_{T_s}Q_{air\,in}\frac{(P_{air}+14.7)}{(P_s+14.7)(T_{air}+459.67)}\right)^2 \quad \text{(gpm)} \tag{15}$$

$$d = \left(U_{P_s}Q_{air\,in}\frac{(P_{air}+14.7)(T_s+459.67)}{(P_s+14.7)^2(T_{air}+459.67)}\right)^2 \quad \text{(gpm)} \tag{16}$$

$$e = \left(U_{T_{air}}Q_{air\,in}\frac{(P_{air}+14.7)(T_s+459.67)}{(P_s+14.7)(T_{air}+459.67)^2}\right)^2 \quad \text{(gpm)} \tag{17}$$

$$U_{\mathrm{Q}} = \sqrt{U_{Q_{water}}^2 + U_{Q_{air}}^2} \quad \text{(gpm)} \tag{18}$$

$$U_{GVF} = \sqrt{\left(U_{Q_{air}}\frac{1}{Q}\right)^2 + \left(U_Q\frac{Q_{air}}{Q^2}\right)^2} \quad (\%) \tag{19}$$

$$U_{\mathrm{Q}\Delta P} = \sqrt{\left(U_Q\Delta P\right)^2 + (U_{\Delta P}\mathrm{Q})^2} \quad \text{(gpm} \cdot \text{psi)} \tag{20}$$

# RESULTS

The results of the experimental investigation will be presented in their entirety, followed by a discussion after all the results have been presented. The PCP is operated with nominal *GVF*'s of 0.90, 0.92, 0.94, 0.96, and 0.98; $P_s$ of 15, 30, and 45 psi; and $\Delta P$ of 0, 30, 60, 90, 120, and 150 psi. LabVIEW records data from the sensors at each configuration for a period of five to 10 seconds, sampling each channel at one millisecond intervals.

The pump's volumetric flow rate at its inlet as a function of the suction pressure is shown in Figure 12 for all test conditions. The two solid lines are the result of linear regressions performed on both the 30 and 60 Hz data sets. The measurement uncertainty is calculated as the 95% confidence interval for each test condition. The average uncertainty in the flow rate measurements is ±5.0 gpm for the full-speed an ±2.2 gpm for half-speed.

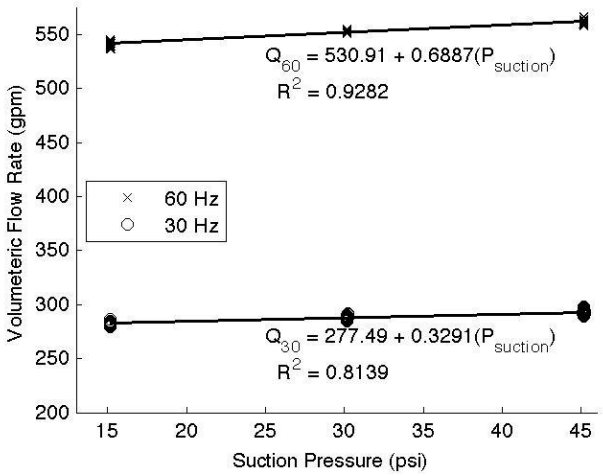


FIGURE 12: VOLUMETRIC FLOW RATE AT THE INLET VERSUS THE SUCTION PRESSURE

Figure 13 shows the ratio of the flow rate at 30 Hz, $Q_{30}$, to the flow rate at 60 Hz, $Q_{60}$. The solid line is the linear regression of the data set. Since the flow rate is, theoretically, directly proportional to the pump speed, this ratio should be ½ regardless of the pressures or operation

fluid. A tachometer attached to the pump measures the pump's rotational speed, $\omega$. The ratio of the pump's speed at half-speed to that at full-speed, with a 95% confidence interval, is,

$$\frac{\omega_{30}}{\omega_{60}} = 0.5061 \pm 0.0014 \tag{21}$$

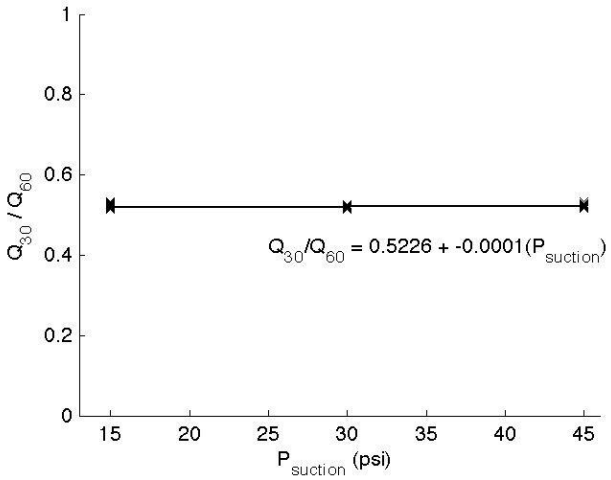


FIGURE 13: RATIO OF THE FLOW RATE AT 30 HZ TO THAT AT 60 HZ FOR TEST CONFIGURATIONS

The product of the flow rate and pressure change, $Q\Delta P$, is commonly used in liquid pumps to measure the power transmitted to the fluid. While these tests used an air-water mixture for the process fluid, $Q\Delta P$ gives a basis for comparing this progressing cavity pump to other pumps. Figure 14 shows $Q\Delta P$ versus $\Delta P$ for all flow conditions at both the full and half-speed tests. The solid lines are the linear regressions of each data set. The error bars show the 95% confidence interval of each calculated value.

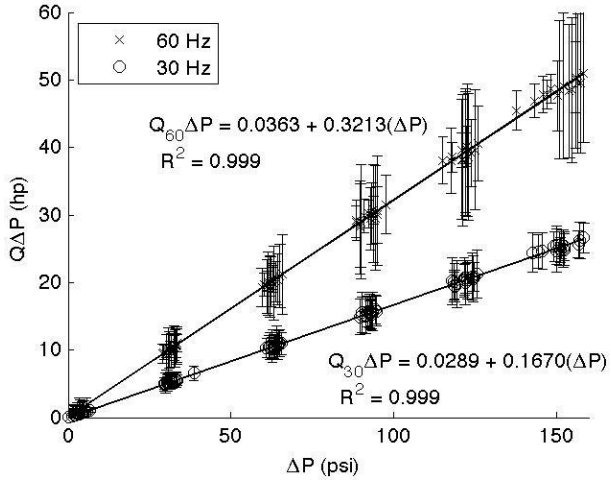


FIGURE 14: QΔP (HP) VERSUS ΔP AT BOTH FULL AND HALF-SPEED TEST CONFIGURATIONS

The large uncertainties in the measured $Q\Delta P$s are largely due to fluctuations in the discharge pressure. These fluctuations occur because the fluid in each cavity is forced out of the pump separately; this causes the pressure between the pump's exit and the discharge control valve to pulsate in a regular fashion. Figure 15 shows these fluctuations for a single test condition.

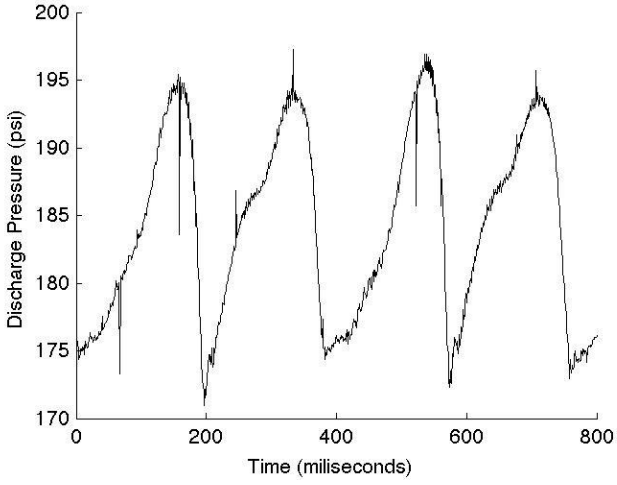


FIGURE 15: PERIODIC FLUCTUATIONS IN THE PUMP'S DISCHARGE PRESSURE OVER A PERIOD OF 0.8 SECONDS. THE DATA IS TAKEN FROM THE TEST CONDITION $GVF$ = 0.94, $P_A$ = 30 PSI, $\Delta P$ = 150 PSI

The pump stator is fitted with four thermocouples, as shown in Figure 3, to measure the temperature of the rubber stator. In all tests, the temperature is highest at the thermocouple nearest the discharge. Figure 16 shows a plot of some typical temperature profiles along the length of the pump for six values of $\Delta P$ at 0.96 *GVF* and 30 psi $P_s$ at half-running speed; the fifth position on the plot is the temperature of the fluid at the pump's exit, in the three inch diameter section of the discharge manifold. The axial temperature data has very low uncertainty, averaging only 0.05 °F over all test conditions with a maximum uncertainty of 0.93 °F.

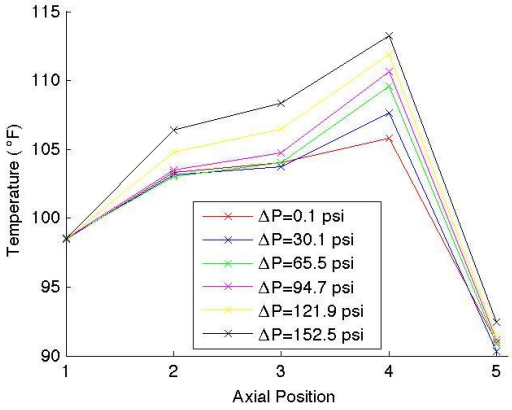


FIGURE 16: TYPICAL AXIAL TEMPERATURE PROFILE, THIS CASE IS WITH 0.96 *GVF* AND 30 PSI $P_S$ AT HALF-SPEED. AXIAL POSITION 5 DENOTES THE TEMPERATURE AT THE PUMP'S DISCHARGE

One goal of this investigation is to determine if the pump could be run at high *GVF*s without overheating the pump and destroying the stator. Consequently, the stator's maximum temperature holds the greatest interest. The maximum temperature of the stator for each test condition is depicted in Figure 17 and Figure 18 for full and half-speed, respectively.

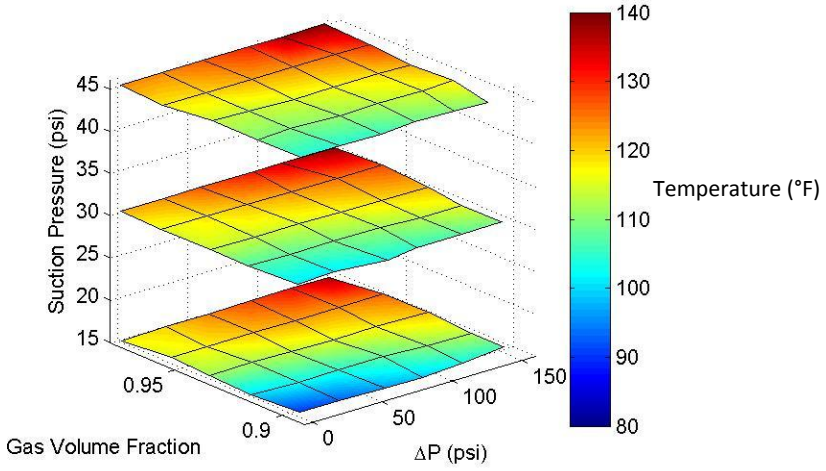


FIGURE 17: MAXIMUM STATOR TEMPERATURE (°F) AT FULL-SPEED TEST CONDITIONS

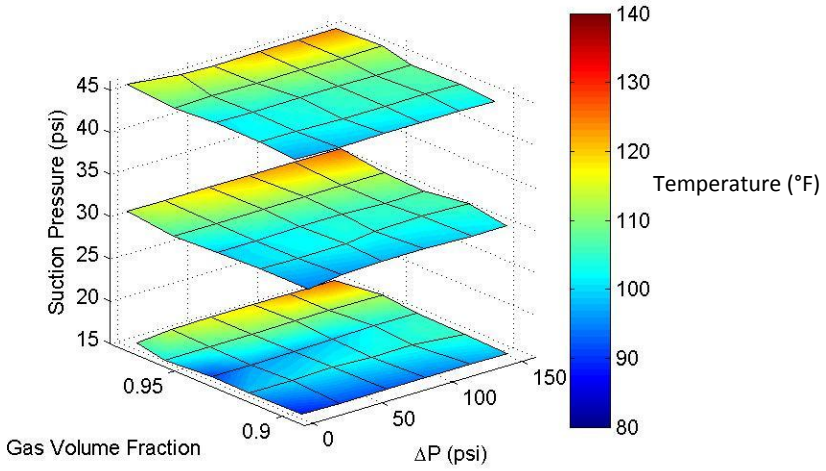


FIGURE 18: MAXIMUM STATOR TEMPERATURE (°F) AT HALF-SPEED TEST CONDITIONS

In these plots, the temperature is represented by color, increasing from blue to red, while the *x*, *y*, and *z* axes show the $\Delta P$, *GVF*, and $P_s$, respectively. The collected data points are represented by the intersections on the grid. The shading estimates what the temperature would be for untested

configurations using bilinear interpolation on the values at the corners of each grid square. The color scales are the same for both plots.

To construct a workable empirical model for the temperature as a function of the *GVF*, $\Delta P$, and $P_s$, a linear model can be constructed using multiple-linear regression. The resultant equations for maximum temperature for the full and half-speeds are:

$$T_{60} = -184.9 + 0.0792(\Delta P) + 0.1616(P_{suct}) + 308.8(GVF) \quad (°F) \qquad (22)$$

$$T_{30} = -148.8 + 0.0579(\Delta P) + 0.1395(P_{suct}) + 257.5(GVF) \quad (°F) \qquad (23)$$

The correlation coefficients, $R^2$, for the full and half-speed models are 0.965 and 0.892, respectively. Figure 19 and Figure 20 depict the temperatures predicted by these models. The scales are identical to those used in Figure 17 and Figure 18.

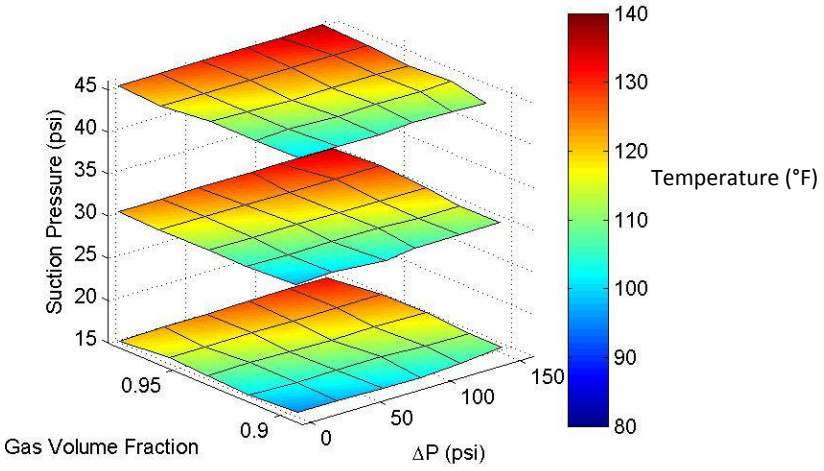


FIGURE 19: PREDICTED MAXIMUM STATOR TEMPERATURES (°F) FOR FULL-SPEED TEST CONDITIONS

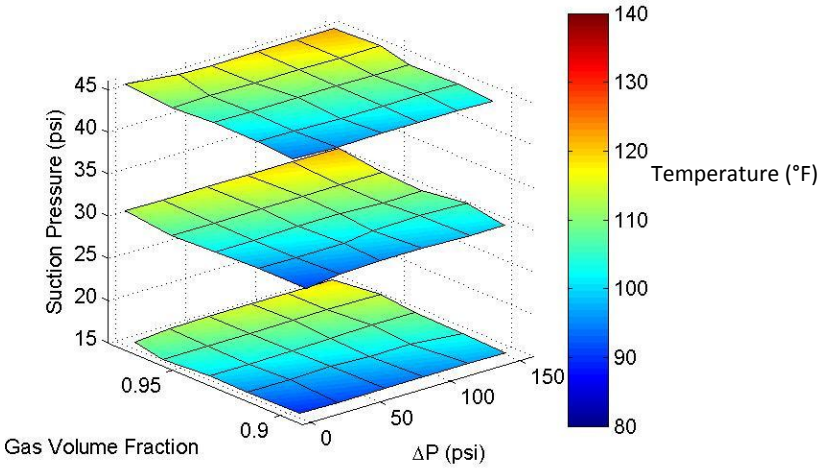


FIGURE 20: PREDICTED MAXIMUM STATOR TEMPERATURES (°F) FOR HALF-SPEED TEST CONDITIONS

Along with the thermocouples measuring the temperature within the pump, the stator is also fitted with four pressure transducers to measure the development of the fluid pressure along the length of the pump. The mean pressure at each tap is normalized according to the equation,

$$Normalized\ Pressure = \frac{P_{ax}^{i} - P_S}{\Delta P} \tag{24}$$

This normalization better reveals trends within the data. Figure 21 and Figure 22 show the normalized mean pressure within the pump at each pressure tap for selected test conditions. The suction pressure has no significant effect on the normalized pressure, so, for any given *GVF* and *ΔP*, the corresponding pressures in Figure 21 and Figure 22 are the average of the normalized pressures at all $P_s$. For clarity, the cases with 0.92 and 0.96 *GVF* have been omitted from the plots.

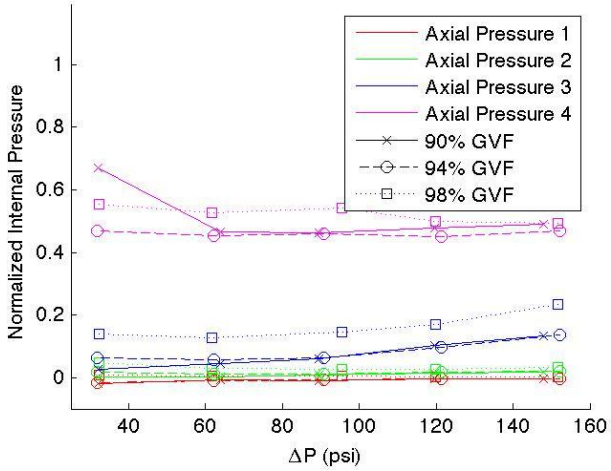


FIGURE 21: NORMALIZED PRESSURES AT THE AXIAL PRESSURE TAPS FOR FULL-SPEED TEST CONDITIONS WITH *GVF*S OF 0.90, 0.94, AND 0.98

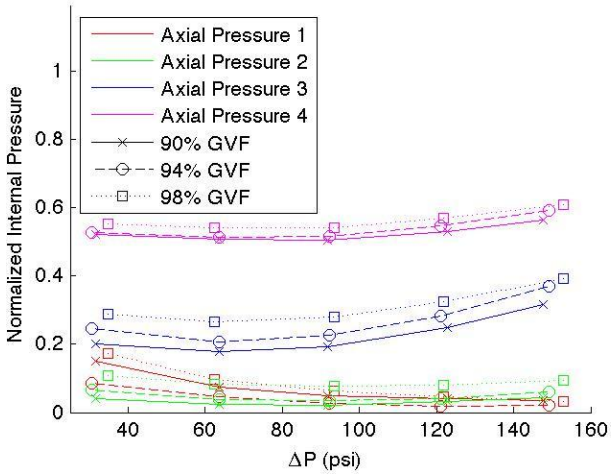


FIGURE 22: NORMALIZED PRESSURES AT THE AXIAL PRESSURE TAPS FOR HALF-SPEED TEST CONDITIONS WITH *GVF*S OF 0.90, 0.94, AND 0.98

The axial pressure data contained two glaring anomalies. The first of these occurred at the first pressure tap, while testing the pump at half-speed with $GVF$ = 0.90, $\Delta P$ = 30 psi, and $P_s$ = 15 psi. The recorded data indicated that $P_{ax}^{1}$ = 473.0 psi with a standard deviation of 7.97 * $10^{-11}$. This is not only physically impossible, but well outside the pressure transducer's range. Based on

the improbably low standard deviation, a loose connector most likely caused this fault. Clearly, this data point is invalid, and it has consequently been discarded.

The second anomaly was not discarded and its effects are visible in Figure 21. While running the pump at full speed with *GVF* = 0.90, $\Delta P$ = 30 psi, and $P_s$ = 15 psi, the recorded data states that $P_{ax}^4$ = 56.96 psi – significantly greater than the discharge pressure for this condition. This measurement's standard deviation was only 7.72 psi. Although this measurement does not follow the expected trends, there is no evidence to suggest that this data point is invalid. Further investigations on this pump should check the results of this test.

The power supplied to the pump was measured using an output from the VFD that gives a voltage proportional to the ratio of the motor load to the maximum power available. This output was used to compute the power supplied based on voltage and current measurements at several test conditions. The power output of the motor to the pump is shown in Figure 23 for all test conditions as a function of $\Delta P$. The two solid lines are the result of the linear regressions on both the 30 and 60 Hz data.

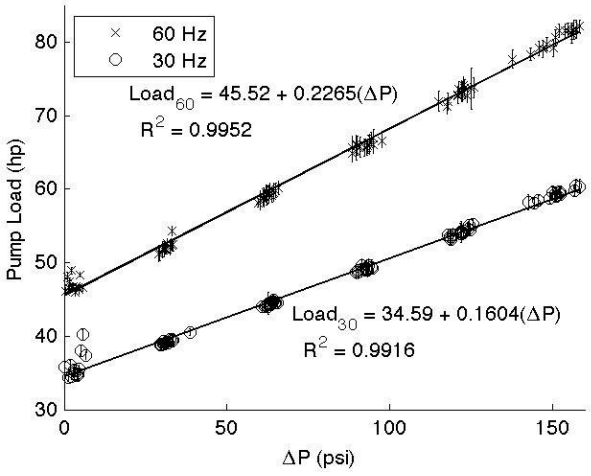


FIGURE 23: CALCULATED MECHANICAL LOAD OF THE PUMP VERSUS ΔP AT BOTH FULL AND HALF-SPEED TEST CONDITIONS

Since both the pump's mechanical load and the nominal power, Q*ΔP*, are calculated for each test condition, dividing Q*ΔP* by the load will give a nominal efficiency for the pump. Figure 24 shows the nominal efficiency of the pump versus *ΔP* for all operation conditions. Dividing the linear regressions of the data in Figure 14 and Figure 23 produces the curves in the Figure 24.

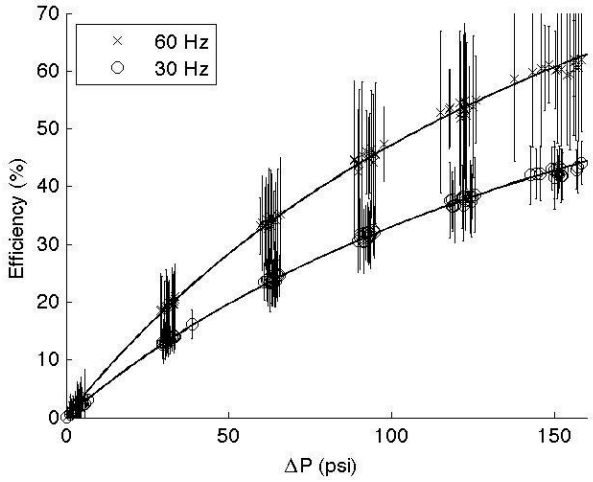


FIGURE 24: CALCULATED NOMINAL EFFICIENCY OF THE PUMP VERSUS *ΔP* AT BOTH FULL AND HALF-SPEED TEST CONDITIONS

# DISCUSSION OF RESULTS

The flow rate of the pump is shown Figure 12 for all test conditions. Since the progressing cavity pump is a type of positive displacement pump, its flow rate should be directly proportional to its running speed. The tests demonstrated that flow rate is indeed independent of both *GVF* and $\Delta P$, but they also revealed that the flow rate increases with $P_s$. According to the regression equations, the flow rate increases 0.678 gpm per psi of $P_s$ when operating at full-speed and 0.301 gpm per psi at half-speed. Comparing the flow rate at 15 psi at the pump suction to that at 45 psi, the tests indicate that the flow rate increases by 20.3 gpm (3.8%) and 9 gpm (3.3%) at full and half-speed, respectively. Furthermore, the flow rate's uncertainty is ±5 gpm at full speed an ± 2.2 gpm at half-speed; the increase in flow rate is beyond what can be explained by uncertainty.

This finding is quite unexpected. The flow rate of a progressing cavity pump is determined strictly by the pump's operating speed and the geometry of the cavities. Tachometer readings confirm that the pump speed remained constant during testing, so the pump cavity geometry must have changed. The increases in suction pressure naturally create higher pressures throughout the pump. The higher pressure must deform the rubber stator to produce a marginally larger cavity within the pump.

Another possibility is that the air-water mixture entering the pump actually has a higher density than what is calculated based on the temperature and pressure measurements at the top of the suction stack. The volumetric flow rate is calculated from the mass flow rate, as measured at the flow meters; if the calculated density is too low the resulting calculated volumetric flow rate will be higher than in reality. Evaporative cooling and simple mixing of the air and water within the suction stack may produce a temperature gradient within the stack, the pump inlet being cooler than the thermocouple at the top. Further investigation is needed to determine if these effects are significant and if the pump's flow rate truly increases with $P_s$.

Regardless of any increase in the flow rate with $P_s$, the ratio of the flow rate at half-speed to that at full speed is effectively constant and independent of $P_s$, $\Delta P$, and *GVF*, as seen in Figure 13. Ideally, a positive displacement pump's flow rate is proportional to its speed; the ratio of the flow rate at 30 Hz to that at 60 Hz should equal the ratio of the rotational speeds at those frequencies, *i.e.*:

$$\frac{Q_{30}}{Q_{60}} - \frac{\omega_{30}}{\omega_{60}} = 0 \tag{25}$$

However, comparing 21 to the regression in Figure 13, it is apparent that

$$\frac{Q_{30}}{Q_{60}} - \frac{\omega_{30}}{\omega_{60}} = 0.0172 \tag{26}$$

The difference greater than what can be accounted for by measurement uncertainty; the pump's flow rate is not proportional to its speed. The flow rate operating the pump at 60 Hz is lower than expected. This decrease in the flow rate is most likely due to an increase in the leakage between stages at higher speeds.

The product of the pump's flow rate and the difference between the suction and discharge pressures, $Q\Delta P$, is plotted with respect to $\Delta P$ in Figure 14. It is immediately apparent that $Q\Delta P$ increases linearly with $\Delta P$, hardly surprising since $Q$ itself is independent of $\Delta P$. While there may be some dependence on $P_s$ as is seen in the flow rate, $\Delta P$ is clearly the dominant variable. This dominance becomes even clearer when examining the uncertainty in this quantity. Although $Q$ has a small uncertainty, as seen in Figure 12, $\Delta P$ has very large uncertainty due to pulsation in the output pressure.

A PCP moves fluid from the suction to the discharge in discrete, sealed chambers. A consequence of this design is that the independent pockets of fluid exit the pump one a time, causing pulsations in the discharge pressure, as seen in Figure 15. If the pump discharges into a very large volume (compared to a pump stage) of fluid under pressure, *e.g.*, a pipeline, the pulsation would be negligible. However, in this test platform, the pump discharges into a reducing bell followed by a short length of pipe and a throttling valve. The pressure fluctuations in this small chamber are large and increase with the discharge pressure. Consequently, any measurement of $\Delta P$ will be have a large uncertainty that will increase with $\Delta P$. Uncertainty could be reduced by increasing the volume between the pump's discharge and the discharge control valve or by simply placing a pressure snubber inline with the pressure transducer.

Interestingly, the uncertainty in $\Delta P$ measurements tends to decrease as the *GVF* increases. This is likely because the fluid is more compressible at higher *GVF*s. Alternatively, the water exiting the pump may from "slugs" that cause a pressure spike as they pass through the discharge. Future

tests operating the PCP at very low *GVF*s may experience potentially damaging pulsations in the discharge pressure, *i.e.*, water hammer. Additionally, designers and operators should be mindful that the pulsations could cause non-linear hardening in the process fluid.

The most important finding evident in Figure 14 is that $Q\Delta P$ is independent of the *GVF*. If the process fluid were incompressible, $Q\Delta P$ would be the power that that the pump adds to the fluid, *i.e.* the hydraulic power. If $Q\Delta P$ remains independent of the *GVF* as the *GVF* decreases to zero, these tests will have found the hydraulic power of the pump running only water without actually requiring a large supply of water for testing. Further testing is needed to demonstrate that $Q\Delta P$ remains independent of the *GVF* below 0.90. If the independence holds, this finding could greatly reduce the requirements for testing any similar progressing cavity pump, even very large pumps that may be impractical to test using only water. Furthermore, these results can be extrapolated to find the hydraulic power into a process fluid with any *GVF*, assuming an accurate knowledge of the fluid's properties as it enters the pump.

One of the primary concerns when running a progressing cavity pump at a very high *GVF* is that the pump may overheat and destroy the rubber stator. Four thermocouples were inserted along the length of the pump's stator to monitor the rubber temperature during testing, as depicted in Figure 3. In every test, the highest temperature in the stator was measured at the thermocouple closest to the pump's discharge. The readings from these thermocouples are represented in Figure 17 and Figure 18 for full and half-speed test conditions, respectively. It is immediately apparent that the pump's temperature depends on all the test variables: running speed, $\Delta P$, $P_s$, and *GVF*. Obviously, and intuitively, the stator temperature increases with each variable.

The temperature data collected along the length of the pump, as depicted in Figure 16, do not provide any useful information about the temperature profile of the pump's stator at the various test conditions. Although the uncertainty in the temperature measurements is quite low (maximally 0.93 °F), there is a great deal of scatter across the test conditions. The temperatures measured in the middle of the stator seem to change randomly, without regard to *GVF*, $P_s$, or $\Delta P$. The only observable pattern is that the temperatures in the middle of the stator lie between the temperatures at the two ends. This randomness suggests that either the temperature profile of the rubber stator did not consistently reach steady state conditions or that the thermocouple probes' placement varied enough to confound their measurements.

The axial temperature data seen in Figure 16 do demonstrate one important point: the thermocouples measured the temperature of the rubber stator, not the process fluid. Because the fluid's temperature increases with compression, it will be hottest at the discharge. However, Figure 16 clearly shows that the fluid temperature at the discharge is cooler than at any point along the stator, and this is true for all test conditions. The thermocouples could not have been measuring the fluid temperature, and so the temperature data must reflect the rubber's temperature within the stator.

For analysis, the maximum stator temperature for each test condition can be adequately modeled as a linear combination of $\Delta P$, $P_s$, and $GVF$; speed is not incorporated into the model because the pump is only tested at two speeds. The models generated with multiple linear regression on the full and half-speed temperature data are given in equations 22 and 23, respectively, while the temperatures predicted by those models are depicted in Figure 19 and Figure 20. Examining the equations, it is apparent that the temperature rise per change in any variable is always greater at full-speed than at half-speed. It is also worth noting that a change in $P_s$ has a larger effect on the stator temperature than a similar change in $\Delta P$.

The highest measured temperature within the rubber stator was 139.6 °F – 20.4 °F less than the maximum safe temperature of 160 °F specified by seepex. However, the tests were conducted over a relatively short period, up to three hours continuous run-time to collect a full set of data for a single speed. Operating for multiple hours at a very high $GVF$, such as 0.98, could allow the stator temperature to climb high enough to damage the pump. This is an avenue for further investigation, but the risk of destroying the pump is so great that such tests should only be conducted when all other research on this pump has concluded.

In addition to the temperature within the pump, the pressure measurements within the stator provide some insight into the pump's operation. By design, a PCP does not increase the pressure of its process fluid by itself. Rather, the increase in pressure across the pump is a natural result of forcing the mass of fluid within a pump cavity out of the pump's discharge line. Ideally, if the pump's stages were perfectly sealed, the fluid within a pump cavity would remain at the pump's suction pressure until that cavity opened to the discharge. Since the seals are not perfect, fluid leaks between the stages and the pressure within a cavity increases as the cavity moves from the suction to the discharge. This effect is clearly demonstrated in Figure 21 and Figure 22 where the

normalized pressure increases from nearly zero at the pressure tap nearest the suction, to as high as 0.8 at the tap nearest the discharge.

The first and most important finding this data reveals is that the normalized pressure at taps one and two are effectively coincident. This indicates that there is no appreciable inter-stage leakage in this region of the pump. Naturally, a PCP must be designed to minimize, if not eliminate, the net leakage from the suction to the discharge. This is typically accomplished by increasing the pump's length, which increases the number of stages, and thus the number of seals between the pump's suction and discharge. However, since there is no apparent leakage between the first and second pressure taps, it follows that the pump could be shortened by 25% with no loss in pumping efficiency. Indeed, the efficiency is likely to increase, given that the power needed to overcome the friction between the rotor and stator would be reduced.

The measurements at the third and fourth pressure taps reveal further insight into the pump's inter-stage leakage. A visual inspection of both Figure 21 and Figure 22 shows that the normalized pressure at these locations tends to increase with both $\Delta P$ and *GVF*. It is expected that increasing the pressure difference across the pump would increase the leakage through the inter-stage seals. However, the pressure's increase with *GVF* would suggest that water in the process fluid improves the quality of the seals. This effect should be investigated at substantially lower *GVF*s. If a higher liquid content does indeed improve the seals, then perhaps the length of the pump could be engineered to better suit the intended process fluid – longer pumps for higher *GVF* mixtures. Alternatively, a shortened pump could be designed to recirculate liquids exiting the pump back to the inlet to improve stage sealing – effectively increasing the *GVF* of only the fluid within the pump.

Figure 23 shows the load on the pump versus the $\Delta P$. This load is the mechanical power delivered by the motor to the pump. Any power losses downstream of the motor, at such places as the gearbox and seal, are included for in this measurement. The pump's power requirement is linear with $\Delta P$ and increases more with $\Delta P$ at full-speed than at half. Furthermore, the frictional load, visible at $\Delta P=0$, is nearly 32% higher at full-speed that at half. Finally, the load on the motor is independent of the *GVF*. While water does provide cooling to the stator, it adds no apparent lubricity to the pump. Further tests would be needed to determine if liquid does indeed lubricate the pump at significantly lower *GVF*s.

The pump's nominal efficiency, determined by dividing $Q\Delta P$ by $L$, is shown in Figure 24. As was pointed out in the previous section, this nominal efficiency calculation assumes that the process fluid is incompressible. By performing an incompressible flow analysis, the pump's hydraulic power can be estimated as $Q\Delta P$. The actual hydraulic power accounts for the air's compression, but the measurements collected for this investigation are insufficient to properly evaluate the hydraulic power. Another student is working to accurately measure and calculate the pump's hydraulic power.

As both $Q\Delta P$ and $L$ are dependent only on $\Delta P$, it naturally follows that their quotient would share this dependence. Since the $L$ is nonzero at zero $\Delta P$, the efficiency curve is non-linear, following an equation of the form,

$$\eta = \frac{a+b\Delta P}{c+d\Delta P} \tag{27}$$

where constants $a$, $b$, $c$, and $d$ are given in Table 3.

TABLE 3: EMPERICAL PARAMETERS FOR THE NOMINAL EFFICIENCY CURVE

| | a (hp) | b (hp/psi) | c (hp) | d (hp/psi) |
|---|---|---|---|---|
| Full-Speed | 0.0339 | 0.3214 | 45.52 | 0.2265 |
| Half-Speed | 0.0289 | 0.1669 | 34.52 | 0.1604 |

If the PCP were to be shortened, as is suggested above, the efficiency would increase due to the decreased frictional load, which is accounted for by $c$. Note that the contribution from $a$ can be neglected since $a<<c$. Thus, the pumps nominal efficiency can be estimated with the equation,

$$\eta = \frac{b\Delta P}{c+d\Delta P} \tag{28}$$

As with the hydraulic power, the uncertainty in the nominal efficiency is dominated by the uncertainty of $\Delta P$. The remarks on that quantity's uncertainty apply to the nominal efficiency as well.

# SUMMARY

The ultimate goal of these tests is to determine if a progressing cavity pump is suitable for pumping the multiphase output of a wet-gas well. If suitable, such a pump could boost the well's output by lowering the bore pressure, possibly to the extent of returning a "dead" well to production. Installing pumps at the several wet gas wells in a field could also allow the wells' output to be collected and separated at a single, large facility. While these same results can be achieved using a combination of separators, compressors, and liquid pumps, such an arrangement requires more equipment, is more likely to malfunction, and is potentially more costly than a single multiphase pump.

Although PCPs function well when pumping a liquid-gas mixture, they should not be used to pump only gas unless liquid is injected into the pump inlet. A PCP's rubber stator is lubricated and cooled by the liquid in the process fluid; running the pump dry risks overheating and destroying the rubber stator. This limitation could limit a PCP's usefulness in a wet gas field where the *GVF* of a well's output varies, although a liquid collection and recirculation system may overcome this limitation. To better gauge the PCPs suitability for wet-gas oilfield applications, this investigation explores the behavior of a PCP pumping liquid-gas mixtures with *GVF*s greater than 0.90.

A PCP manufactured by seepex, model no. BN 130-12, is tested pumping air-water mixtures with a *GVF* of 0.90, 0.92, 0.94, 0.96, and 0.98. The pumps suction pressure is held at 15, 30, and 45 psi while the discharge pressure is adjusted to produce a pressure difference across the pump between 0 and 150 psi (in 30 psi increments). The pump is tested at both half-speed and full-speed. During the tests, an array of flow meters, pressure transducers, and thermocouples measure the air and water flows entering the pump and the temperature and pressure in the pump's suction, discharge, and pump stator.

The most important finding of this investigation is that the pump's stator did not overheat at any of the test conditions. The maximum temperature measured within the stator was 139.6 °F – 20.4 °F less than the maximum operating temperature of 160 °F. These tests only required approximately three hours of total running time, so it is possible that the stator may still overheat if the pump runs a 0.98 *GVF* mixture for several hours.

While some of this investigation's findings are intuitive (*e.g.*, increasing $\Delta P$ increases the stator temperature), others are quite unexpected. The pump's flow rate, Q, is independent of both $\Delta P$ and *GVF*, but increases with the suction pressure, $P_s$. This suggests that increasing the pump's suction pressure increases the size of the pump's cavities by deforming the rubber stator. By extension, $Q\Delta P$, the pump's hydraulic power assuming incompressible flow, is independent of *GVF* and may be treated as independent of $P_s$ because of $\Delta P$'s dominance.

The load on the pump increases linearly with $\Delta P$ but is independent of $P_s$ and *GVF*. While water does cool the stator, it adds no measurable lubricity to the pump. Comparing the load on the pump at half and full speed shows that the pump's frictional load increases with speed.

The pressure measurements within the stator demonstrate that leakage between pump stages increases with both $\Delta P$ and *GVF*. In the high *GVF* regions investigated, the amount of liquid travelling significantly affects the quality of the seal between the steel rotor and rubber stator. If a PCP is expected to operate with dry gasses at times, it may be advantageous to install a liquid collection and recirculation system to improve the pump's efficiency.

Additionally, the pressure taps along the stator observed no appreciable leakage between the two locations nearest the stator, even at the highest $\Delta P$ and *GVF* conditions. The pump can be shortened by as much as 25% without decreasing the pump's flow rate. Indeed shortening the pump ought to increase the pump's efficiency by reducing its frictional loading.

These many observations provide a valuable understanding of PCP behavior when pumping high *GVF* fluids. However, some further testing would greatly improve this understanding. This investigation suggests three tests that would be especially helpful in determining a PCP's suitability for pumping in a wet-gas field. First, these same tests should be repeated with a hydrocarbon blend similar to what might be encountered in the field. The different lubricity, thermal conductivity, specific heat, *etc.*, will undoubtedly produce significantly different results. Second the pump should be tested with fluids of *GVF* between 0 and 0.90 to determine if the parameters independent of *GVF* in these tests remain so. Finally, the pump needs to be tested running a mixture with *GVF* of 0.98 or higher until the stator reaches the maximum operating temperature. This last test risks destroying the pump's stator and should be performed only as a final investigation, but it is necessary to determine the most extreme operating condition the pump can sustain.

# REFERENCES

[1] Lehman, M., 2004, "Progressing Cavity Pumps in Oil and Gas Production," World Pumps, 2004(457), pp. 20-22.

[2] Mirza, K. Z. and Wild, A.G., 1997, "Key Advantages of the Progressing Cavity Pump in Multiphase Transfer Applications," Proceedings of the SPE Annual Technical Conference and Exhibition, San Antonio, TX, October 5-8.

[3] Moineau, R., 1930, "A New Capsulism," Ph.D. thesis, University of Paris, Paris.

[4] Vetter, G., and Paluchowski, D., 1997, "Modeling of NPSHR for Progresseing Cavity Pumps," ASME Fluids Engineering Division Summer Meeting, Vancouver

[5] Gamboa, J., Olivet, A., Espin, S., 2003, "New Approach for Modeling Progressive Cavity Pumps Performance," Proceedings of the SPE Annual Technical Conference and Exhibition, Denver, CO, October 5-8

[6] Robello, S., Saveth, K., 1998, "Progressing Cavity Pump (PCP): New Performance Equations for Optimal Design," Proceedings of the Permian Basin Oil & Gas Recovery Conference, Midland, TX, March 23-26

[7] Bratu, C., 2005, "Progressing Cavity Pump (PCP) Behavior in Multiphase Conditions," Proceedings - SPE Annual Technical Conference and Exhibition, Dallas, TX, October 9-12

# APPENDIX

Data processing is handled by eight MATLAB scripts and 2 MATLAB functions. The scripts extract and process raw data files from LabVIEW and generate the plots in this document. The functions are used to perform single and multiple linear regression. This appendix shows these codes in their entirety and explains their operation.

## DATA EXTRACTION FOR 30 AND 60 HZ TESTS

This script extracts data from the raw text files generated by LabVIEW during the half-speed tests. Each column in the text file contains the data from a single sensor. The mean and standard deviation is computed and recorded for each test. This script also uses measurements to compute the volumetric air and water flow rates into the pump. The variables are recorded and saved to a ".mat" file that can be opened by other scripts for further processing and plotting. Each data variable is given a standard prefix with a word or abbreviation followed by a number. The word indicates the variables type, such an average of raw data (Mean), the standard deviation (Std), or a value calculated from measured values (Calc). The number is either 30 or 60 indicating that the data is from a half or full speed test, respectively.

```
clc
clear all

GVF=[90,92,94,96,98];

SuctPress=[15,30,45];
for l=1:length(GVF)            %Gas Volume Fraction
    for k=1:length(SuctPress)       %Suction Pressure
        for i=1:6   %DeltaP
%Import Data File - must be in same directory as SeepexScript
            filename = sprintf('%d_%d_%d.lvm',GVF(l),SuctPress(k),i-1);
            clear A
            A=importdata(filename, '\t', 24);
            filename

%Average Flowmeter Data - one entry every hundred rows
            Raw30_Flowmeter1reading = 0;
            Raw30_Flowmeter2reading = 0;
            Raw30_Flowmeter_Air_reading = 0;

            for p=1:50;
```

```
            Raw30_Flowmeter1reading(p) = A.data((p*100)+1,22);
            Raw30_Flowmeter2reading(p) = A.data((p*100)+1,23);
            Raw30_Flowmeter_Air_reading(p) = A.data((p*100)+1,24);
        end

        Mean30_Flowmeter1reading(l,k,i) = mean(Raw30_Flowmeter1reading);
        Mean30_Flowmeter2reading(l,k,i) = mean(Raw30_Flowmeter2reading);
        Mean30_Flowmeter_Air_reading(l,k,i) = mean(Raw30_Flowmeter_Air_reading);
        Std30_Flowmeter1reading(l,k,i) = std(Raw30_Flowmeter1reading);
        Std30_Flowmeter2reading(l,k,i) = std(Raw30_Flowmeter2reading);
        Std30_Flowmeter_Air_reading(l,k,i) = std(Raw30_Flowmeter_Air_reading);


%Temperature and Pressure Readings - not on stator
        Mean30_T_Air(l,k,i)= mean(A.data(:,2));
        Mean30_T_Pump(l,k,i)= mean(A.data(:,3));
        Mean30_T_Water(l,k,i)= mean(A.data(:,4));
        Mean30_T_Outlet(l,k,i)= mean(A.data(:,5));
        Mean30_Load(l,k,i) = (mean(A.data(:,7))*10);
        Mean30_P_Air(l,k,i) =(mean(A.data(:,8))-1)*50;
        Mean30_P_Pump(l,k,i) = (mean(A.data(:,9))-1)*15;
        Mean30_P_Outlet(l,k,i) =(mean(A.data(:,10))-1)*75;
        Calc30_Delta_P(l,k,i)= Mean30_P_Outlet(l,k,i)-Mean30_P_Pump(l,k,i);

        Std30_T_Air(l,k,i)= std(A.data(:,2));
        Std30_T_Pump(l,k,i)= std(A.data(:,3));
        Std30_T_Water(l,k,i)= std(A.data(:,4));
        Std30_T_Outlet(l,k,i)= std(A.data(:,5));
        Std30_Load(l,k,i) = std((A.data(:,7))*10));
        Std30_P_Air(l,k,i) =std((A.data(:,8))-1)*50);
        Std30_P_Pump(l,k,i) = std((A.data(:,9))-1)*15);
        Std30_P_Outlet(l,k,i) =std((A.data(:,10))-1)*75);


%Calculated Flowrates
        Calc30_Vdot_Air_in(l,k,i) = (((Mean30_Flowmeter_Air_reading(l,k,i)*60)/2385)/
0.133680556);
        Calc30_Vdot_Water(l,k,i)                                                    =
Mean30_Flowmeter1reading(l,k,i)*60/911+Mean30_Flowmeter2reading(l,k,i)*60/116.5;
        Calc30_Vdot_Air_Pump(l,k,i)                                                 =
((Mean30_P_Air(l,k,i)+14.7)*(Mean30_T_Pump(l,k,i)+459.67)*Calc30_Vdot_Air_in(l,k,i))/((Me
an30_P_Pump(l,k,i)+14.7)*(Mean30_T_Air(l,k,i)+459.67));
        Calc30_GVF(l,k,i)                                                           =
Calc30_Vdot_Air_Pump(l,k,i)/(Calc30_Vdot_Air_Pump(l,k,i)+Calc30_Vdot_Water(l,k,i));

Calc30_Vdot_Total(l,k,i)=Calc30_Vdot_Air_Pump(l,k,i)+Calc30_Vdot_Water(l,k,i);

%Temperature and Pressure on Pump Stator
        for j=1:4
            AxPos30(j,k,l)=j;
            Mean30_AxPres(i,j,k,l)=(mean(A.data(:,10+j))-1)*50;
```

```
                Mean30_AxTemp(i,j,k,l)=(mean(A.data(:,14+j)));
                Std30_AxPres(i,j,k,l)=std(((A.data(:,10+j))-1)*50);
                Std30_AxTemp(i,j,k,l)=std((A.data(:,14+j)));
            end

% Tachometer Data
             Mean30_Pump_Freq(l,k,i)=GetAvgFreq(A.data(:,21));
        end
    end
end

save('ExtractedData30','AxPos30','Mean30*','Calc30*','Std30*')
```

This script extracts data from the raw text files generated by LabVIEW during the full-speed tests. It is identical to the previous script except that exported variables carry a 60 in the prefix to denote full-speed.

```
clc
clear all

GVF=[90,92,94,96,98];

SuctPress=[15,30,45];
for l=1:length(GVF)            %Gas Volume Fraction
    for k=1:length(SuctPress)       %Suction Pressure
        for i=1:6   %DeltaP
%Import Data File - must be in same directory as SeepexScript
            filename = sprintf('%d_%d_%d.lvm',GVF(l),SuctPress(k),i-1);
            clear A
            A=importdata(filename, '\t', 24);
            filename

%Average Flowmeter Data - one entry every hundred rows
            Raw60_Flowmeter1reading = 0;
            Raw60_Flowmeter2reading = 0;
            Raw60_Flowmeter_Air_reading = 0;

            for p=1:50
                Raw60_Flowmeter1reading(p) = A.data((p*100)+1,22);
                Raw60_Flowmeter2reading(p) = A.data((p*100)+1,23);
                Raw60_Flowmeter_Air_reading(p) = A.data((p*100)+1,24);
            end

            Mean60_Flowmeter1reading(l,k,i) = mean(Raw60_Flowmeter1reading);
            Mean60_Flowmeter2reading(l,k,i) = mean(Raw60_Flowmeter2reading);
            Mean60_Flowmeter_Air_reading(l,k,i) = mean(Raw60_Flowmeter_Air_reading);
            Std60_Flowmeter1reading(l,k,i) = std(Raw60_Flowmeter1reading);
            Std60_Flowmeter2reading(l,k,i) = std(Raw60_Flowmeter2reading);
            Std60_Flowmeter_Air_reading(l,k,i) = std(Raw60_Flowmeter_Air_reading);


%Temperature and Pressure Readings - not on stator
            Mean60_T_Air(l,k,i)= mean(A.data(:,2));
            Mean60_T_Pump(l,k,i)= mean(A.data(:,3));
            Mean60_T_Water(l,k,i)= mean(A.data(:,4));
            Mean60_T_Outlet(l,k,i)= mean(A.data(:,5));
            Mean60_Load(l,k,i) = (mean(A.data(:,7))*10);
            Mean60_P_Air(l,k,i) =(mean(A.data(:,8))-1)*50;
            Mean60_P_Pump(l,k,i) = (mean(A.data(:,9))-1)*15;
            Mean60_P_Outlet(l,k,i) =(mean(A.data(:,10))-1)*75;
            Calc60_Delta_P(l,k,i)= Mean60_P_Outlet(l,k,i)-Mean60_P_Pump(l,k,i);
```

```
            Std60_T_Air(l,k,i)= std(A.data(:,2));
            Std60_T_Pump(l,k,i)= std(A.data(:,3));
            Std60_T_Water(l,k,i)= std(A.data(:,4));
            Std60_T_Outlet(l,k,i)= std(A.data(:,5));
            Std60_Load(l,k,i) = std(((A.data(:,7))*10));
            Std60_P_Air(l,k,i) =std(((A.data(:,8))-1)*50);
            Std60_P_Pump(l,k,i) = std(((A.data(:,9))-1)*15);
            Std60_P_Outlet(l,k,i) =std(((A.data(:,10))-1)*75);


%Calculated Flowrates
            Calc60_Vdot_Air_in(l,k,i) = (((Mean60_Flowmeter_Air_reading(l,k,i)*60)/2385)/
0.133680556);
            Calc60_Vdot_Water(l,k,i)                                                    =
Mean60_Flowmeter1reading(l,k,i)*60/911+Mean60_Flowmeter2reading(l,k,i)*60/116.5;
            Calc60_Vdot_Air_Pump(l,k,i)                                                 =
((Mean60_P_Air(l,k,i)+14.7)*(Mean60_T_Pump(l,k,i)+459.67)*Calc60_Vdot_Air_in(l,k,i))/((Me
an60_P_Pump(l,k,i)+14.7)*(Mean60_T_Air(l,k,i)+459.67));
            Calc60_GVF(l,k,i)                                                           =
Calc60_Vdot_Air_Pump(l,k,i)/(Calc60_Vdot_Air_Pump(l,k,i)+Calc60_Vdot_Water(l,k,i));

Calc60_Vdot_Total(l,k,i)=Calc60_Vdot_Air_Pump(l,k,i)+Calc60_Vdot_Water(l,k,i);

%Temperature and Pressure on Pump Stator
            for j=1:4
                AxPos60(j,k,l)=j;
                Mean60_AxPres(i,j,k,l)=(mean(A.data(:,10+j))-1)*50;
                Mean60_AxTemp(i,j,k,l)=(mean(A.data(:,14+j)));
                Std60_AxPres(i,j,k,l)=std(((A.data(:,10+j))-1)*50);
                Std60_AxTemp(i,j,k,l)=std((A.data(:,14+j)));
            end

% Tachometer Data
             Mean60_Pump_Freq(l,k,i)=GetAvgFreq(A.data(:,21));
        end
    end
end

save('ExtractedData60','AxPos60','Mean60*','Calc60*','Std60*')
```

UNCERTAINTY CALCULATIONS

This script calculates the uncertainty of each measured and derived value with a 95% confidence interval. Uncertainties of measured values carry the prefix U, while those of derived values use UC. The UM prefix denotes the maximum uncertainty for a variable across all test conditions. Uncertainty variables are saved to a ".mat" file that can be loaded into other scripts.

```
% Calculated uncertainties
clc
clear all

load 'ExtractedData30.mat'
load 'ExtractedData60.mat'

%HALF SPEED----------------
U30_Flowmeter1reading=2*Std30_Flowmeter1reading;
U30_Flowmeter2reading=2*Std30_Flowmeter2reading;
U30_Flowmeter_Air_reading=2*Std30_Flowmeter_Air_reading;
U30_Load=Std30_Load*2;
U30_P_Air=2*Std30_P_Air;
U30_P_Outlet=2*Std30_P_Outlet;
U30_P_Pump=2*Std30_P_Pump;
U30_T_Air=2*Std30_T_Air;
U30_T_Outlet=2*Std30_T_Outlet;
U30_T_Pump=2*Std30_T_Pump;

UM30_Flowmeter1reading=[max(max(max(2*Std30_Flowmeter1reading))),mean(mean(mean(2*Std30_F
lowmeter1reading)))];
UM30_Flowmeter2reading=[max(max(max(2*Std30_Flowmeter2reading))),mean(mean(mean(2*Std30_F
lowmeter2reading)))];
UM30_Flowmeter_Air_reading=[max(max(max(2*Std30_Flowmeter_Air_reading))),mean(mean(mean(2
*Std30_Flowmeter_Air_reading)))];
UM30_Load=[max(max(max(Std30_Load*2))),mean(mean(mean(Std30_Load*2)))];
UM30_P_Air=[max(max(max(2*Std30_P_Air))),mean(mean(mean(2*Std30_P_Air)))];
UM30_P_Outlet=[max(max(max(2*Std30_P_Outlet))),mean(mean(mean(2*Std30_P_Outlet)))];
UM30_P_Pump=[max(max(max(2*Std30_P_Pump))),mean(mean(mean(2*Std30_P_Pump)))];
UM30_T_Air=[max(max(max(2*Std30_T_Air))),mean(mean(mean(2*Std30_T_Air)))];
UM30_T_Outlet=[max(max(max(2*Std30_T_Outlet))),mean(mean(mean(2*Std30_T_Outlet)))];
UM30_T_Pump=[max(max(max(2*Std30_T_Pump))),mean(mean(mean(2*Std30_T_Pump)))];

empty=zeros(5,3,6);
UC30_DeltaP=empty;
UC30_Vdot_Water=empty;
UC30_Vdot_Air_in=empty;
UC30_Vdot_Air_Pump=empty;
UC30_GVF=empty;
```

```
UC30_Vdot_Total=empty;
UC30_QdP=empty;



for i=1:5
    for j=1:3
        for k=1:6

            UC30_DeltaP(i,j,k)=(U30_P_Pump(i,j,k)^2+U30_P_Outlet(i,j,k)^2)^.5;

UC30_Vdot_Water(i,j,k)=((U30_Flowmeter1reading(i,j,k)*60/911)^2+(U30_Flowmeter2reading(i,
j,k)*60/116.5)^2)^.5;
            UC30_Vdot_Air_in(i,j,k)=U30_Flowmeter_Air_reading(i,j,k)*60/(2385*0.133681);

p1=((U30_P_Air(i,j,k))*(Mean30_T_Pump(i,j,k)+459.67)*Calc30_Vdot_Air_in(i,j,k))/((Mean30_
P_Pump(i,j,k)+14.7)*(Mean30_T_Air(i,j,k)+459.67));

p2=((Mean30_P_Air(i,j,k)+14.7)*(U30_T_Pump(i,j,k))*Calc30_Vdot_Air_in(i,j,k))/((Mean30_P_
Pump(i,j,k)+14.7)*(Mean30_T_Air(i,j,k)+459.67));

p3=((Mean30_P_Air(i,j,k)+14.7)*(Mean30_T_Pump(i,j,k)+459.67)*UC30_Vdot_Air_in(i,j,k))/((M
ean30_P_Pump(i,j,k)+14.7)*(Mean30_T_Air(i,j,k)+459.67));

p4=U30_P_Pump(i,j,k)*((Mean30_P_Air(i,j,k)+14.7)*(Mean30_T_Pump(i,j,k)+459.67)*Calc30_Vdo
t_Air_in(i,j,k))/((Mean30_P_Pump(i,j,k)+14.7)^2*(Mean30_T_Air(i,j,k)+459.67));

p5=U30_T_Air(i,j,k)*((Mean30_P_Air(i,j,k)+14.7)*(Mean30_T_Pump(i,j,k)+459.67)*Calc30_Vdot
_Air_in(i,j,k))/((Mean30_P_Pump(i,j,k)+14.7)*(Mean30_T_Air(i,j,k)+459.67)^2);
            UC30_Vdot_Air_Pump(i,j,k)=(p1^2+p2^2+p3^3+p4^2+p5^2)^.5;

UC30_Vdot_Total(i,j,k)=(UC30_Vdot_Air_Pump(i,j,k)^2+UC30_Vdot_Water(i,j,k)^2)^.5;

UC30_GVF(i,j,k)=((UC30_Vdot_Air_Pump(i,j,k)/Calc30_Vdot_Total(i,j,k))^2+(UC30_Vdot_Total(
i,j,k)*Calc30_Vdot_Air_Pump(i,j,k)/Calc30_Vdot_Total(i,j,k)^2)^2)^.5;

UC30_QdP(i,j,k)=((UC30_Vdot_Total(i,j,k)*Calc30_Delta_P(i,j,k)/1714.3)^2+(UC30_DeltaP(i,j
,k)*Calc30_Vdot_Total(i,j,k)/1714.3)^2)^.5;
        end
    end
end

UM30_DeltaP=[max(max(max(UC30_DeltaP))),mean(mean(mean(UC30_DeltaP)))];
UM30_Vdot_Water=[max(max(max(UC30_Vdot_Water))),mean(mean(mean(UC30_Vdot_Water)))];
UM30_Vdot_Air_in=[max(max(max(UC30_Vdot_Air_in))),mean(mean(mean(UC30_Vdot_Air_in)))];
UM30_Vdot_Air_Pump=[max(max(max(UC30_Vdot_Air_Pump))),mean(mean(mean(UC30_Vdot_Air_Pump))
)];
UM30_GVF=[max(max(max(UC30_GVF))),mean(mean(mean(UC30_GVF)))];
UM30_Vdot_Total=[max(max(max(UC30_Vdot_Total))),mean(mean(mean(UC30_Vdot_Total)))];
UM30_QdP=[max(max(max(UC30_QdP))),mean(mean(mean(UC30_QdP)))];
```

```
%FULL SPEED-------------


U60_Flowmeter1reading=2*Std60_Flowmeter1reading;
U60_Flowmeter2reading=2*Std60_Flowmeter2reading;
U60_Flowmeter_Air_reading=2*Std60_Flowmeter_Air_reading;
U60_Load=Std60_Load*2;
U60_P_Air=2*Std60_P_Air;
U60_P_Outlet=2*Std60_P_Outlet;
U60_P_Pump=2*Std60_P_Pump;
U60_T_Air=2*Std60_T_Air;
U60_T_Outlet=2*Std60_T_Outlet;
U60_T_Pump=2*Std60_T_Pump;

UM60_Flowmeter1reading=[max(max(max(2*Std60_Flowmeter1reading))),mean(mean(mean(2*Std60_F
lowmeter1reading)))];
UM60_Flowmeter2reading=[max(max(max(2*Std60_Flowmeter2reading))),mean(mean(mean(2*Std60_F
lowmeter2reading)))];
UM60_Flowmeter_Air_reading=[max(max(max(2*Std60_Flowmeter_Air_reading))),mean(mean(mean(2
*Std60_Flowmeter_Air_reading)))];
UM60_Load=[max(max(max(Std60_Load*2))),mean(mean(mean(Std60_Load*2)))];
UM60_P_Air=[max(max(max(2*Std60_P_Air))),mean(mean(mean(2*Std60_P_Air)))];
UM60_P_Outlet=[max(max(max(2*Std60_P_Outlet))),mean(mean(mean(2*Std60_P_Outlet)))];
UM60_P_Pump=[max(max(max(2*Std60_P_Pump))),mean(mean(mean(2*Std60_P_Pump)))];
UM60_T_Air=[max(max(max(2*Std60_T_Air))),mean(mean(mean(2*Std60_T_Air)))];
UM60_T_Outlet=[max(max(max(2*Std60_T_Outlet))),mean(mean(mean(2*Std60_T_Outlet)))];
UM60_T_Pump=[max(max(max(2*Std60_T_Pump))),mean(mean(mean(2*Std60_T_Pump)))];

empty=zeros(5,3,6);
UC60_DeltaP=empty;
UC60_Vdot_Water=empty;
UC60_Vdot_Air_in=empty;
UC60_Vdot_Air_Pump=empty;
UC60_GVF=empty;
UC60_Vdot_Total=empty;
UC60_QdP=empty;



for i=1:5
    for j=1:3
        for k=1:6

            UC60_DeltaP(i,j,k)=(U60_P_Pump(i,j,k)^2+U60_P_Outlet(i,j,k)^2)^.5;

UC60_Vdot_Water(i,j,k)=((U60_Flowmeter1reading(i,j,k)*60/911)^2+(U60_Flowmeter2reading(i,
j,k)*60/116.5)^2)^.5;
            UC60_Vdot_Air_in(i,j,k)=U60_Flowmeter_Air_reading(i,j,k)*60/(2385*0.133681);

p1=((U60_P_Air(i,j,k))*(Mean60_T_Pump(i,j,k)+459.67)*Calc60_Vdot_Air_in(i,j,k))/((Mean60_
```

```
P_Pump(i,j,k)+14.7)*(Mean60_T_Air(i,j,k)+459.67));

p2=((Mean60_P_Air(i,j,k)+14.7)*(U60_T_Pump(i,j,k))*Calc60_Vdot_Air_in(i,j,k))/((Mean60_P_
Pump(i,j,k)+14.7)*(Mean60_T_Air(i,j,k)+459.67));

p3=((Mean60_P_Air(i,j,k)+14.7)*(Mean60_T_Pump(i,j,k)+459.67)*UC60_Vdot_Air_in(i,j,k))/((M
ean60_P_Pump(i,j,k)+14.7)*(Mean60_T_Air(i,j,k)+459.67));

p4=U60_P_Pump(i,j,k)*((Mean60_P_Air(i,j,k)+14.7)*(Mean60_T_Pump(i,j,k)+459.67)*Calc60_Vdo
t_Air_in(i,j,k))/((Mean60_P_Pump(i,j,k)+14.7)^2*(Mean60_T_Air(i,j,k)+459.67));

p5=U60_T_Air(i,j,k)*((Mean60_P_Air(i,j,k)+14.7)*(Mean60_T_Pump(i,j,k)+459.67)*Calc60_Vdot
_Air_in(i,j,k))/((Mean60_P_Pump(i,j,k)+14.7)*(Mean60_T_Air(i,j,k)+459.67)^2);
            UC60_Vdot_Air_Pump(i,j,k)=(p1^2+p2^2+p3^3+p4^2+p5^2)^.5;

UC60_Vdot_Total(i,j,k)=(UC60_Vdot_Air_Pump(i,j,k)^2+UC60_Vdot_Water(i,j,k)^2)^.5;

UC60_GVF(i,j,k)=((UC60_Vdot_Air_Pump(i,j,k)/Calc60_Vdot_Total(i,j,k))^2+(UC60_Vdot_Total(
i,j,k)*Calc60_Vdot_Air_Pump(i,j,k)/Calc60_Vdot_Total(i,j,k)^2)^2)^.5;

UC60_QdP(i,j,k)=((UC60_Vdot_Total(i,j,k)*Calc60_Delta_P(i,j,k)/1714.3)^2+(UC60_DeltaP(i,j
,k)*Calc60_Vdot_Total(i,j,k)/1714.3)^2)^.5;
        end
    end
end

UM60_DeltaP=[max(max(max(UC60_DeltaP))),mean(mean(mean(UC60_DeltaP)))];
UM60_Vdot_Water=[max(max(max(UC60_Vdot_Water))),mean(mean(mean(UC60_Vdot_Water)))];
UM60_Vdot_Air_in=[max(max(max(UC60_Vdot_Air_in))),mean(mean(mean(UC60_Vdot_Air_in)))];
UM60_Vdot_Air_Pump=[max(max(max(UC60_Vdot_Air_Pump))),mean(mean(mean(UC60_Vdot_Air_Pump))
)];
UM60_GVF=[max(max(max(UC60_GVF))),mean(mean(mean(UC60_GVF)))];
UM60_Vdot_Total=[max(max(max(UC60_Vdot_Total))),mean(mean(mean(UC60_Vdot_Total)))];
UM60_QdP=[max(max(max(UC60_QdP))),mean(mean(mean(UC60_QdP)))];
save('UncertaintyData','UM30*','UM60*','U30*','U60*','UC30*','UC60*')
```

SPEED COMPARISON PLOTS

This script generates all plots containing both 30 Hz and 60 Hz data. The plots are automatically saved as ".gif" files.

```
clc
clear all
close all

load('ExtractedData30')
load('ExtractedData60')
load('ExtractedDataPow30')
load('ExtractedDataPow60')
load('UncertaintyData')

bogusx=[-5,165];
bogusy=[-5,-5];

Flow_Ratio=Calc30_Vdot_Total./Calc60_Vdot_Total;
P_suct=[15,30,45];

%Flow ratio by suction pressure
figure
clear xd yd err
for s=1:6           %DeltaP Loop
    for t=1:5       %GVF Loop
        yd1(s,t)=Flow_Ratio(t,1,s);
        xd1(s,t)=15;
        yd2(s,t)=Flow_Ratio(t,2,s);
        xd2(s,t)=30;
        yd3(s,t)=Flow_Ratio(t,3,s);
        xd3(s,t)=45;
    end
end

[Ratio_Coef_lin,Pred_Ratio_lin,R_sq_Ratio_lin]=regress_1var_1coef([yd1,yd2,yd3],[xd1,xd2,
xd3]);

hold on

plot([min(min([xd1,xd2,xd3])),max(max([xd1,xd2,xd3]))],[min(min(Pred_Ratio_lin)),max(max(
Pred_Ratio_lin))],'color','k','linewidth',1)
plot(xd1(:,1),yd1(:,1),'xk','LineStyle','none')
plot(xd1(:,2),yd1(:,2),'xk','LineStyle','none')
plot(xd1(:,3),yd1(:,3),'xk','LineStyle','none')
plot(xd1(:,4),yd1(:,4),'xk','LineStyle','none')
plot(xd1(:,5),yd1(:,5),'xk','LineStyle','none')
plot(xd2(:,1),yd2(:,1),'xk','LineStyle','none')
plot(xd2(:,2),yd2(:,2),'xk','LineStyle','none')
```

```
plot(xd2(:,3),yd2(:,3),'xk','LineStyle','none')
plot(xd2(:,4),yd2(:,4),'xk','LineStyle','none')
plot(xd2(:,5),yd2(:,5),'xk','LineStyle','none')
plot(xd3(:,1),yd3(:,1),'xk','LineStyle','none')
plot(xd3(:,2),yd3(:,2),'xk','LineStyle','none')
plot(xd3(:,3),yd3(:,3),'xk','LineStyle','none')
plot(xd3(:,4),yd3(:,4),'xk','LineStyle','none')
plot(xd3(:,5),yd3(:,5),'xk','LineStyle','none')

xlim([13,46]);
ylim([0,1]);
xlabel('P_s_u_c_t_i_o_n (psi)');
ylabel('Q_3_0 / Q_6_0');
l1=sprintf('%s                %4.4f                %s                %4.4f%s','Q_3_0/Q_6_0
=',Ratio_Coef_lin(1),'+',Ratio_Coef_lin(2),'(P_s_u_c_t_i_o_n)');
text(25,.4,l1);

hold off

set(gcf,'PaperPosition',[1.915,3.75,4.67,3.5])
saveas(gcf,'Flow_Ratio_Plot.jpg')

%Q\DeltaP by DeltaP at each Suction Pressure and GVF-----------------------
figure
clear xd yd err
for s=1:6           %DeltaP Loop
    for t=1:5       %GVF Loop
        yd61(s,t)=Calc60_Vdot_Total(t,1,s)*Calc60_Delta_P(t,1,s)/1714.3;
        xd61(s,t)=Calc60_Delta_P(t,1,s);
        ed61(s,t)=UC60_QdP(t,1,s);
        yd62(s,t)=Calc60_Vdot_Total(t,2,s)*Calc60_Delta_P(t,2,s)/1714.3;
        xd62(s,t)=Calc60_Delta_P(t,2,s);
        ed62(s,t)=UC60_QdP(t,2,s);
        yd63(s,t)=Calc60_Vdot_Total(t,3,s)*Calc60_Delta_P(t,3,s)/1714.3;
        xd63(s,t)=Calc60_Delta_P(t,3,s);
        ed63(s,t)=UC30_QdP(t,3,s);
        yd31(s,t)=Calc30_Vdot_Total(t,1,s)*Calc30_Delta_P(t,1,s)/1714.3;
        xd31(s,t)=Calc30_Delta_P(t,1,s);
        ed31(s,t)=UC30_QdP(t,1,s);
        yd32(s,t)=Calc30_Vdot_Total(t,2,s)*Calc30_Delta_P(t,2,s)/1714.3;
        xd32(s,t)=Calc30_Delta_P(t,2,s);
        ed32(s,t)=UC30_QdP(t,2,s);
        yd33(s,t)=Calc30_Vdot_Total(t,3,s)*Calc30_Delta_P(t,3,s)/1714.3;
        xd33(s,t)=Calc30_Delta_P(t,3,s);
        ed33(s,t)=UC30_QdP(t,3,s);

    end
end

[VdotDeltaP60_Coef_lin,Pred_VdotDeltaP60_lin,R_sq_VdotDeltaP60_lin]=regress_1var_1coef([y
d61,yd62,yd63],[xd61,xd62,xd63]);
```

```
[VdotDeltaP30_Coef_lin,Pred_VdotDeltaP30_lin,R_sq_VdotDeltaP30_lin]=regress_1var_1coef([y
d31,yd32,yd33],[xd31,xd32,xd33]);

hold on

plot(bogusx,bogusy,'kx')
plot(bogusx,bogusy,'ko')

plot([min(min([xd61,xd62,xd63])),max(max([xd61,xd62,xd63]))],[min(min(Pred_VdotDeltaP60_l
in)),max(max(Pred_VdotDeltaP60_lin))],'color','k','linewidth',1)
plot([min(min([xd31,xd32,xd33])),max(max([xd31,xd32,xd33]))],[min(min(Pred_VdotDeltaP30_l
in)),max(max(Pred_VdotDeltaP30_lin))],'color','k','linewidth',1)

errorbar(xd61,yd61,ed61,'xk','LineStyle','none')
errorbar(xd62,yd62,ed62,'xk','LineStyle','none')
errorbar(xd63,yd63,ed63,'xk','LineStyle','none')
errorbar(xd31,yd31,ed31,'ko','LineStyle','none')
errorbar(xd32,yd32,ed32,'ko','LineStyle','none')
errorbar(xd33,yd33,ed33,'ko','LineStyle','none')

xlim([0,160]);
ylim([0,60]);
xlabel('\DeltaP (psi)');
ylabel('Q\DeltaP (hp)');
legend('60 Hz','30 Hz','Location','NorthWest')
l1=sprintf('%s      %4.4f      %s      %4.4f%s      \n            %s%4.3f','Q_6_0\DeltaP
=',VdotDeltaP60_Coef_lin(1),'+',VdotDeltaP60_Coef_lin(2),'(\DeltaP)','R^2               =
',R_sq_VdotDeltaP60_lin);
text(35,40,l1);
l2=sprintf('%s      %4.4f      %s      %4.4f%s      \n            %s%4.3f','Q_3_0\DeltaP
=',VdotDeltaP30_Coef_lin(1),'+',VdotDeltaP30_Coef_lin(2),'(\DeltaP)','R^2               =
',R_sq_VdotDeltaP30_lin);
text(80,8,l2);

hold off

set(gcf,'PaperPosition',[1.915,3.75,4.67,3.5])
saveas(gcf,'QDeltaP_Plot.jpg')

%Flowrate versus Suction Pressure------------------------------
figure
clear xd yd zd
for s=1:6           %DeltaP Loop
    for t=1:5       %GVF Loop
        for u=1:3   %Suction Pressure Loop
            Vdot60(t+(u-1)*5,s)=Calc60_Vdot_Total(t,u,s);
            EVdot60(t+(u-1)*5,s)=UC60_Vdot_Total(t,u,s);
            xd60(t+(u-1)*5,s)=Mean60_P_Pump(t,u,s);
            Vdot30(t+(u-1)*5,s)=Calc30_Vdot_Total(t,u,s);
            EVdot30(t+(u-1)*5,s)=UC30_Vdot_Total(t,u,s);
            xd30(t+(u-1)*5,s)=Mean30_P_Pump(t,u,s);
```

```
        end
    end
end

[Vdot60_Coef_lin,Pred_Vdot60_lin,R_sq_Vdot60_lin]=regress_1var_1coef(Vdot60,xd60);
[Vdot30_Coef_lin,Pred_Vdot30_lin,R_sq_Vdot30_lin]=regress_1var_1coef(Vdot30,xd30);

hold on

plot(bogusx,bogusy,'kx')
plot(bogusx,bogusy,'ko')

plot(xd60,Vdot60,'LineStyle','none','Marker','x','MarkerEdgeColor','k')
plot([min(min(xd60)),max(max(xd60))],[min(min(Pred_Vdot60_lin)),max(max(Pred_Vdot60_lin))
],'color','k','linewidth',1)
plot(xd30,Vdot30,'LineStyle','none','Marker','o','MarkerEdgeColor','k')
plot([min(min(xd30)),max(max(xd30))],[min(min(Pred_Vdot30_lin)),max(max(Pred_Vdot30_lin))
],'color','k','linewidth',1)

ylim([200,575])
xlim([13,46])
xlabel('Suction Pressure (psi)')
ylabel('Volumeteric Flow Rate (gpm)')
legend('60 Hz','30 Hz','Location','West')
l1=sprintf('%s       %4.2f       %s       %4.4f%s       \n               %s%4.4f','Q_6_0
=',Vdot60_Coef_lin(1),'+',Vdot60_Coef_lin(2),'(P_s_u_c_t_i_o_n)','R^2                    =
',R_sq_Vdot60_lin);
text(25,510,l1)
l2=sprintf('%s       %4.2f       %s       %4.4f%s       \n               %s%4.4f','Q_3_0
=',Vdot30_Coef_lin(1),'+',Vdot30_Coef_lin(2),'(P_s_u_c_t_i_o_n)','R^2                    =
',R_sq_Vdot30_lin);
text(25,250,l2)
hold off

set(gcf,'PaperPosition',[1.915,3.75,4.67,3.5])
saveas(gcf,'Flowrate_Plot.jpg')



%Load plots vs DeltaP-----------------------------------------------------
figure

clear xd yd zd
for s=1:6           %DeltaP Loop
    for t=1:5       %GVF Loop
        for u=1:3   %Suction Pressure Loop
            Power30(t+(u-1)*5,s)=Mean30_Load(t,u,s)*CalcPow30_Multiplier                +
CalcPow30_Constant;
            EPower30(t+(u-1)*5,s)=U30_Load(t,u,s)*CalcPow30_Multiplier;
            xd30(t+(u-1)*5,s)=Calc30_Delta_P(t,u,s);
            Power60(t+(u-1)*5,s)=Mean60_Load(t,u,s)*CalcPow60_Multiplier                +
```

```
CalcPow60_Constant;
            EPower60(t+(u-1)*5,s)=U60_Load(t,u,s)*CalcPow60_Multiplier;
            xd60(t+(u-1)*5,s)=Calc60_Delta_P(t,u,s);
        end
    end
end

[Power60_Coef_lin,Pred_Power60_lin,R_sq_Power60_lin]=regress_1var_1coef(Power60,xd60);
[Power30_Coef_lin,Pred_Power30_lin,R_sq_Power30_lin]=regress_1var_1coef(Power30,xd30);

hold on

plot(bogusx,bogusy,'kx')
plot(bogusx,bogusy,'ko')

errorbar(xd60,Power60,EPower60,'kx')
errorbar(xd30,Power30,EPower30,'ko')
plot([min(min(xd60)),max(max(xd60))],[min(min(Pred_Power60_lin)),max(max(Pred_Power60_lin
))],'color','k','linewidth',1)
plot([min(min(xd30)),max(max(xd30))],[min(min(Pred_Power30_lin)),max(max(Pred_Power30_lin
))],'color','k','linewidth',1)

ylim([30,85])
xlim([0,160])
xlabel('\DeltaP (psi)')
ylabel('Pump Load (hp)')
legend('60 Hz','30 Hz','Location','NorthWest')
l1=sprintf('%s       %4.2f       %s       %4.4f%s       \n              %s%4.4f','Load_6_0
=',Power60_Coef_lin(1),'+',Power60_Coef_lin(2),'(\DeltaP)','R^2 = ',R_sq_Power60_lin);
text(20,70,l1)
l2=sprintf('%s       %4.2f       %s       %4.4f%s       \n              %s%4.4f','Load_3_0
=',Power30_Coef_lin(1),'+',Power30_Coef_lin(2),'(\DeltaP)','R^2 = ',R_sq_Power30_lin);
text(70,40,l2)

hold off

set(gcf,'PaperPosition',[1.915,3.75,4.67,3.5])
saveas(gcf,'Load_Plot.jpg')

%Efficiency plots vs DeltaP-------------------------------------------------------
figure

clear xd yd zd
for s=1:6           %DeltaP Loop
    for t=1:5       %GVF Loop
        for u=1:3   %Suction Pressure Loop
            Efficiency60(t+(u-
1)*5,s)=100*Calc60_Vdot_Total(t,u,s)*Calc60_Delta_P(t,u,s)/(1714.3*(Mean60_Load(t,u,s)*Ca
lcPow60_Multiplier + CalcPow60_Constant));
            UEff60a(t+(u-
1)*5,s)=100*UC60_QdP(t,u,s)/((Mean60_Load(t,u,s)*CalcPow60_Multiplier                +
```

```
CalcPow60_Constant));
            UEff60b(t+(u-
1)*5,s)=U60_Load(t,u,s)*100*Calc60_Vdot_Total(t,u,s)*Calc60_Delta_P(t,u,s)*CalcPow60_Mult
iplier/(1714.3*(Mean60_Load(t,u,s)*CalcPow60_Multiplier + CalcPow60_Constant)^2);
            UEff60(t+(u-1)*5,s)=sqrt(UEff60a(t+(u-1)*5,s)^2+UEff60b(t+(u-1)*5,s)^2);
            xd60(t+(u-1)*5,s)=Calc60_Delta_P(t,u,s);
            Efficiency30(t+(u-
1)*5,s)=100*Calc30_Vdot_Total(t,u,s)*Calc30_Delta_P(t,u,s)/(1714.3*(Mean30_Load(t,u,s)*Ca
lcPow30_Multiplier + CalcPow30_Constant));
            UEff30a(t+(u-
1)*5,s)=100*UC30_QdP(t,u,s)/((Mean30_Load(t,u,s)*CalcPow30_Multiplier                    +
CalcPow30_Constant));
            UEff30b(t+(u-
1)*5,s)=U30_Load(t,u,s)*100*Calc30_Vdot_Total(t,u,s)*Calc30_Delta_P(t,u,s)*CalcPow30_Mult
iplier/(1714.3*(Mean30_Load(t,u,s)*CalcPow30_Multiplier + CalcPow30_Constant)^2);
            UEff30(t+(u-1)*5,s)=sqrt(UEff30a(t+(u-1)*5,s)^2+UEff30b(t+(u-1)*5,s)^2);
            xd30(t+(u-1)*5,s)=Calc30_Delta_P(t,u,s);
        end
    end
end

dp=[1:160];
Pred_Efficiency60=100*(VdotDeltaP60_Coef_lin(1)+VdotDeltaP60_Coef_lin(2)*dp)./(Power60_Co
ef_lin(1)+Power60_Coef_lin(2)*dp);
Pred_Efficiency30=100*(VdotDeltaP30_Coef_lin(1)+VdotDeltaP30_Coef_lin(2)*dp)./(Power30_Co
ef_lin(1)+Power30_Coef_lin(2)*dp);

hold on

plot(bogusx,bogusy,'kx')
plot(bogusx,bogusy,'ko')

errorbar(xd60,Efficiency60,UEff60,'kx','LineStyle','none')
errorbar(xd30,Efficiency30,UEff30,'ko','LineStyle','none')
plot(dp,Pred_Efficiency60,'color','k','linewidth',1)
plot(dp,Pred_Efficiency30,'color','k','linewidth',1)

ylim([0,70])
xlim([0,160])
xlabel('\DeltaP (psi)')
ylabel('Efficiency (%)')
legend('60 Hz','30 Hz','Location','NorthWest')


hold off

set(gcf,'PaperPosition',[1.915,3.75,4.67,3.5])
saveas(gcf,'Efficiency_Plot.jpg')
```

SINGLE SPEED PLOTS

These two scripts generate plots containing data from only one test speed.

```
clc
clear all
close all

load('ExtractedData30')
load('ExtractedDataPow30')


%Dimensionless axial pressures ------------------------------------------
clear xd yd
press1=zeros(5,5);
press2=zeros(5,5);
press3=zeros(5,5);
press4=zeros(5,5);
xd=zeros(5,5);
yd=zeros(5,5);
zd1=zeros(5,5);
zd2=zeros(5,5);
zd3=zeros(5,5);
zd4=zeros(5,5);


for s=2:6           %DeltaP Loop
    for t=1:5       %GVF Loop
        for u=1:3   %Suction Pressure Loop
            if s==2 && u==1 && t==1     %Drops a bad data point
                press(t,s-1)=press1(t,s-1);
            else
                press1(t,s-1)=press1(t,s-1)+(Mean30_AxPres(s,1,u,t)-
Mean30_P_Pump(t,u,s))/Calc30_Delta_P(t,u,s);
            end
            press2(t,s-1)=press2(t,s-1)+(Mean30_AxPres(s,2,u,t)-
Mean30_P_Pump(t,u,s))/Calc30_Delta_P(t,u,s);
            press3(t,s-1)=press3(t,s-1)+(Mean30_AxPres(s,3,u,t)-
Mean30_P_Pump(t,u,s))/Calc30_Delta_P(t,u,s);
            press4(t,s-1)=press4(t,s-1)+(Mean30_AxPres(s,4,u,t)-
Mean30_P_Pump(t,u,s))/Calc30_Delta_P(t,u,s);
            xd(t,s-1)=xd(t,s-1)+Calc30_Delta_P(t,u,s);
            yd(t,s-1)=yd(t,s-1)+Calc30_GVF(t,u,s);
        end
    end
end

press1(1,1)=1.5*press1(1,1); %Compensates for dropped data point in average
xd=xd./3;
press1=press1./3;
```

```
press2=press2./3;
press3=press3./3;
press4=press4./4;

% Plots for legend
a=[25,30];
b=[-1,-1];
figure
hold on
plot(a,b,'r')
plot(a,b,'g')
plot(a,b,'b')
plot(a,b,'m')
plot(a,b,'k-')
plot(a,b,'k--')
plot(a,b,'k:')

plot(xd(1,:),press1(1,:),'r-',xd(3,:),press1(3,:),'r--',xd(5,:),press1(5,:),'r:')
plot(xd(1,:),press2(1,:),'g-',xd(3,:),press2(3,:),'g--',xd(5,:),press2(5,:),'g:')
plot(xd(1,:),press3(1,:),'b-',xd(3,:),press3(3,:),'b--',xd(5,:),press3(5,:),'b:')
plot(xd(1,:),press4(1,:),'m-',xd(3,:),press4(3,:),'m--',xd(5,:),press4(5,:),'m:')

ylim([0,1])
xlim([25,160])
xlabel('\DeltaP (psi)')
ylabel('Normalized Internal Pressure')
legend('Axial Pressure 1','Axial Pressure 2','Axial Pressure 3', 'Axial Pressure 4', '90%
GVF','94% GVF','98% GVF')
hold off

saveas(gcf,'AxialPressure_Plot30.jpg')

%Temperature 4 by DeltaP at each Suction Pressure and GVF-----------------
figure
clear xd yd zd
for s=1:6           %DeltaP Loop
    for t=1:5       %GVF Loop
        for u=1:3   %Suction Pressure Loop
            t4(t+(u-1)*5,s)=Mean30_AxTemp(s,4,u,t);
            xd(t+(u-1)*5,s)=Calc30_Delta_P(t,u,s);
            zd(t+(u-1)*5,s)=Calc30_GVF(t,u,s);
            yd(t+(u-1)*5,s)=Mean30_P_Pump(t,u,s);
        end
    end
end

[Temp_Coef_lin,Pred_t4_lin,R_sq_Temp_lin]=regress_3var_3coef(t4,xd,yd,zd);

hold on

surf(xd(1:5,:),zd(1:5,:),yd(1:5,:),t4(1:5,:),'edgecolor',[0 0 0],'facecolor','interp')
```

```
surf(xd(6:10,:),zd(6:10,:),yd(6:10,:),t4(6:10,:),'edgecolor',[0                    0
0],'facecolor','interp')
surf(xd(11:15,:),zd(11:15,:),yd(11:15,:),t4(11:15,:),'edgecolor',[0                 0
0],'facecolor','interp')

caxis([80,140]);
grid on
colorbar
view([-40,22])
zlim([15,46])
ylim([.89,.98])
xlim([-5,160])

xlabel('\DeltaP (psi)')
ylabel('Gas Volume Fraction')
zlabel('Suction Pressure (psi)')
hold off

set(gcf,'PaperPosition',[1.915,3.75,4.67,3.5])
saveas(gcf,'MaxTemp_Plot30.jpg')


%Linear Multiple Regression Temperature Model------------------------------
figure
hold on

surf(xd(1:5,:),zd(1:5,:),yd(1:5,:),Pred_t4_lin(1:5,:),'edgecolor',[0                0
0],'facecolor','interp')
surf(xd(6:10,:),zd(6:10,:),yd(6:10,:),Pred_t4_lin(6:10,:),'edgecolor',[0            0
0],'facecolor','interp')
surf(xd(11:15,:),zd(11:15,:),yd(11:15,:),Pred_t4_lin(11:15,:),'edgecolor',[0        0
0],'facecolor','interp')

caxis([80,140]);
grid on
colorbar
view([-40,22])

zlim([15,46])
ylim([.89,.98])
xlim([-5,160])
xlabel('\DeltaP (psi)')
ylabel('Gas Volume Fraction')
zlabel('Suction Pressure (psi)')
hold off

set(gcf,'PaperPosition',[1.915,3.75,4.67,3.5])
saveas(gcf,'MaxTempLin_Plot30.jpg')
```

```
clc
clear all
close all

load('ExtractedData60')
load('ExtractedDataPow60')

%Dimensionless axial pressures ------------------------------------------
clear xd yd
press1=zeros(5,5);
press2=zeros(5,5);
press3=zeros(5,5);
press4=zeros(5,5);
xd=zeros(5,5);
yd=zeros(5,5);
zd1=zeros(5,5);
zd2=zeros(5,5);
zd3=zeros(5,5);
zd4=zeros(5,5);


for s=2:6           %DeltaP Loop
    for t=1:5       %GVF Loop
        for u=1:3   %Suction Pressure Loop
            press1(t,s-1)=press1(t,s-1)+(Mean60_AxPres(s,1,u,t)-
Mean60_P_Pump(t,u,s))/Calc60_Delta_P(t,u,s);
            press2(t,s-1)=press2(t,s-1)+(Mean60_AxPres(s,2,u,t)-
Mean60_P_Pump(t,u,s))/Calc60_Delta_P(t,u,s);
            press3(t,s-1)=press3(t,s-1)+(Mean60_AxPres(s,3,u,t)-
Mean60_P_Pump(t,u,s))/Calc60_Delta_P(t,u,s);
            press4(t,s-1)=press4(t,s-1)+(Mean60_AxPres(s,4,u,t)-
Mean60_P_Pump(t,u,s))/Calc60_Delta_P(t,u,s);
            xd(t,s-1)=xd(t,s-1)+Calc60_Delta_P(t,u,s);
        end
    end
end

% Average normalized pressures across P_suction
xd=xd./3;
press1=press1./3;
press2=press2./3;
press3=press3./3;
press4=press4./4;

a=[25,30];
b=[-1,-1];
figure
hold on
plot(a,b,'r')
plot(a,b,'g')
plot(a,b,'b')
```

```
plot(a,b,'m')
plot(a,b,'k-')
plot(a,b,'k--')
plot(a,b,'k:')

plot(xd(1,:),press1(1,:),'r-',xd(3,:),press1(3,:),'r--',xd(5,:),press1(5,:),'r:')
plot(xd(1,:),press2(1,:),'g-',xd(3,:),press2(3,:),'g--',xd(5,:),press2(5,:),'g:')
plot(xd(1,:),press3(1,:),'b-',xd(3,:),press3(3,:),'b--',xd(5,:),press3(5,:),'b:')
plot(xd(1,:),press4(1,:),'m-',xd(3,:),press4(3,:),'m--',xd(5,:),press4(5,:),'m:')


ylim([-.1,1])
xlim([25,160])
xlabel('\DeltaP (psi)')
ylabel('Normalized Internal Pressure')
legend('Axial Pressure 1','Axial Pressure 2','Axial Pressure 3', 'Axial Pressure 4', '90%
GVF','94% GVF','98% GVF')
hold off

saveas(gcf,'AxialPressure_Plot60.jpg')


%Temperature 4 by DeltaP at each Suction Pressure and GVF------------------
figure
clear xd yd zd
for s=1:6           %DeltaP Loop
    for t=1:5       %GVF Loop
        for u=1:3   %Suction Pressure Loop
            t4(t+(u-1)*5,s)=Mean60_AxTemp(s,4,u,t);
            xd(t+(u-1)*5,s)=Calc60_Delta_P(t,u,s);
            zd(t+(u-1)*5,s)=Calc60_GVF(t,u,s);
            yd(t+(u-1)*5,s)=Mean60_P_Pump(t,u,s);
        end
    end
end

[Temp_Coef_lin,Pred_t4_lin,R_sq_Temp_lin]=regress_3var_3coef(t4,xd,yd,zd);

hold on

surf(xd(1:5,:),zd(1:5,:),yd(1:5,:),t4(1:5,:),'edgecolor',[0 0 0],'facecolor','interp')
surf(xd(6:10,:),zd(6:10,:),yd(6:10,:),t4(6:10,:),'edgecolor',[0                          0
0],'facecolor','interp')
surf(xd(11:15,:),zd(11:15,:),yd(11:15,:),t4(11:15,:),'edgecolor',[0                       0
0],'facecolor','interp')

caxis([80,140]);
grid on
colorbar
view([-40,22])
zlim([15,46])
```

```
ylim([.89,.98])
xlim([-5,160])

xlabel('\DeltaP (psi)')
ylabel('Gas Volume Fraction')
zlabel('Suction Pressure (psi)')

hold off

set(gcf,'PaperPosition',[1.915,3.75,4.67,3.5])
saveas(gcf,'MaxTemp_Plot60.jpg')

%Linear Multiple Regression Temperature Model------------------------------
figure
hold on

surf(xd(1:5,:),zd(1:5,:),yd(1:5,:),Pred_t4_lin(1:5,:),'edgecolor',[0                    0
0],'facecolor','interp')
surf(xd(6:10,:),zd(6:10,:),yd(6:10,:),Pred_t4_lin(6:10,:),'edgecolor',[0                0
0],'facecolor','interp')
surf(xd(11:15,:),zd(11:15,:),yd(11:15,:),Pred_t4_lin(11:15,:),'edgecolor',[0            0
0],'facecolor','interp')

caxis([80,140]);
grid on
colorbar
view([-40,22])

zlim([15,46])
ylim([.89,.98])
xlim([-5,160])
xlabel('\DeltaP (psi)')
ylabel('Gas Volume Fraction')
zlabel('Suction Pressure (psi)')

hold off

set(gcf,'PaperPosition',[1.915,3.75,4.67,3.5])
saveas(gcf,'MaxTempLin_Plot60.jpg')
```

LINEAR REGRESSION FUNCTIONS

These two functions perform the single and multiple linear regression, respectively, on the supplied data variables. The multiple regression function approximates one variable as a linear combination of three others.

```
function [A,My_reg,R_sq]=regress_1var_1coef(My,Mx1)
%this function performs a linear regression on My with Mx1,
%uses the equation
%y=a0+a1*x1

s=1;
[l,m,n]=size(My);
for i=1:l
    for j=1:m
        for k=1:n
            y(s)=My(i,j,k);
            x1(s)=Mx1(i,j,k);

            s=s+1;
        end
    end
end

M=[ones(length(y),1),x1'];
y=y';
A=M\y;
yhat=M*A;
ybar=mean(y);
SSE=(y-yhat)'*(y-yhat);
SST=(y-ybar)'*(y-ybar);
R_sq=1-SSE/SST;
Deviation=(SSE/(length(y)-length(A)))^.5;

My_reg=A(1)*ones(l,m,n)+A(2)*Mx1;
return



function [A,My_reg,R_sq]=regress_3var_3coef(My,Mx1,Mx2,Mx3)
%this function performs a multiple regression on My with Mx1, Mx2, and Mx3
%uses the equation
%y=a0+a1*x1+a2*x2+a3*x3

s=1;
[l,m,n]=size(My);
for i=1:l
    for j=1:m
        for k=1:n
```

```
            y(s)=My(i,j,k);
            x1(s)=Mx1(i,j,k);
            x2(s)=Mx2(i,j,k);
            x3(s)=Mx3(i,j,k);
            s=s+1;
        end
    end
end

M=[ones(length(y),1),x1',x2',x3'];
y=y';
A=M\y;
yhat=M*A;
ybar=mean(y);
SSE=(y-yhat)'*(y-yhat);
SST=(y-ybar)'*(y-ybar);
R_sq=1-SSE/SST;
Deviation=(SSE/(length(y)-length(A)))^.5;

My_reg=A(1)*ones(l,m,n)+A(2)*Mx1+A(3)*Mx2+A(4)*Mx3;
return
```

# VITA


| | |
|---|---|
| Name: | Michael W. Glier |
| Address: | Texas A&M University<br>Department of Mechanical Engineering<br>3123 TAMU<br>College Station TX 77843-3123 |
| Email Address: | glierm@tamu.edu |
| Education: | B.S. Mechanical Engineering, Texas A&M University, 2009<br>M.S. Mechanical Engineering, Texas A&M University, 2011 |