# EFFECT OF DEGRADED OIL ON TERRY TURBINE BEARING PERFORMANCE AT HIGH TEMPERATURE

A Thesis

by

GAGE BRANSON GREEN

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

MASTER OF SCIENCE

| | |
|---|---|
| Committee Chair, | Karen Vierow Kirkland |
| Committee Members, | Mark Kimber |
| | Adolfo Delgado |
| Head of Department, | Michael Nastasi |

May 2020

Major Subject: Nuclear Engineering

# ABSTRACT

In 2011, a series of seismic events induced a station blackout condition at the Japanese nuclear power site Fukushima Daiichi. Without power, all three units were unable to maintain proper core cooling. To supplement the inadequate cooling, the RCIC system began operation in Units 2 and 3. RCIC is estimated to have a battery life of 4 to 12 hours, though, during the Fukushima accident, was able to stay in operation for nearly three days, but then suddenly failed. The theory behind this behavior is that the Terry turbine (a major component of the RCIC system) was able to enter a “self-regulating” mode after the batteries died, but then suffered a bearing failure. This thesis explores how oil temperature and quality affect the journal bearings of a ZS-1 Terry turbine. Tests were conducted to evaluate the change in the composition of oil as it thermally degrades over the course of 72 hours, and to investigate the effect thermally degraded oil had on the performance of the turbine. Ultimately, the change in oil composition over the course of a 72-hour 121$^{o}$C degradation was found to be negligibly small. Additionally, when the same degraded oil was used as bearing lubricant in a ZS-1 Terry turbine, the effect on torque was noticeable, but minor. Scoring to the bearings did occur when degraded oil was used, but the degree of damage was superficial.

# ACKNOWLEDGEMENTS

Foremost, I offer my deepest gratitude to Dr. Matt Solom, whose expertise, assistance, and guidance greatly enhanced the quality of this research. I would also like to recognize the work Diana Peck as a precursor to this thesis, and thank her for facilitating a smooth transition between researchers, and for her continued cooperation. Additionally, Ben Lear deserves acknowledgement for his work in helping to set up the new DAQ system, and constructing the bearing facility's LabVIEW program; none of the bearing tests could have been performed without it. Lastly, I would like to express my appreciation towards the members of the NHTS Lab; their presence was beneficial to this research, and enriching to the spirit.

# CONTRIBUTORS AND FUNDING SOURCES

## Contributors

This work was supported by a thesis committee consisting of Dr. Karen Kirkland and Dr. Mark Kimber of the Department of Nuclear Engineering, and Dr. Adolfo Delgado of the Department of Mechanical Engineering.

## Funding Sources

Funding for this study was provided by the Institute of Applied Energy. Additionally, equipment and labor were donated by Revak Keene Turbomachinery Services.

# NOMENCLATURE

| | |
|---|---|
| BWR | Boiling Water Reactor |
| DAQ | Data Acquisition Center |
| NHTS | Nuclear Heat Transfer Systems |
| RCIC | Reactor Core Isolation Cooling |
| TTEXOB | Terry Turbine Expanded Operating Band |
| VFD | Variable Frequency Drive |

# TABLE OF CONTENTS

Page

ABSTRACT .......... ii

ACKNOWLEDGEMENTS .......... iii

CONTRIBUTORS AND FUNDING SOURCES .......... iv

NOMENCLATURE .......... v

TABLE OF CONTENTS .......... vi

LIST OF FIGURES .......... x

LIST OF TABLES .......... xiii

1. INTRODUCTION .......... 1

1.1. Terry Turbines in Nuclear Power .......... 1
1.2. Project Motivation .......... 3
1.3. Importance of Work .......... 3

2. LITERATURE SURVEY .......... 5

2.1. Terry Turbines .......... 5
2.2. Journal Bearings .......... 6
2.3. Turbine Oil .......... 7

3. FACILITY .......... 8

3.1. Oil Facility .......... 8
3.1.1. Equipment .......... 8
3.1.1.1 Beakers .......... 8
3.1.1.2 Data Acquisition Center .......... 9
3.1.1.3 Fume Hood .......... 10

3.1.1.4 Hot Plates ... 11
3.1.2. Instrumentation ... 12
3.1.2.1 Thermocouples ... 12
3.1.2.2 Viscometer ... 13
3.1.3. Layout ... 14
3.1.4. Improvements ... 15
3.2. Bearing Facility ... 15
3.2.1. Equipment ... 15
3.2.1.1 Data Acquisition Center ... 15
3.2.1.2 Heaters ... 16
3.2.1.3 Motor ... 17
3.2.1.4 Safety Coupling ... 17
3.2.1.5 Terry Turbine ... 18
3.2.1.6 Variable Frequency Drive ... 19
3.2.2. Instrumentation ... 20
3.2.2.1 Accelerometers ... 20
3.2.2.2 Barometer and Hygrometer ... 22
3.2.2.3 Tachometer ... 24
3.2.2.4 Thermocouples ... 25
3.2.2.5 Torque Meter ... 25
3.2.3. Layout ... 26
3.2.4. Improvements ... 30
4. TESTING ... 35
4.1. Oil Degradation Tests ... 35
4.1.1. Procedure ... 35
4.1.2. Test Matrix ... 36
4.2. Bearing Tests ... 37
4.2.1. Procedure ... 37
4.2.2. Test Matrix ... 38
5. RESULTS ... 41
5.1. Oil Degradation Tests ... 41
5.2. Bearing Tests ... 53

5.3. Uncertainty Analysis ... 63
6. CONCLUSION ... 67
6.1. Summary ... 67
6.2. Future Work ... 68
REFERENCES ... 69
APPENDIX A: OIL DEGRADATION PLOTS ... 71
A.1. 121ºC Oil Degradation - 2/18/2019 ... 71
A.2. 93ºC 10% Wetness Oil Degradation - 3/7/2019 ... 75
A.3. 121ºC Oil Degradation - 7/8/2019 ... 81
APPENDIX B: BEARING TEST PLOTS ... 85
B.1. Shakedown ... 85
B.1.1. Multiple Speed Tests - 2/22/2019 ... 85
B.1.2. 6 Hour Shakedown - 4/3/2019 ... 87
B.1.3. Heater Tape Shakedown - 7/26/2019 ... 89
B.1.4. Immersion Heater Calibration - 10/21/2019 ... 93
B.1.5. Immersion Heater Shakedown - 10/24/2019 ... 97
B.2. Tests ... 101
B.2.1. Bearing Stress Tests ... 101
B.2.1.1. Fresh Oil, No Heat, 1000 rpm - 11/8/2019 ... 101
B.2.1.2. Fresh Oil, No Heat, 3000 rpm - 11/12/2019 ... 108
B.2.1.3. Fresh Oil, 121ºC, 1000 rpm - 11/13/2019 ... 115
B.2.1.4. Fresh Oil, 121ºC, 3000 rpm - 12/03/2019 ... 122
B.2.1.5. Degraded Oil, No Heat, 1000 rpm - 12/16/2019 ... 129
B.2.1.6. Degraded Oil, No Heat, 3000 rpm - 01/06/2020 ... 136
B.2.1.7. Degraded Oil, 121ºC, 1000 rpm - 01/14/2020 ... 144
B.2.1.8. Degraded Oil, 121ºC, 3000 rpm - 01/15/2020 ... 151
B.2.2. Steady State Torque ... 158
B.2.2.1. Fresh Oil, No Heat - 10/29/2019 ... 158
B.2.2.2. Fresh Oil, 121ºC - 11/26/2019 ... 160
B.2.2.3. Degraded Oil, No Heat - 01/07/2020 ... 162

B.2.2.4. Degraded Oil, 121$^{o}$C - 01/15/2020 .......... 164

APPENDIX C: OPERATING PROCEDURES .......... 166

C.1. Oil Degradation Tests .......... 166

C.2. Heated Bearing Tests .......... 167

C.2.1. Bearing Inspection .......... 167

C.2.2. Bearing Test .......... 169

APPENDIX D: PYTHON SCRIPTS .......... 171

D.1. Oil Data Reader .......... 171

D.2. Bearing Data Reader (LabVIEW 2015) .......... 178

D.3. Bearing Data Reader (3 Thermocouples) .......... 179

D.4. Bearing Data Reader (4 Thermocouples, Misconfiguration) .......... 201

D.5. Bearing Data Reader (4 Thermocouples) .......... 223

D.6. Torque vs Speed Regression Generator (LabVIEW 2015) .......... 245

D.7. Torque vs Speed Regression Generator .......... 250

# LIST OF FIGURES

Page

Figure 1.1: RCIC system layout and flow paths for a Mark I BWR .................................. 2

Figure 3.1: Shown here are the two types of beaker used during the oil degradations ........................................................................................... 8

Figure 3.2: Oil facility LabVIEW program user interface .................................................. 9

Figure 3.3: Front view of the fume hood ........................................................................ 10

Figure 3.4: Hot plate used for degradation tests .......................................................... 11

Figure 3.5: Shown here are examples of the thermocouples used during testing ............................................................................................................ 12

Figure 3.6: Viscometer used for oil degradation tests ................................................... 13

Figure 3.7: Front view of the oil degradation experimental setup .................................. 14

Figure 3.8: Bearing facility LabVIEW program user interface ........................................ 16

Figure 3.9: Safety coupling used to pair the turbine to the torque meter ....................... 18

Figure 3.10: Initial configuration of the ZS-1 Terry Turbine ........................................... 19

Figure 3.11: Front panel of the VFD .............................................................................. 20

Figure 3.12: Shown here are all four mounted accelerometers ..................................... 22

Figure 3.13: Front display of the barometer .................................................................. 23

Figure 3.14: Front display and measurement probe of the hygrometer ......................... 23

Figure 3.15: Tachometer (left) and its display box (right) ............................................ 24

Figure 3.16: Shown here is the torque meter ............................................................ 26

Figure 3.17: Externally visible key components of the bearing facility, indicated by color.............................................................................................. 27

Figure 3.18: Mounting locations of the four accelerometers, and the 3-dimensional reference frame ....................................................................... 28

Figure 3.19: Front view (top left), side view (top right), and top view (bottom) of the thermocouple and heater arrangement in the oil wells ........................................................................................................ 29

Figure 3.20: VFD before (left) and after (right) the cable extension ............................. 31

Figure 3.21: Steel plate used to seal the governor penetration .................................... 32

Figure 3.22: Turbine configuration at the start of the research..................................... 33

Figure 3.23: Turbine configuration at the end of this research ..................................... 34

Figure 5.1: Average oil temperature over time for the initial 121$^{o}$C oil degradation.......................................................................................... 42

Figure 5.2: This plot shows the relationship between viscosity and temperature before and after the degradation, for the initial 121$^{o}$C oil degradation ........................................................................................ 43

Figure 5.3: Shown here is a comparison between oil prior to (left) and after (right) the initial 121$^{o}$C oil degradation............................................................ 44

Figure 5.4: Temperature over time for the 93$^{o}$C 10% wet oil degradation.................... 45

Figure 5.5: This plot shows the relationship between viscosity and temperature before and after the degradation, for the 93$^{o}$C 10% wet oil degradation ........................................................................................ 46

Figure 5.6: Progression of the oil/water mixture’s appearance over the course of the test. The timestamps in order of numbering are 0:51:00, 1:08:00, 2:19:00, 4:06:00, and 6:18:00 ................................................ 47

Figure 5.7: Shown here is the oil/water mixture the day after the test (left), and one week later (right) ...................................................................... 48

Figure 5.8: Average oil temperature over time for the second 121$^{o}$C oil degradation........................................................................................................ 49

Figure 5.9: This plot shows the relationship between viscosity and temperature before and after the degradation, for the second 121$^{o}$C oil degradation........................................................................................ 50

Figure 5.10: Comparison between the two sets of viscosity vs temperature curves, for the two 72-hour tests ............................................................ 51

Figure 5.11: Comparison between the two sets of viscosity vs temperature curves, for the initial dry test and the 10% wet test.......................................... 52

Figure 5.12: Rotational shaft speed over time for the unheated, fresh oil, 3000 rpm test.............................................................................................. 54

Figure 5.13: Motor torque over time for the four 1000 rpm bearing stress tests.................................................................................................................. 55

Figure 5.14: Motor torque over time for the four 3000 rpm bearing stress tests.................................................................................................................. 56

Figure 5.15: This plot shows the relationship between motor torque and oil temperature for the heated, degraded oil, 3000 rpm test................................... 57

Figure 5.16: This plot shows the relationship between motor torque and oil temperature for the heated, degraded oil, 1000 rpm test................................... 57

Figure 5.17: ZS-1 journal bearings; on the left is the governor side set, and on the right is the motor side set......................................................................... 58

Figure 5.18: Shown here is the damage to the bottom bearings (left - governor side, right -motor side) following the unheated, degraded oil, 3000 rpm test..................................................................................... 59

Figure 5.19: Comparison of all of the steady state torque tests .................................... 60

Figure 5.20: Comparison of the fresh oil steady state torque tests .............................. 61

Figure 5.21: Shown here is a comparison of the degraded oil steady state torque tests......................................................................................................... 61

Figure 5.22: Shown here is a comparison of the unheated steady state torque tests......................................................................................................... 62

Figure 5.23: Comparison of the heated steady state torque tests................................ 62

# LIST OF TABLES

Page

Table 4.1: TTEXOB P9roject Proposed Oil Testing Matrix .......................................... 36

Table 4.2: TTEXOB Project Proposed Heated Bearing Testing Matrix ......................... 38

Table 4.3: Testing Matrix for Bearing Stress Tests ...................................................... 39

Table 4.4: Testing Matrix for Torque vs Speed Tests.................................................... 39

# 1. INTRODUCTION

## 1.1. Terry Turbines in Nuclear Power

Initial models of the Terry turbine were designed in the early 1900s.[1] The basic concept behind the function of the Terry turbine is similar to that of most turbines. A working fluid (usually steam) is channeled through the turbine, and used to spin a wheel; this in turn generates mechanical power in the shaft. The shaft is supported and lubricated through the use of journal bearings. Journal bearings are less efficient, but more durable than the more commonly known ball bearing. The shaft rotates inside the bearing, and “floats” on the lubricating film (oil) at the interface of the bearing inner bore and outside shaft surfaces. The two series of Terry turbine being considered in this project are the ZS and GS. The Z model has an 18 in wheel diameter and has brass bearings, whereas the G has a 24 in wheel diameter and Babbitt bearings.

In the nuclear field, the primary function of turbines is to generate electricity. However, the Terry turbine serves a different purpose. The Terry turbine is a major component of the Reactor Core Isolation Cooling (RCIC) system. The RCIC system is a self-regulated, emergency cooling system used in Boiling Water Reactors (BWRs). RCIC operates to deliver coolant to the core when the other feed water systems fail. Additionally, RCIC can operate under station black out conditions because it is battery powered. Figure 1.1, on the next page, illustrates the layout of the RCIC system.

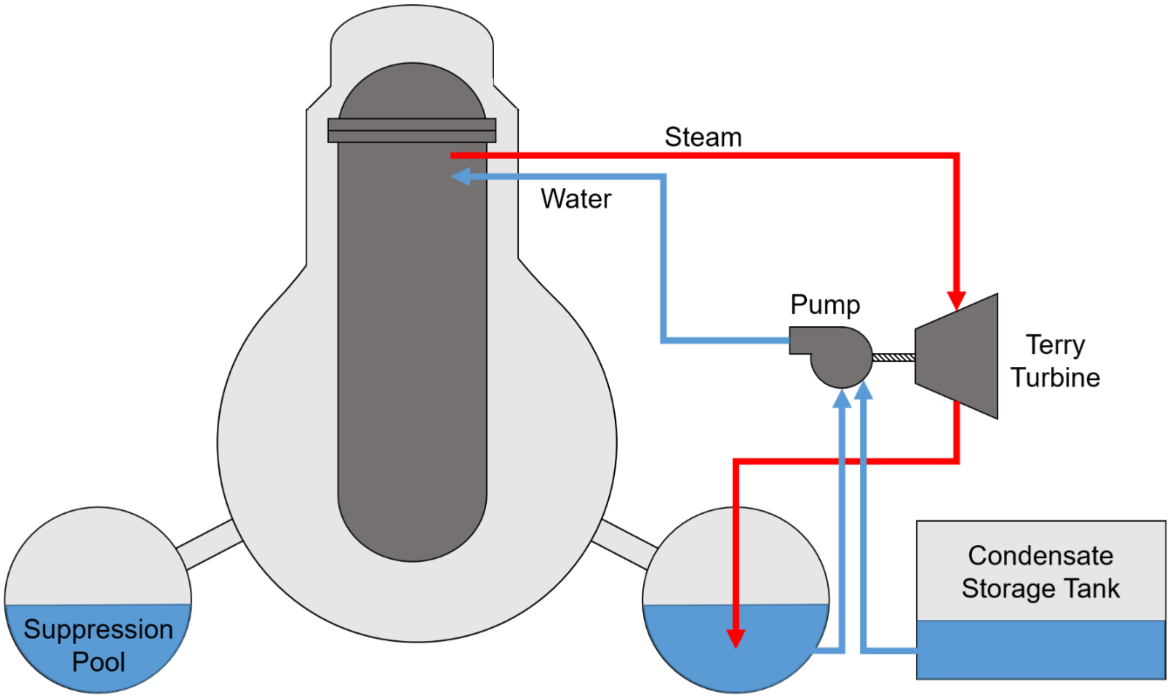


**Figure 1.1:** RCIC system layout and flow paths for a Mark I BWR.

Steam is channeled from the core through the Terry turbine to power a pump; the steam is then injected into the suppression pool, where it condenses into water. The pump is used to feed water from the condensate storage tank into the core, thereby replenishing the coolant inventory. Additionally, if the coolant inventory of the condensate storage tank is depleted, the pump can also pull from the suppression pool.[3]

### 1.2. Project Motivation

During the events of the 2011 Fukushima accident, a tsunami simultaneously disabled on site power and wiped out 12 of the 13 diesel generators available to the six onsite reactors. Due to the near station blackout scenario, the three units at Fukushima Daiichi lost the ability to maintain proper core cooling.[4] To supplement the inadequate cooling, RCIC began operation in Units 2 and 3. The power requirement for RCIC comes from the system’s valves and turbine controls (i.e. turning the system on and off and controlling the turbine speed). For this reason, the RCIC system is estimated to have an operation time equal to its battery life of 4 to 12 hours; once valve controls are lost, RCIC is expected to either trip on overspeed, or flood its own steam lines and subsequently fail.[5] However, during the Fukushima accident, RCIC remained in operation for nearly three days. Not only did RCIC remain in operation well beyond its battery life, but the Terry turbine was also able to function, despite the flooded environment. One leading theory behind the RCIC’s shutdown is that the lubricating oil in the Terry turbine lost adequate cooling, which resulted in a thermal failure of the bearings. It is these details that warrant an investigation into the robustness and limits of the Terry turbine, and its ability to operate under beyond design basis conditions.

### 1.3. Importance of Work

The significance of this research is multi-dimensional, as it helps to answer questions regarding both past events, and future operating standards. First of all, much of what occurred during the Fukushima accident regarding the performance of the RCIC system is unknown, and (due to the nature of nuclear accidents) will remain unknown for many years; the experiments conducted here will help to provide answers in a much more timely fashion. Secondly, this research serves to put data into writing. Outside of the efforts of the Terry Turbine Expanded Operating Band (TTEXOB) project, very little documentation exists regarding the Terry turbine. Even less information exists regarding the thermal and impurity (i.e. water, air, debris) limits of the Terry turbine’s journal bearings. This is largely the result of the “age” of the technology. Stemming from the early 20th century, much of what is known about Terry turbines has either been passed down by word of mouth, or can be found in illegible, scan-of-a-scan documents. Because of the resources and regulations of the time, little is known about the aforementioned limits. Finally, this project will provide additional safety credit to Terry turbines being used in the industry, and could potentially indicate overly conservative assumptions regarding the durability of the system; this in turn could lead to the expansion of the current Terry turbine operating band.

# 2. LITERATURE SURVEY

## 2.1. Terry Turbines

As previously mentioned, the basic concept behind the function of Terry turbines is similar to that of conventional turbines, but there are some key differences. Terry turbines are impulse turbines with a Pelton wheel type. For this reason, they aren't very efficient, but are extremely robust. Much of the existing in-depth literature regarding Terry turbines stems from the efforts of this project, and in most cases strictly pertains to the Fukushima accident. The remaining documents are as previously described: old (60 plus years) and generally low quality. The history of Terry turbines is often referred to as an "oral" one. The precursor to this thesis, "Turbine Oil Degradation and its Effects on Performance of a Terry Turbine in Nuclear Applications" by Diana Peck, focused on facility design for future testing and provided a small set of preliminary data.[6] Various other TTEXOB reports propose possible reasons as to why the RCIC system (and by extension, the GS-2 Terry turbine) operated beyond it's expected battery life, and then suddenly failed. In regards to the extended operation of the RCIC system, the leading theory is that the Terry turbine was able to achieve self-regulation, without the assistance of automated steam throttling. Unthrottled high quality steam from the reactor core allowed the turbine to rapidly pump coolant into the core. The rapid influx of coolant into the core decreased the quality of the steam being fed into the turbine, slowing the rate at which it pumped water. This in turn allowed the coolant boiloff rate to overcome the injection rate, thus increasing the steam quality again.[2, 5] As for the sudden failure, it is assumed that the Terry turbine suffered a catastrophic journal bearing failure resulting from any combination of the following factors: high temperatures in the bearings and oil, degraded oil, aerated oil, wet oil, bearing corrosion, and particulates in the bearings.

## 2.2. Journal Bearings

The function of a bearing is to both support a load, and to reduce friction between moving parts. Bearings can (for the most part) be separated into two main groups: rolling element and fluid film.[7] The journal bearings of the Terry turbine fall under the latter category. In this situation, the inner surface of the bearings is completely separated from the surface of the shaft by an oil film. While the shaft is spinning, it essentially “floats” on the film, therefore being supported and having its frictional losses greatly reduced.

The effectiveness of journal bearings is heavily dependent upon the configuration of the oil film. As long as there is a thick, consistent film between the shaft and the bearings, the bearing lifetime will essentially be infinite. However, if the oil film should become too thin or irregular, the bearings could suffer damage and fail. The thickness of the oil film is dependent upon the oil viscosity, rotational shaft speed, applied load, and clearance between the shaft and the bearings.[8] As the oil heats up, its viscosity will decrease, and so too will the film. If the shaft does not spin fast enough, the film will become irregular and could break down altogether. Lastly, the clearance between the shaft and the bearings can be reduced (therefore reducing the thickness of the film) by overloading and underloading the bearings, catching particulates between the shaft and the bearing, and through varnish buildup in the inside surface of the journal bearing. Additionally, the introduction of varnish to the bearings can act as an insulator, allowing heat to build up inside the bearings. This results in a viscosity drop in the oil, and can cause the oil to degrade. Oil degradation leads to more varnish buildup, which further exacerbates the issue. Any of these scenarios can lead to direct rubbing of the bearings, and some cases can even cause the shaft to “bounce” inside the bearing. Regardless of the reason, these behaviors will negatively affect the performance of the turbine and degrade the bearing over time, possibly causing them to fail.

### 2.3. Turbine Oil

The type of lubricating oil used in this research was Mobil DTE-732, which is a conventional, light turbine oil. Referencing the official Safety Data Sheet, this oil has a flash point of 215$^{o}$C, boiling point of 316$^{o}$C, pour point of -18$^{o}$C, and a viscosity of 31.5 cSt at 40$^{o}$C and 5.68 cSt at 100$^{o}$C. This oil was specifically chosen because it is what’s used in the industry. The primary motivation that the industry has for using a conventional oil over synthetic is that, simply put, it works and reactor sites would need to be relicensed if they wanted to change.

# 3. FACILITY

## 3.1. Oil Facility

### *3.1.1. Equipment*

3.1.1.1 Beakers

Beakers were used as a container for the oil while it was being heated during the oil degradation tests. For the initial 121$^{o}$C test, as well as the 93$^{o}$C 10% wet test, two 3000 mL glass Veegee Griffin style beakers (model no. 20229-3000-PK) were used. For the second 121$^{o}$C test, the change was made to two stainless steel Cole Parmer Griffin style beakers (model no. EW-07205-80) for purposes of better heat transfer. Both of these beaker styles are shown in Figure 3.1, below.

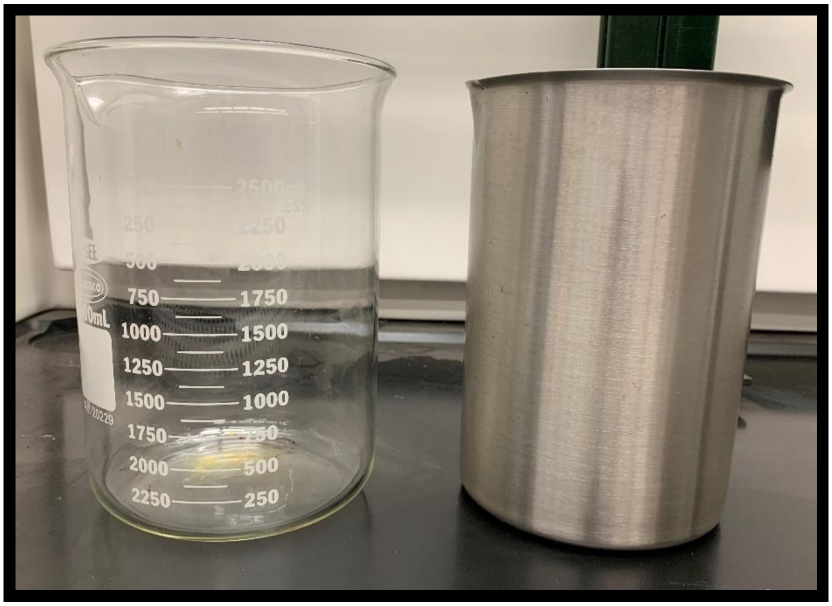
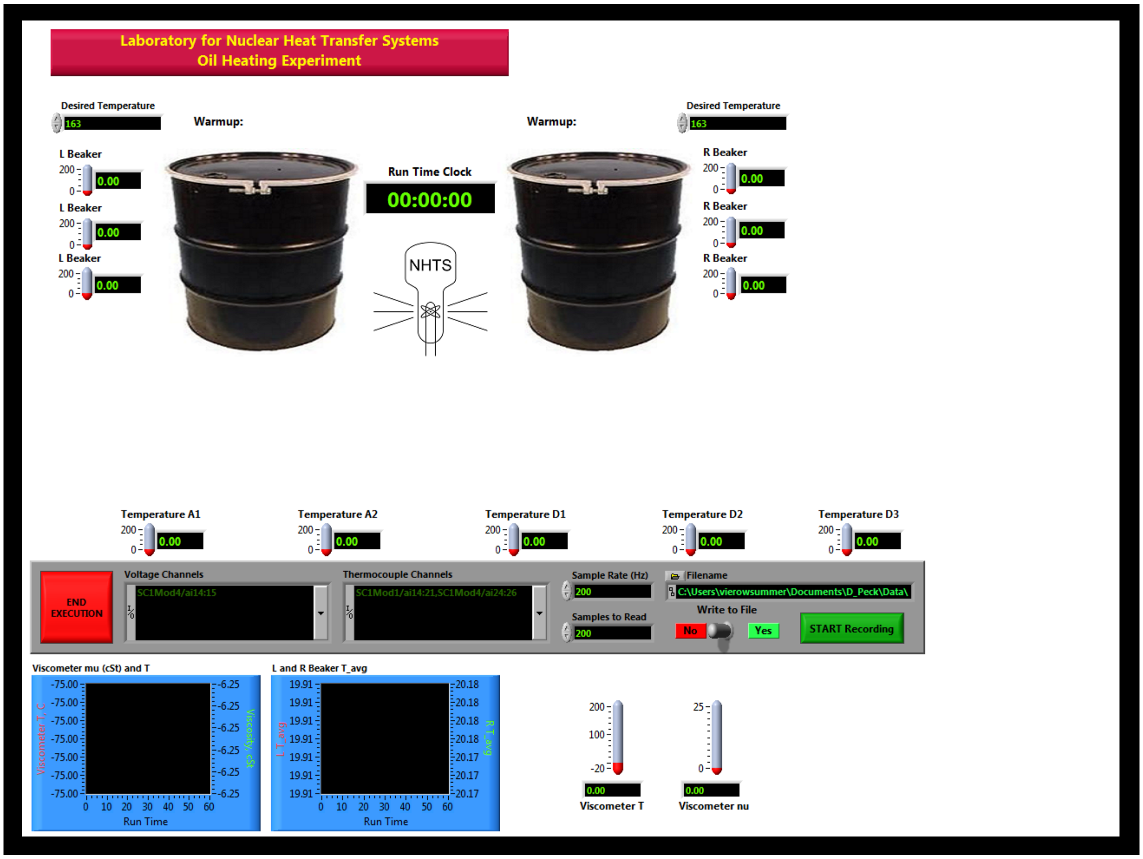


**Figure 3.1:** Shown here are the two types of beaker used during the oil degradations.

### 3.1.1.2 Data Acquisition Center

The data acquisition center (DAQ) comprises the researcher/instrumentation interface. Here, all of the data was assembled and displayed using a LabVIEW 2015 program.[6] The hardware for this system consisted of a SCXI-1000 chassis, four modules (two SCXI-1102, SCXI-1102B, and SCXI-1102C), and a PCIe-6341 data acquisition card; all of these are National Instruments products. Figure 3.2, below, shows the graphical user interface for the DAQ.



Laboratory for Nuclear Heat Transfer Systems
Oil Heating Experiment

Desired Temperature: 163
Desired Temperature: 163
Run Time Clock: 00:00:00
L Beaker: 0.00
L Beaker: 0.00
L Beaker: 0.00
R Beaker: 0.00
R Beaker: 0.00
R Beaker: 0.00
Temperature A1: 0.00
Temperature A2: 0.00
Temperature D1: 0.00
Temperature D2: 0.00
Temperature D3: 0.00
Voltage Channels: SC1Mod4/ai14:15
Thermocouple Channels: SC1Mod1/ai14:21,SC1Mod4/ai24:26
Sample Rate (Hz): 200
Samples to Read: 200
Filename: C:\Users\vierowsummer\Documents\D_Peck\Data\
Viscometer T: 0.00
Viscometer nu: 0.00

**Figure 3.2:** Oil facility LabVIEW program user interface.

### 3.1.1.3 Fume Hood

The fume hood used in these experiments is a part of the Labcono Protector Premier Laboratory Hood series. This piece of equipment allowed for the safe execution of oil degradation tests. The fume hood is equipped with an external blower, fluorescent lighting, two independent pairs of outlets, an adjustable blast resistant sash, and cabinet space for the storage of equipment and chemicals. Figure 3.3, below, provides a front view of the fume hood.

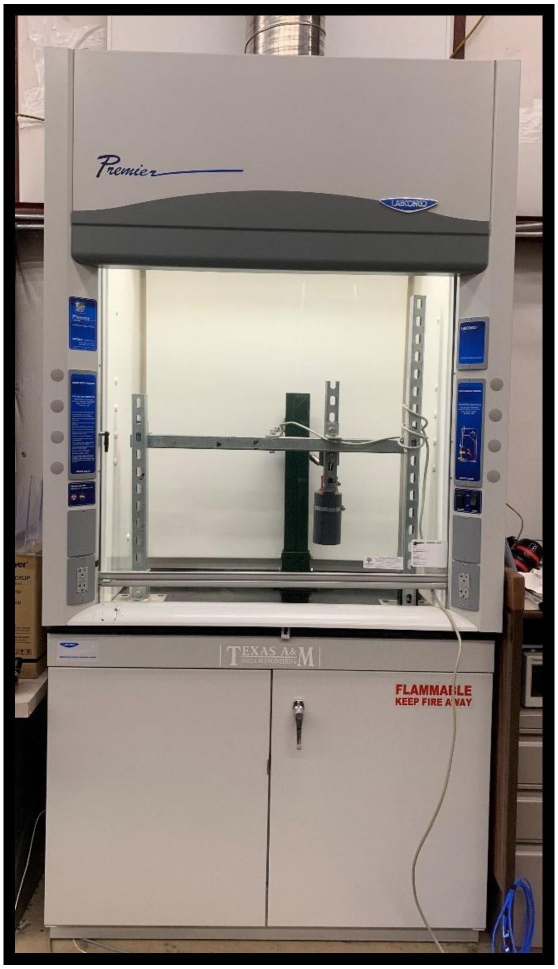


**Figure 3.3:** Front view of the fume hood.

### 3.1.1.4 Hot Plates

The type of hot plate used for the oil tests was the Thermo Scientific Type 2200 Aluminum Top Hot Plate (model no. HPA2235MQ). This hot plate has a 12x12 in aluminum heating surface, with a maximum load of 25 lbs, and a temperature range of 30$^{o}$C – 371$^{o}$C. This hot plate maintains a desired surface temperature by means of an analog control knob and an internal magnetic contact plate. When the surface temperature falls below the desired temperature, the magnetic plate snaps onto the knob’s contact plate, completing a circuit and inducing an “on” state. When the surface temperature exceeds the desired temperature, the magnetic plate detaches, and cuts the circuit, resulting in an “off” state. Unfortunately, at medium temperatures over a prolonged period of time (>100$^{o}$C, >12 hours), the magnetic contact has a tendency to arc weld itself to the knob’s contact plate, permanently leaving it in an “on” state. To resolve this, the knob has to be turned all the way off (sometimes for a couple of minutes) to build up tension and separate the weld. Both hot plates suffered from this defect, so they had to be monitored very closely, and one was ultimately replaced. Figure 3.4, below, shows one of the hot plates.

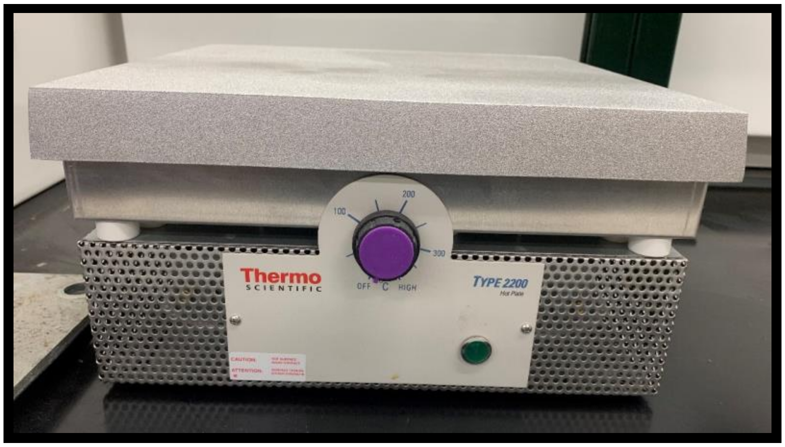


**Figure 3.4:** Hot plate used for oil degradation tests.

### *3.1.2. Instrumentation*

#### 3.1.2.1 Thermocouples

The thermocouples used during testing were the Omega Quick Disconnect Thermocouples. These are type T, ungrounded, 316 stainless steel sheathed, with a 0.062 in diameter; in most cases, the probes were 18 in (model no. TMQSS-062U-18). These thermocouples have a temperature range of -250°C – 400°C, and an uncertainty of ±0.5°C (or ±0.4%, if greater). The reason behind using the type T thermocouples is that they are the cheapest for the temperature range required, and have the smallest amount of error. Figure 3.5, below, shows an example of one of these thermocouples.

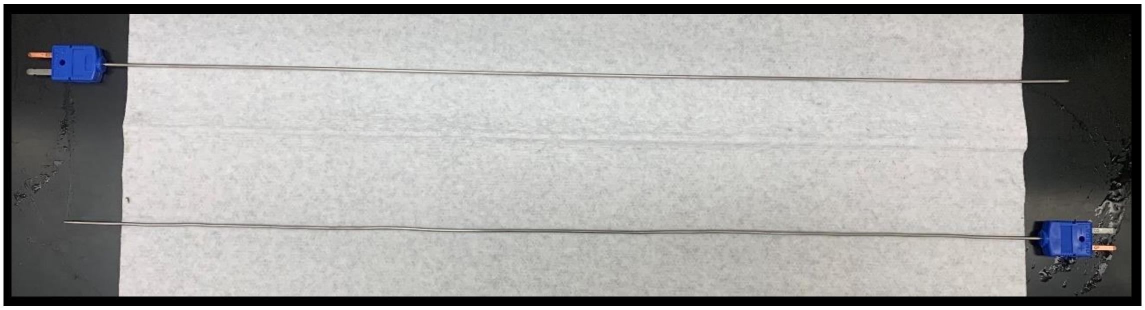


**Figure 3.5:** Shown here are examples of the thermocouples used during testing.

#### 3.1.2.2 Viscometer

The viscometer used during the oil degradations was the Brookfield Model AST-100IP Viscometer. This is a vibratory style viscometer that generates a digital signal proportional to the measured viscosity. This instrument measures both temperature and viscosity, and has a thermal range of -18°C – 200°C with an associated error of ±4%. Figure 3.6, below, shows the viscometer.

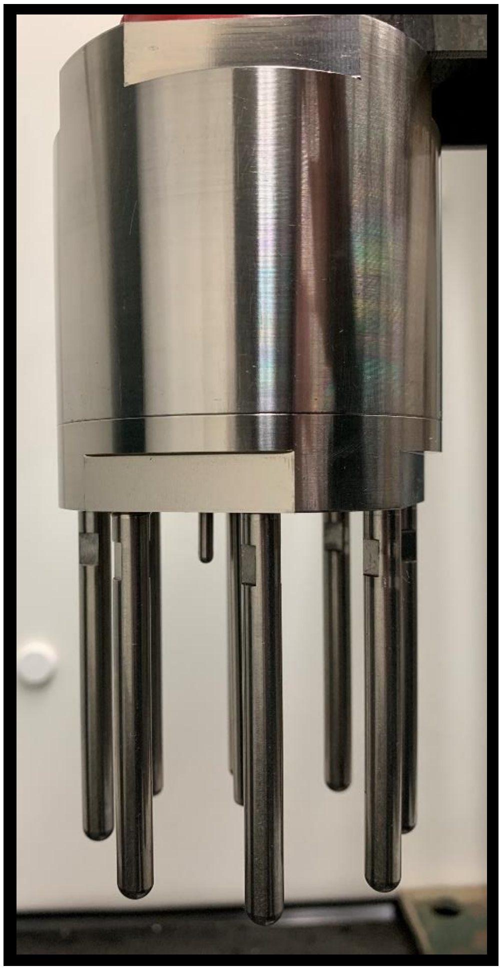


**Figure 3.6:** Viscometer used for oil degradation tests.

### *3.1.3. Layout*

The layout for these tests was quite simple. Inside the fume hood, two beakers of oil each sat on their own hot plate. The hot plates were both plugged into independent outlets, and both beakers were wrapped in two layers of fiberglass insulation. Each beaker had three thermocouples in them. The thermocouples were all submerged in the oil equidistant from one another, at low, medium, and shallow depths. The viscometer was always placed in the right beaker. Additionally, it was also submerged deep enough in the oil that the probes were completely below the oil surface, but not so much that the steel body of the viscometer introduced unnecessary thermal inertia. The fume hood shroud maintained a 3 in opening (this level was found to have the least amount of heat loss), the light was kept on, and the blower was left running. Figure 3.7, below, shows a typical experimental setup, this one specifically from the second 121ºC oil degradation (7/08/2019).

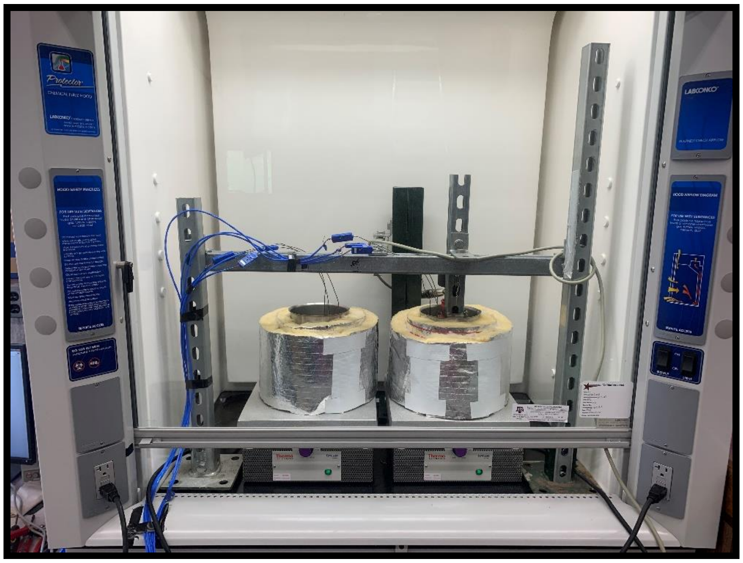


**Figure 3.7:** Front view of the oil degradation experimental setup.

### *3.1.4. Improvements*

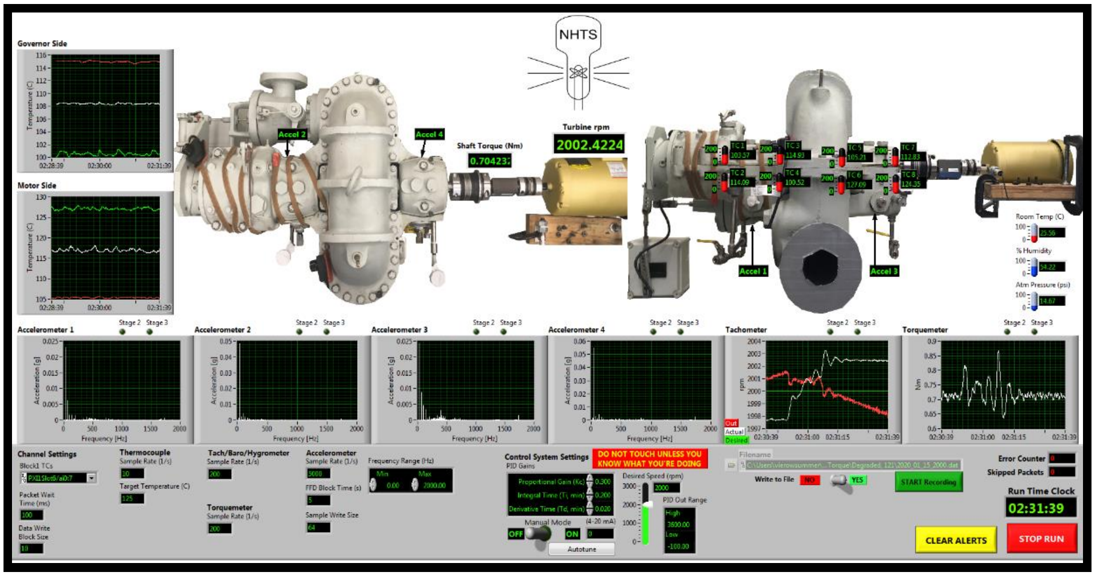
As the oil degradation process itself is simple, few changes needed to be made from the original setup. Facility wise, the previously discussed viscometer was a new addition to the tests; the degradations performed by Peck only collected temperature data. With the incorporation of the viscometer came the channel strut support system, seen in Figure 3.7. This structure suspended the viscometer in the oil of the right beaker, and provided a clean and convenient means to route cables and thermocouples. Lastly, glass beakers were swapped out for stainless steel beakers. This accelerated the heat up process, and provided more uniform heating to the oil. Procedurally, these oil degradations were 72 hours at temperature, which resulted in 100+ hour tests; in the past, these tests were limited to 72 hours.

## **3.2. Bearing Facility**

### *3.2.1. Equipment*

#### 3.2.1.1 Data Acquisition Center

As was the case with the oil facility, the data acquisition center comprises the researcher/instrumentation interface. Here, all of the data was assembled and displayed using a LabVIEW 2018 program. The hardware for this system consisted of a PXIe-1075 chassis, five modules (TB-4302, TB-4302C, TB-4322, TB-4353, and PXIe-4464), and the same PCIe-6341 data acquisition card as was used in the oil tests; all of these are National Instruments products. Figure 3.8, on the next page, shows the graphical user interface of the DAQ for the bearing facility.



Shaft Torque (Nm) 0.704232

Turbine rpm 2002.4224

TC 1 103.57
TC 2 114.09
TC 3 114.93
TC 4 100.52
TC 5 105.21
TC 6 127.09
TC 7 112.83
TC 8 124.35

Room Temp (C) 25.56
% Humidity 54.22
Atm Pressure (psi) 14.67

Channel Settings
Block1 TCs PXI1Slot6/ai0:7
Packet Wait Time (ms) 100
Data Write Block Size 10

Thermocouple
Sample Rate (1/s) 10
Target Temperature (C) 125

Tach/Baro/Hygrometer
Sample Rate (1/s) 200

Torquemeter
Sample Rate (1/s) 200

Accelerometer
Sample Rate (1/s) 5000
FFD Block Time (s) 5
Sample Write Size 64

Frequency Range (Hz)
Min 0.00
Max 2000.00

Control System Settings
PID Gains
Proportional Gain (Kc) 0.300
Integral Time (Ti, min) 0.200
Derivative Time (Td, min) 0.020
Manual Mode OFF ON
(4-20 mA) 0

Desired Speed (rpm) 2000
PID Out Range
High 3600.00
Low -100.00

Filename C:\Users\vierowsummer\...Torque\Degraded_12\2020_01_15_2000.dat

Write to File NO YES

Error Counter 0
Skipped Packets 0

Run Time Clock 02:31:39

**Figure 3.8:** Bearing facility LabVIEW program user interface.

### 3.2.1.2 Heaters

The type of heater used in the heated bearing tests is a custom Watlow ¾ in Firerod Cartridge Heaters (model no. 2080-2071). This heater is 5 in in length, 120 V, 800 W, and is equipped with an on-sheath internal thermocouple (type K, incaloy sheath). In total, two of these heaters were used (one for each oil well).

Additionally, the heaters are controlled by a custom Valin control console (model no. VTS-110-2-1-120-K). This is a 120 V, two-zone controller with type K thermocouple connections. This console can be controlled directly, or wirelessly via smartphone app (Watlow EZ-Link).

### 3.2.1.3 Motor

The motor used to spin the turbine in the bearing tests was a Baldor Reliance Super E Motor (model no. EM3555T). This is a 2 hp, 230 V, 3 phase motor, with a ⅞ in shaft diameter. This motor has a peak torque of 19 Nm, a continuous operating torque of 4 Nm, and a synchronous speed of 3600 rpm.

### 3.2.1.4 Safety Coupling

The safety coupling used in the bearing tests was a custom R+W SK2 Torque Limiter. This coupling was located between the turbine and the torque meter, and has a disengagement torque of 19.25 Nm. The purpose of the safety coupling was to protect the torque meter and motor from harmful torque levels, as well as provide a safe failure mode. In the event that the set torque is exceeded, the safety coupling will disengage from the torque meter, preventing any damage. For added security, the shaft is covered by a rotating part guard. Though no disengagements occurred, events that could have caused one include shaft seizure due to bearing failure, and torque overload during VFD tuning. Figure 3.9, on the next page, shows the safety coupling.

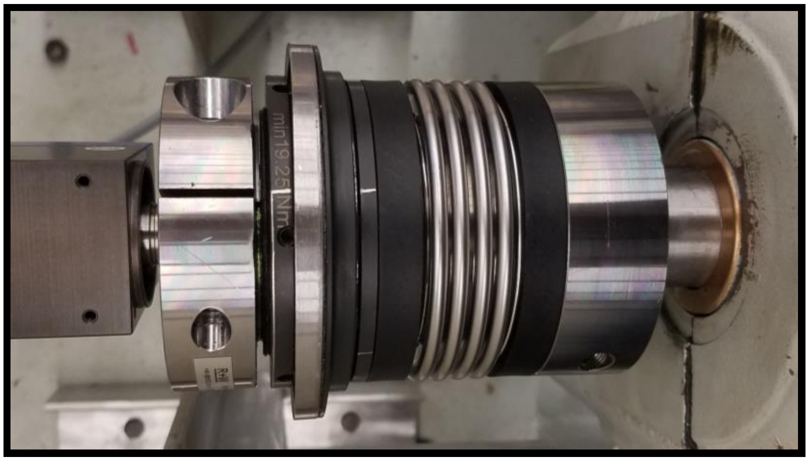


**Figure 3.9:** Safety coupling used to pair the turbine to the torque meter.

### 3.2.1.5 Terry Turbine

Of course, the most critical piece of equipment for the bearing tests was the turbine; specifically, a ZS-1 Terry turbine. The turbine was purchased refurbished from Revak-Keene.[6] It came equipped with two sets of bearings, oil bubblers, and a governor valve (though, the governor valve was not used for any tests). The turbine itself was secured in place by bolting and welding it to a steel sled. The bearing material is 660 bronze (a high leaded tin bronze). There are four unique pieces that pair together to form two complete journal bearings, one for each oil well. The purpose of the oil bubblers was to maintain and provide a visual indication of the oil level. Figure 3.10, on the next page, shows the Terry turbine prior to any modifications.

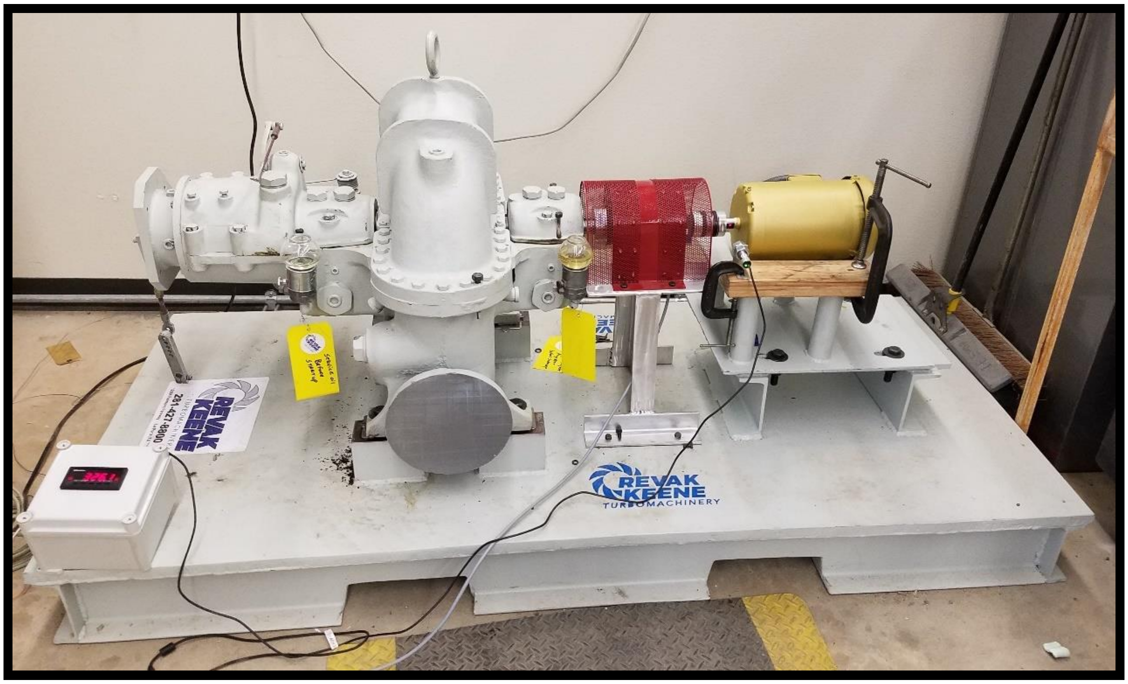


**Figure 3.10:** Initial configuration of the ZS-1 Terry turbine.

3.2.1.6 Variable Frequency Drive

The variable frequency drive (VFD) used for the bearing tests was the Galt G300 Series VFD (model no. G320-00070UL-01). This VFD pairs with a 2 hp, 230 V, 3 phase motor. The function of the VFD is to deliver power to (and sometimes absorb from) the motor, as well as control it. The VFD can be operated directly via control panel, though for these tests it was managed automatically by a PID (proportional integral derivative) control loop in the LabVIEW program. Figure 3.11, on the next page, shows the VFD.

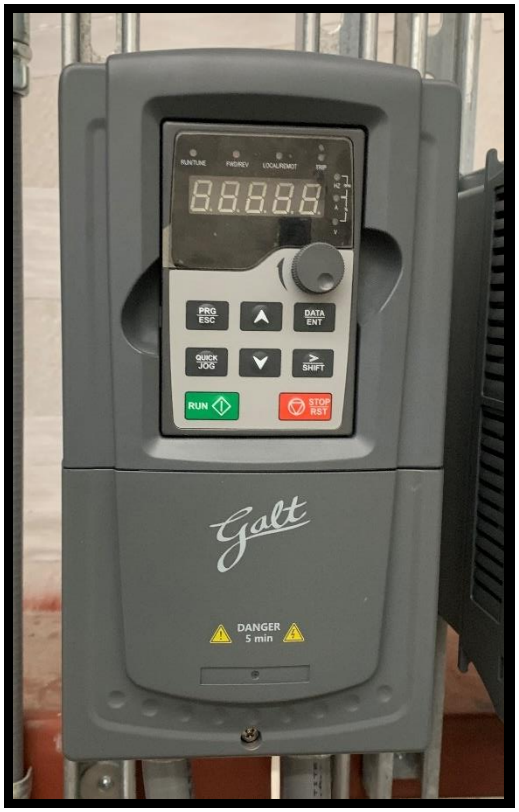


**Figure 3.11:** Front panel of the VFD.

### *3.2.2. Instrumentation*

#### 3.2.2.1 Accelerometers

There were four accelerometers used in the bearing tests; these were PCB Piezotronics High Temperature ICP Accelerometers (model no. 320C33). These accelerometers have a measurement range of ±50 g pk, a frequency range of 1 Hz – 4000 Hz, a sensitivity of 100 mV/g (±10%), and a maximum amplitude (g) uncertainty of ±2% (for the observed frequency range). The purpose of the accelerometers was to monitor the vibrational profile of the turbine while testing, in the event that an oil film breakdown occurred or some other shaft/bearing interaction. Ultimately, no such event took place. Proximity probes were considered for this application, but were deemed inappropriate due to the expected high temperatures. The accelerometers themselves measure vibration in the direction normal to the surface they're mounted to. Additionally, they measure vibrational force per user defined frequency bins. Despite having a range of 1 Hz – 4000 Hz, a max frequency of 2000 Hz was set, with a bin size of 31.25 Hz for a total of 64 bins per accelerometer. The max frequency was lowered because only low-level frequencies (<1000 Hz) were expected to be seen. Additionally, the 64 frequency bins per accelerometer was a compromise between spectrum resolution and output file size. The accelerometers were mounted directly to the turbine body, centered as close as possible on the bearings. This was done by sanding down the mounting surface (for maximum adhesion and to eliminate the possibility of an angled mount), attaching the mount to the sanded surface via high temperature epoxy, and screwing the accelerometer into the mount. Figure 3.12, on the next page, shows all four mounted accelerometers; the location of the mounts relative to the turbine will be discussed later, in Section 3.2.3.

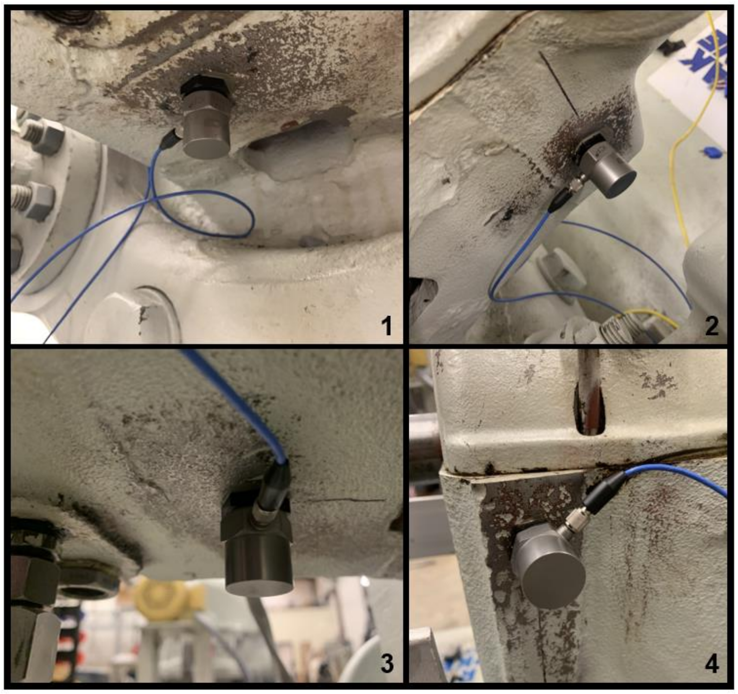


**Figure 3.12:** Shown here are all four mounted accelerometers.

#### 3.2.2.2 Barometer and Hygrometer

The purpose of the barometer and hygrometer was to monitor the ambient conditions of the lab during testing; this is a good resource in the event that seasonal/weather conditions have an effect on the collected data. The barometer used was a Rosemount 3051 Coplanar Pressure Transmitter (model no. 3051CA1A22A1AM4Q4HR7), which has a max working pressure of 207 kPa. The hygrometer used was a Vaisala Humidity and Temperature Transmitter (model no. HMT334). This instrument measures both ambient humidity and temperature; it has a temperature range of -70$^{o}$C – 180$^{o}$C, and an operating pressure range of 0 MPa – 10 MPa. Figure 3.13 and 3.14, on the next page, show the barometer and hygrometer, respectively.

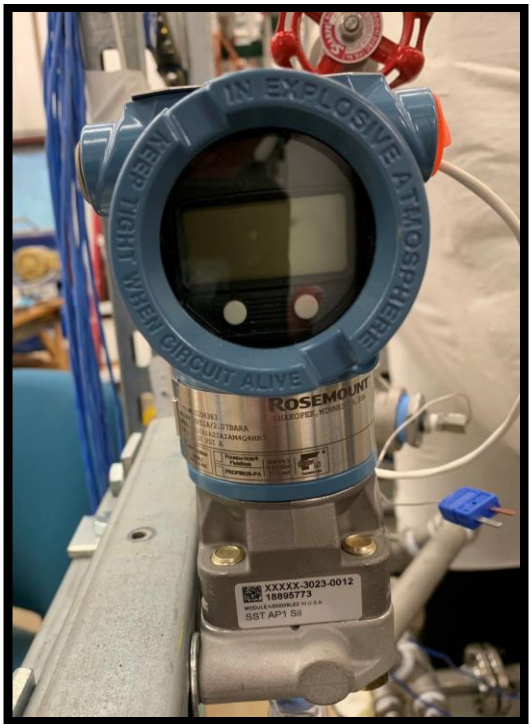


**Figure 3.13:** Front display of the barometer.

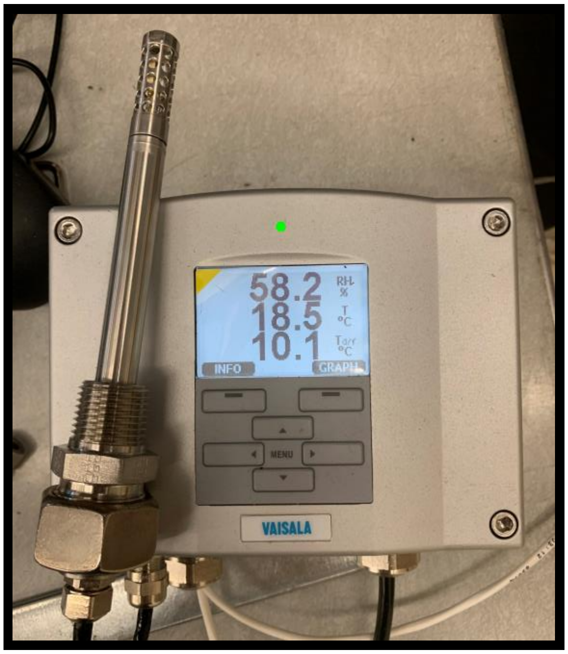


**Figure 3.14:** Front display and measurement probe of the hygrometer.

### 3.2.2.3 Tachometer

The tachometer used in the bearing tests was the Monarch Remote Optical Laser Sensor-W (model no. 6180-030). The tachometer measures shaft speed, and has an operating range of 1 rpm – 250,000 rpm; it also has an associated error of ±0.01%. At low speeds (< 100 rpm) the tachometer had a tendency to double or triple count rotations, as can be seen in the large peaks at the beginning and end of the speed vs time plots in Section 5 and Appendix B. This behavior was due to the fact that the motor shaft is keyed, and the "key slot" created a small crease in the reflective tape; at low speeds the tachometer could see the crease and would interpret it as a separation in the reflective tape. Figure 3.15, below, shows the tachometer and its display box.

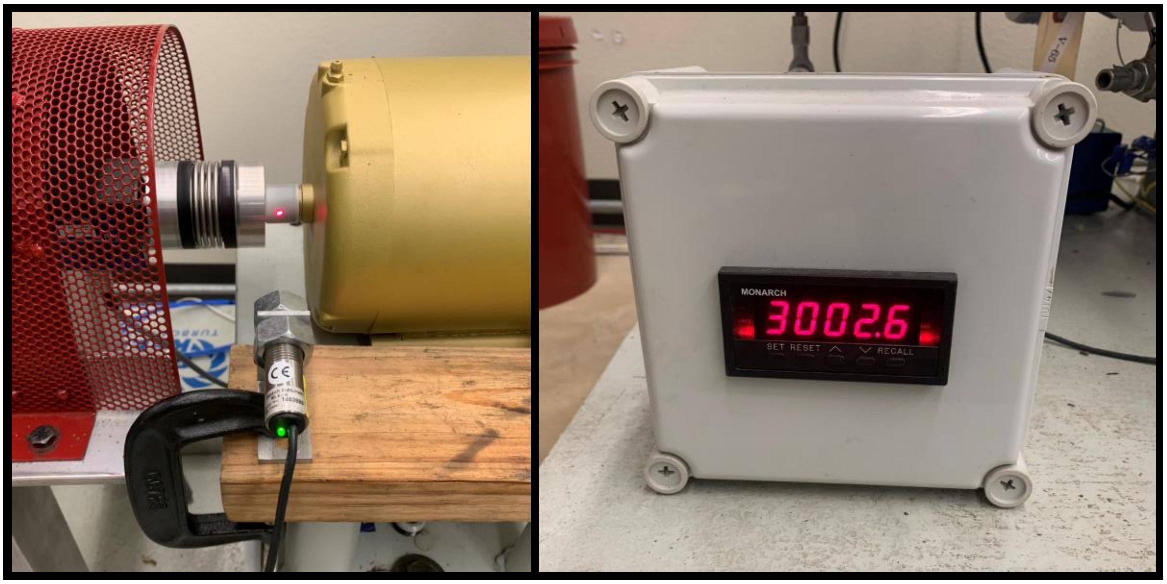


**Figure 3.15:** Tachometer (left) and its display box (right).

#### 3.2.2.4 Thermocouples

In addition to the same type T thermocouples used in the oil facility, the bearing tests also required the use of two type K thermocouples. The style of K type thermocouple employed was the McMaster-Carr Thermocouple Probe for Liquids and Gases (model no. 39095K96); these probes were grounded, 316 stainless steel sheathed, with a diameter of .062 in, temperature range of -198$^{o}$C – 921$^{o}$C, and uncertainty of ±0.75%. The two K type thermocouples were used to collect a reference temperature in each oil well for the heater control console; they were not used for data collection. The type T thermocouples were used for tracking the temperature of the oil over the course of a test. Each oil well had four thermocouples uniformly distributed in them. The exact configuration of the thermocouples will be discussed later, in Section 3.2.3.

#### 3.2.2.5 Torque Meter

The torque meter used in the bearing tests was the HBM Torque Transducer (model no. T21WN/20NM). This torque meter has a measurable torque limit of 20 Nm, maximum load of 40 Nm, max measurable rotational speed of 20,000 rpm (i.e. torque measurements cannot be made above this speed), shaft size of 16 mm, and an uncertainty of ±0.2%. Figure 3.16, below, shows the torque meter.

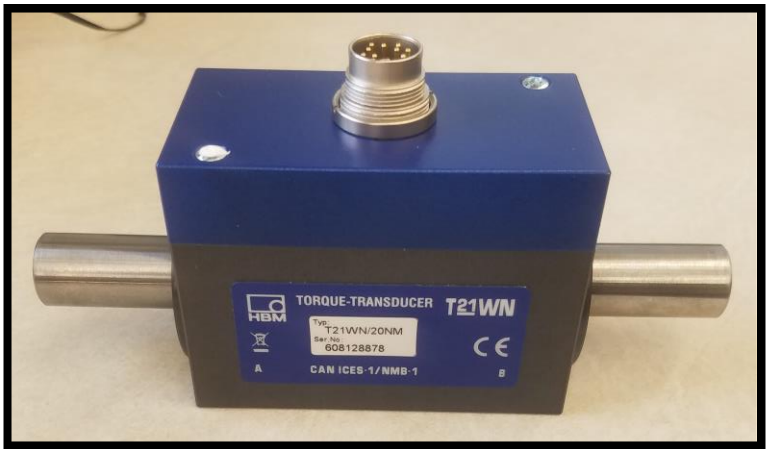


**Figure 3.16:** Shown here is the torque meter.

### *3.2.3. Layout*

The layout of the bearing facility is not complicated, but it is multidimensional. To avoid confusion, this description will first highlight all of the externally visible components, and then move on to the internal/less visible instruments.

Figure 3.17, on the next page, highlights the externally visible key components of the bearing facility. From right to left, the motor (black) is seen coupled to the torque meter (blue) via the safety coupling (white). The tachometer (red) measures shaft speed by observing the rotation of the motor's shaft. The bubbler's (green) passively control and maintain the oil level in the oil wells, as well as provide a visual indication of the oil levels. Lastly, the heaters (yellow) screw into the turbine body and protrude halfway into the oil wells, roughly an inch from the bottom.

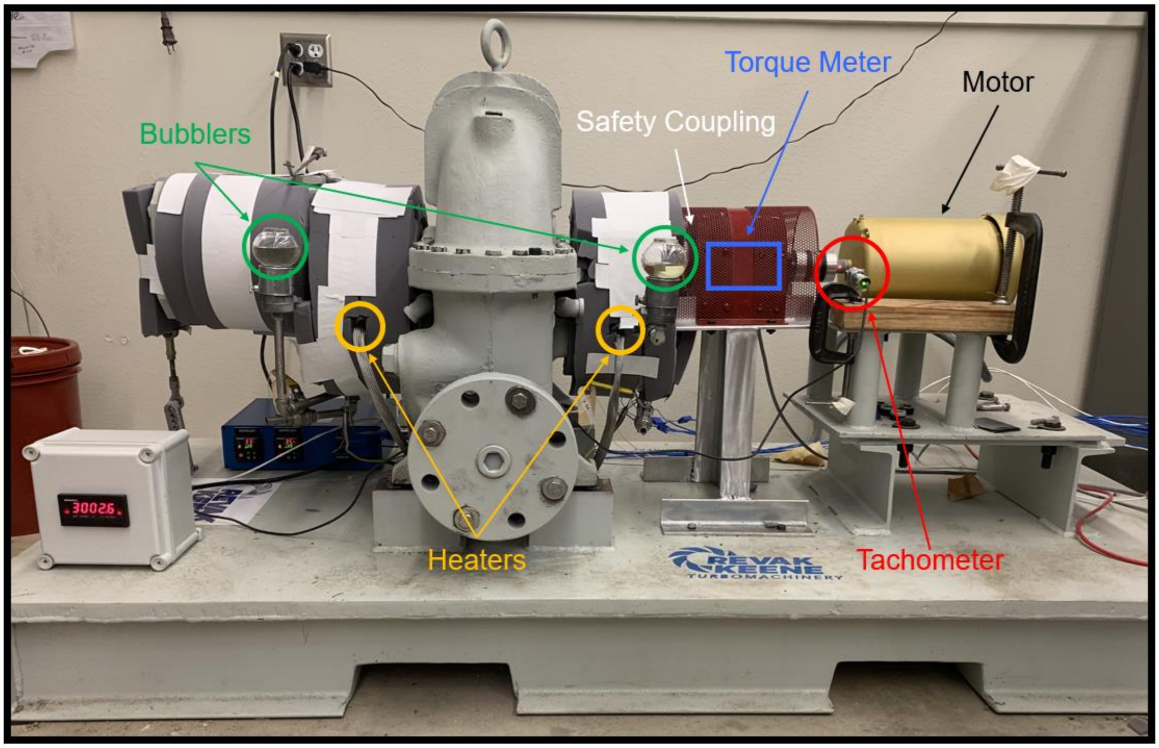


**Figure 3.17:** Externally visible key components of the bearing facility, indicated by color.

In Figure 3.18, on the next page, the mounting locations of the four accelerometers are marked in red. Accelerometer 1 (AC1) and accelerometer 3 (AC3) are mounted on the bottom of the turbine, and measure vertical vibration in the y-axis. Accelerometer 2 (AC2) and accelerometer 4 (AC4) are mounted on the back of the turbine and measure horizontal vibration in the x-axis. These locations are also shown on the turbine graphic in Figure 3.8. As previously mentioned, the probe mounts were placed as close to the center of the bearings as possible, but, due to the curvature of the turbine body, the “x” probes had to be placed slightly off center. It was important to identify the accelerometers from one another, because each one had a unique calibration constant. Specifically, AC1, AC2, AC3, and AC4 correspond to SN 19993, SN 20128, SN 20127, and SN 19994, respectively.

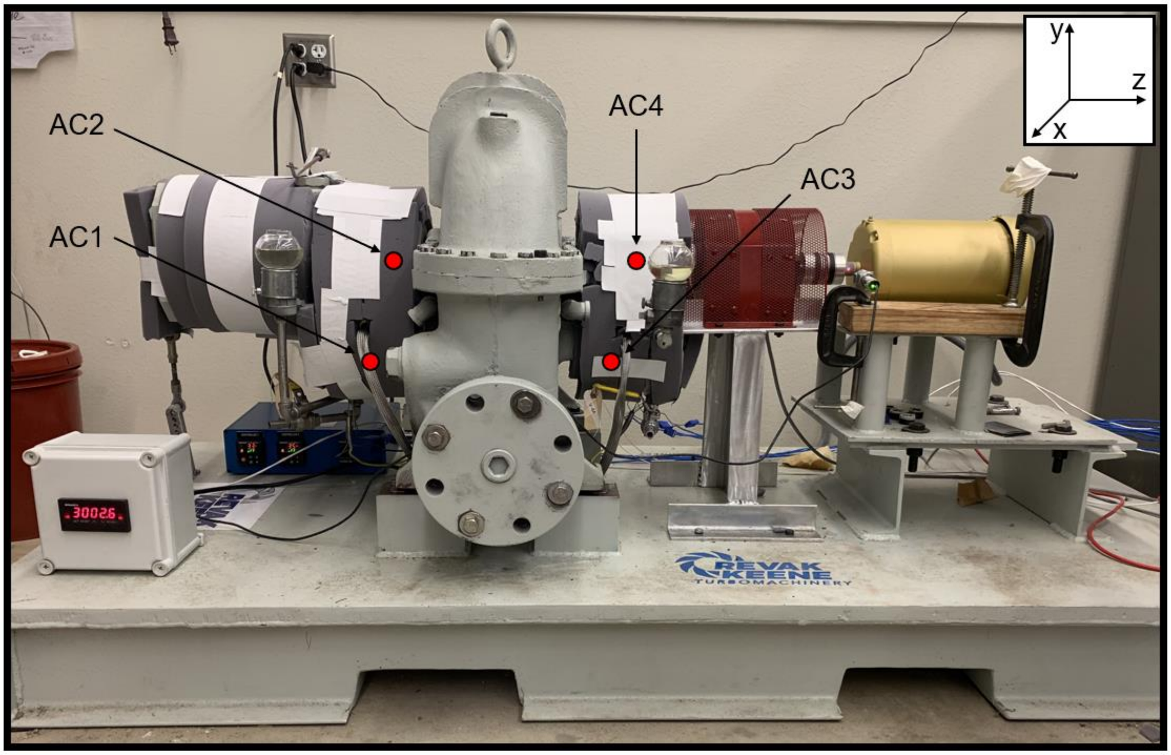


**Figure 3.18:** Mounting locations of the four accelerometers, and the 3-dimensional reference frame.

Figure 3.19, on the next page, gives a front, side and top view of the thermocouple and heater placement in the oil wells. The black thermocouples represent the type T thermocouples used for data collection; the brown thermocouple represents the type K thermocouple used as a reference temperature for the heater control console. To get an accurate average oil temperature reading, the type T thermocouples were uniformly distributed in the oil well; each was in a different corner, and alternated between a “high” or “low” elevation. When the heaters were on, there was significant thermal stratification in the oil. Therefore, the reference probe was placed near the heater, above its radial centerline (heated shakedowns had the refence probe placed about 0.5 in lower, which resulted in a reading roughly 30$^{o}$C below the average oil temperature).

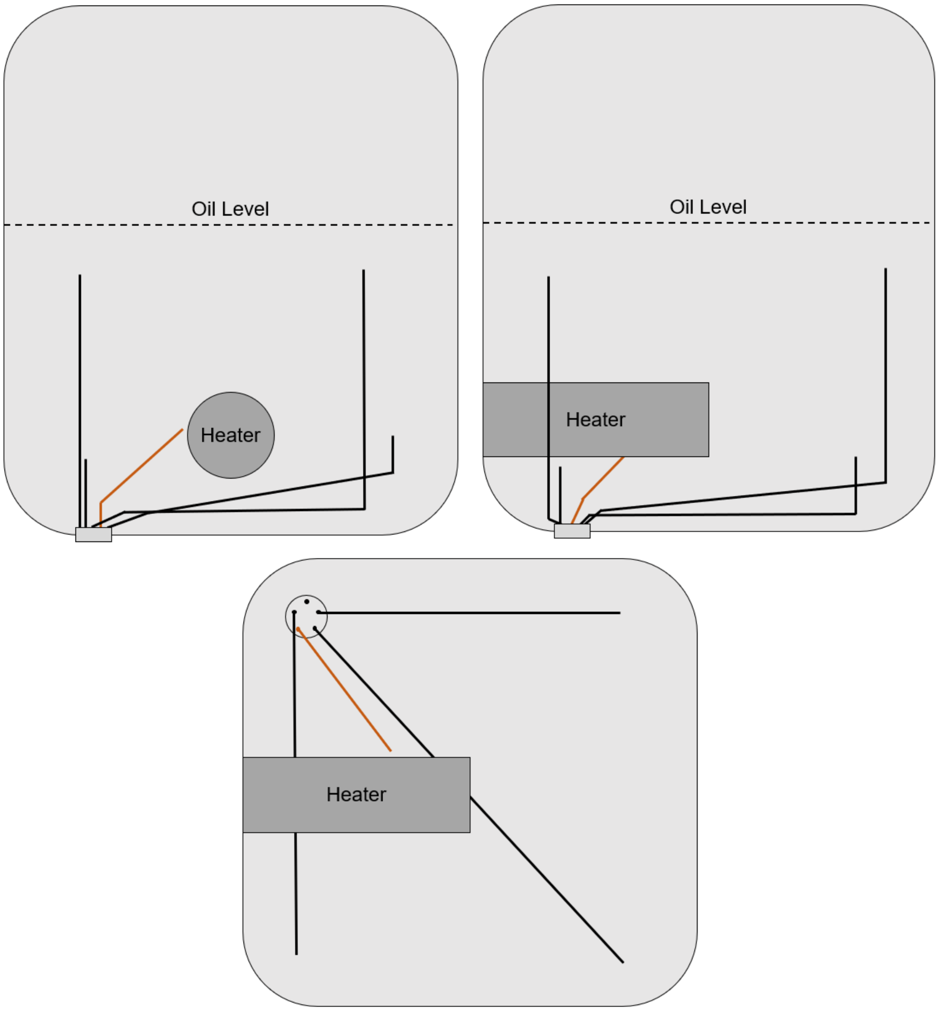


**Figure 3.19:** Front view (top left), side view (top right), and top view (bottom) of the thermocouple and heater arrangement in the oil wells.

### *3.2.4. Improvements*

The bearing facility has seen many improvements since the work of Peck. At the start of this research, two significant changes were made. First, it was discovered that, in all previous tests, the turbine had been spinning backwards. Additionally, a critical error in the LabVIEW program regarding how torque was being measured was found; this bug resulted in torque readings an order of magnitude lower than the true value. Both of these issues were rectified, and the ability to change the turbine's direction of spin was later integrated into testing.

Another major change was the DAQ upgrade from the SCXI-1000 chassis to the PXIe-1075 chassis. Subsequently, this led to the integration of the accelerometers, barometer, hygrometer, and thermocouples, none of which had been present before. Along with the chassis change, the LabVIEW program saw a complete overhaul. The new program processes and displays the data of the new instrumentation (accelerometers, barometer, hygrometer, and thermocouples) as well as the older pieces (tachometer and torque meter). Then new program also automatically controls the turbine speed through the VFD via a PID control loop. Not only is it more convenient to not have to control the speed manually though the VFD front panel, it is also much more precise.

Despite the LabVIEW automation, the "Start" and "Stop" buttons still need to be pressed on the VFD in order to begin or end a test. For ease of operation, the front panel of the VFD was removed from VFD body, and the connecting cable was extended. This change allows for the test operator to be able to use the DAQ and VFD at the same time, at a safe distance from the turbine. Figure 3.20, on the next page, shows this change.

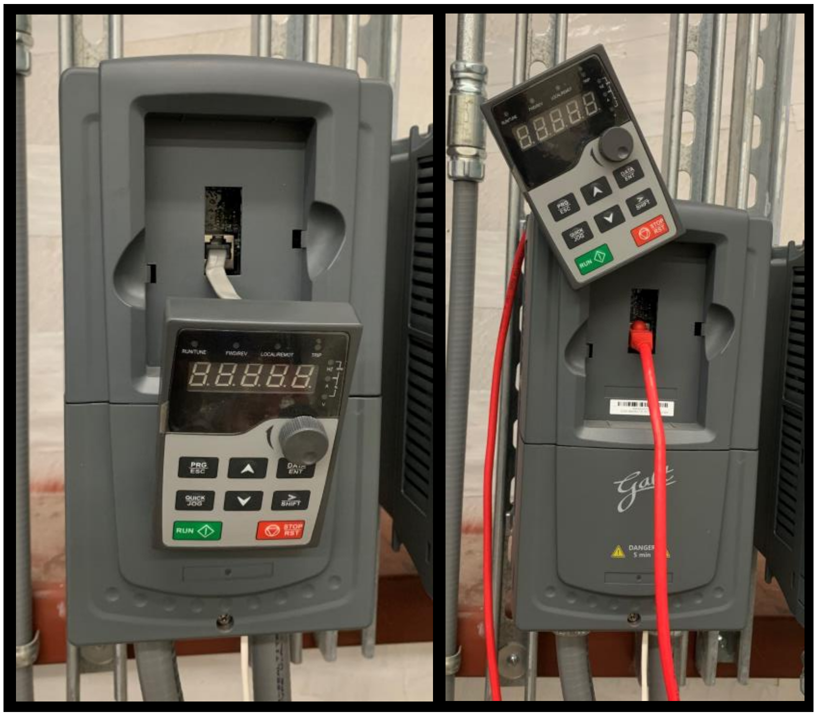


**Figure 3.20:** VFD before (left) and after (right) the cable extension.

Another issue that was remedied was that of excessive oil drain; this was caused by two factors. Firstly, the oil level was set too high. With the spiders in the oil bubblers set as low as they could go, the bubblers themselves had to be re-piped. As part of this process, ball valves were installed on both sides to allow for easy oil sampling/draining. The other factor contributing to the oil drain was a vacuum effect on the governor side. The governor penetration combined with the extra empty volume for the trip mechanism would create a vacuum effect when the turbine was spinning that would pull oil out of the governor side oil well. This was solved by installing a steel plate to cover up the governor penetration, shown in Figure 3.21 on the next page.

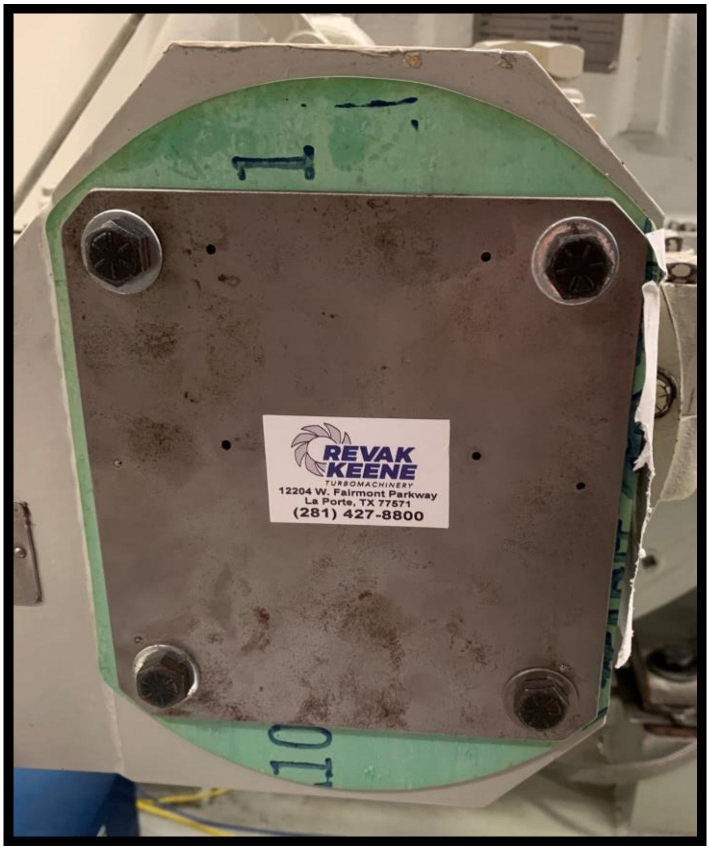


**Figure 3.21:** Steel plate used to seal the governor penetration.

Though, prior to this research, no heated tests had been conducted, the proposed method of heating was to cover the turbine body in a heater wrap and heat the oil via conduction through the turbine. Following an unexpected thermal event during a shakedown test resulting from improper heat dissipation and equipment failure, this method was deemed unsafe, and the switch to immersion heaters was subsequently made.

The final two changes made to the bearing facility was the addition of blind flanges to the turbine inlet and outlet, and custom cut melamine insulation. The blind flanges greatly reduced operational noise, and also slightly reduced heat loss during heated tests. The melamine insulation greatly reduced heat loss, and, compared to fiberglass insulation, was easier and safer to work with; it also did not present the risk of dispersing fiberglass dust into the oil or bearings. Figure 3.22 (same as Figure 3.10) and 3.23, show the initial and final turbine configuration, respectively.

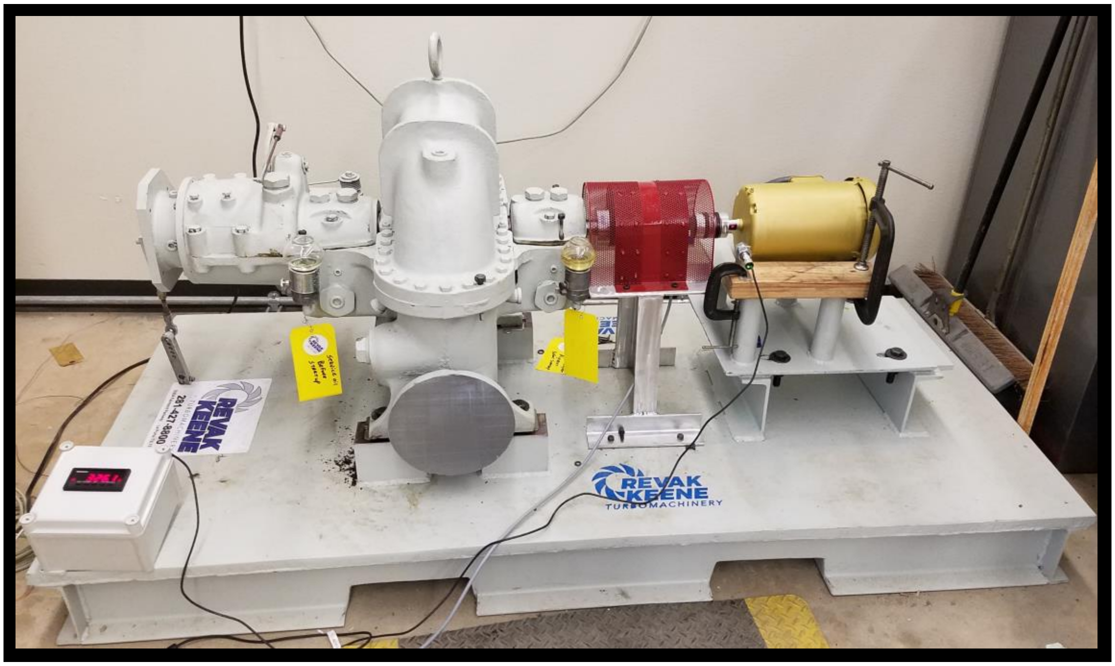


**Figure 3.22:** Turbine configuration at the start of this research.

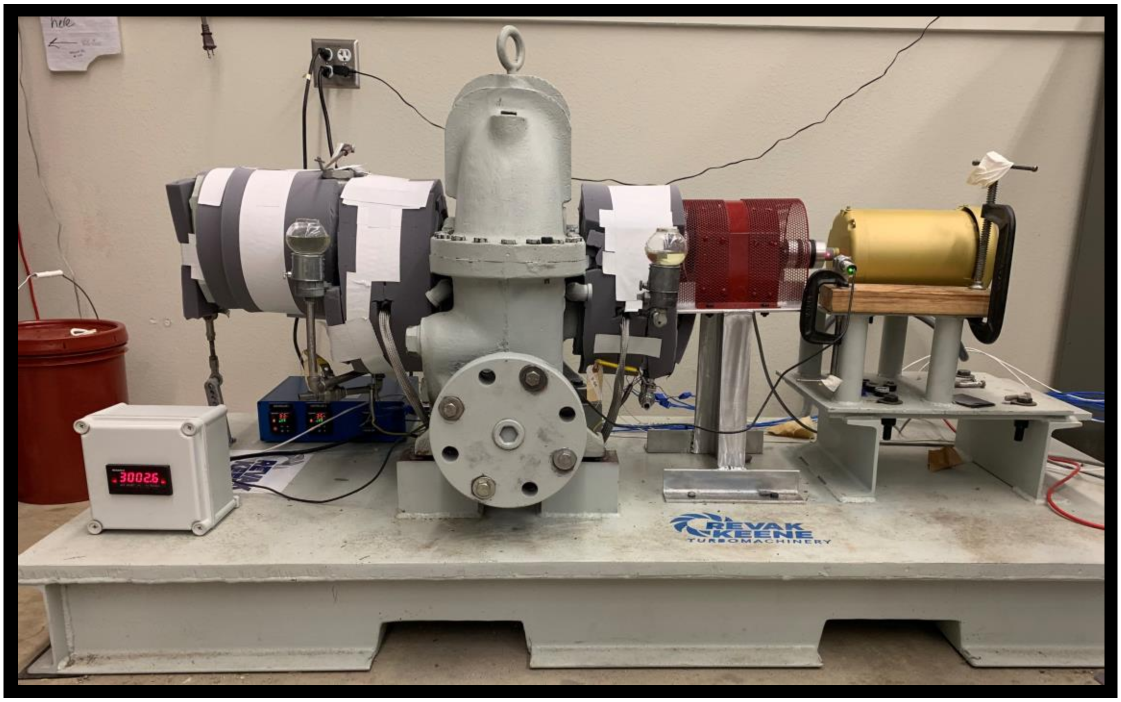


**Figure 3.23:** Turbine configuration at the end of this research.

# 4. TESTING

## 4.1. Oil Degradation Tests

The process of the oil degradation tests consists of heating a volume of oil (or an oil mixture) to a target temperature, and maintaining that temperature (±1$^o$C) for 72 hours. For the duration of the test, temperature and viscosity are recorded, and changes in the oil's appearance are documented. The purpose of these tests was to characterize the thermal degradation of oil over 72 hours at various temperatures, as well as produce degraded oil to be used in bearing tests.

### *4.1.1. Procedure*

The generalized procedure for conducting oil degradation tests is as follows. Beakers are filled with Mobil DTE 732 turbine oil, and then wrapped with two layers of fiberglass insulation. Three thermocouples are placed in the oil of each beaker at varying depths (typically in 1/3 increments), and a viscometer probe is submerged in the oil of one beaker. The beakers are then heated via hot plate inside a fume hood for 72 hours. Every 8 to 12 hours, photos are taken in order to track color changes in the oil. Once the test time has elapsed, the hot plates are turned off, and the insulation is removed. The test is completed once the oil cools down to room temperature. A detailed oil degradation procedure can be found in Appendix C.

### *4.1.2. Test Matrix*

The oil degradation tests performed for this research were selected from the proposed project test matrix, shown in Table 4.1 on the next page. Specifically, two 121$^o$C dry 72-hour tests and one 93$^o$C 10% wet 6-hour test were performed.

TABLE 4.1
TTEXOB Project Proposed Oil Testing Matrix

| x: No Air | A: Aerated | | |
|---|---|---|---|
| Temperature ($^o$C) | 0% Wet | 1% Wet | 5% Wet |
| 71 | x, A | - | x, A |
| 88 | x, A | - | x, A |
| 121 | x, A | x | - |
| 154 | x, A | - | - |
| 199 | x | - | - |

Here, “A” denotes “aerated” and “x” denotes “not aerated”. The aerated tests involve bubbling air into the oil prior to the degradation. Additionally, the “% Wet” refers to the percent of water by volume that was mixed into the oil prior to the degradation.

The 121$^o$C dry tests were selected primarily to provide degraded oil for the bearing tests. Additionally, these and the 93$^o$C wet test were conducted to provide preliminary oil data, as well as insights into future testing possibilities. The rest of the oil tests presented in Table 4.1 will be conducted later in a Texas A&M petroleum engineering oil analysis lab.

## 4.2. Bearing Tests

The process of the heated bearing tests consists of inspecting and documenting the state of the turbine’s bearings prior to operation, operating the turbine at a constant speed for 3 to 6 hours, and then inspecting the bearings for any changes. These tests can be done with fresh or degraded oil, and with or without applied heat. During turbine operation, shaft speed and torque, as well as oil temperature, turbine vibration, room temperature, and ambient humidity are recorded. The purpose of these tests is to form a baseline set of data representing how these turbines behave under operational conditions, and to replicate the accident conditions of Fukushima in order to glean potential insights into the behavior of the RCIC system at that time.

### *4.2.1. Procedure*

Prior to and after testing, a bearing inspection must be completed. The purpose of these inspections is to reveal any changes that may have occurred in the bearings during the test. This process involves removing the bearings from the turbine, wiping them down with a lint free cloth, investigating the bearings for any defects or changes, photographing them, and then restoring them to their proper places inside the turbine. A detailed bearing inspection procedure can be found in Appendix C.

Following inspection, the test is conducted. For bearing stress tests, the turbine is simply maintained at a specified speed for 3 hours. Oil temperature, shaft speed, torque, vibrational data, and ambient weather data is collected. These tests are used to characterize the effect of operation under certain conditions on the performance of the turbine. For steady state torque tests, data is recorded in 5-minute increments at various speeds for both forward and reverse spin of the turbine (the reverse spin tests were conducted to provide data for modeling). Only torque and shaft speed values are considered for these tests. This data is used to generate a torque vs speed curve. Both of these two types of tests are conducted with either fresh or degraded oil, and can be done at room temperature or with applied heat. More detailed procedures for both the bearing stress tests and the steady state torque tests can be found in Appendix C.

### *4.2.2. Test Matrix*

The heated bearing tests performed for this thesis were inspired by the proposed project test matrix, shown in Table 4.2 on the next page.

TABLE 4.2
TTEXOB Project Proposed Heated Bearing Testing Matrix

| S: 6 hours | | | |
|---|---|---|---|
| Temperature ($^oC$) | 1000 rpm | 3000 rpm | Alternating |
| 121 | S | S | S |
| 154 | - | S | - |

Here, “S” is the maximum test time; the test times range from 3 to 6 hours. Additionally, the “Alternating” speed test simply means the shaft speed alternates between a constant 1000 rpm and a constant 3000 rpm during turbine operation.

The shaft speed values of 1000 and 3000 rpm from Table 4.2, as well as the temperature value of 121$^{o}$C, were interpreted as values of importance and were therefore used as the main testing conditions. The actual tests performed can be found below; Table 4.3 shows the bearing stress tests, and Table 4.4 shows the steady state torque tests.

TABLE 4.3
Testing Matrix for Bearing Stress Tests

| | Temperature ($^{o}$C) | | | |
|---|---|---|---|---|
| Speed (rpm) | Fresh Oil | | Degraded Oil | |
| 1000 | - | 121 | - | 121 |
| 3000 | - | 121 | - | 121 |

TABLE 4.4
Testing Matrix for Torque vs Speed Tests

| | Temperature ($^{o}$C) | | | |
|---|---|---|---|---|
| Speed (rpm) | Fresh Oil | | Degraded Oil | |
| -500 | - | 121 | - | 121 |
| -1000 | - | 121 | - | 121 |
| -1500 | - | 121 | - | 121 |
| -2000 | - | 121 | - | 121 |
| -2250 | - | 121 | - | 121 |
| -2500 | - | 121 | - | 121 |
| 500 | - | 121 | - | 121 |
| 1000 | - | 121 | - | 121 |
| 1500 | - | 121 | - | 121 |
| 2000 | - | 121 | - | 121 |
| 2500 | - | 121 | - | 121 |
| 3000 | - | 121 | - | 121 |
| 3600 | - | 121 | - | 121 |

For both tables, “Temperature” refers to the temperature of the oil inside the turbine, and a blank field indicates a room temperature condition. In Table 4.4, negative speeds indicate reverse spin of the turbine.

# 5. RESULTS

## 5.1. Oil Degradation Tests

The oil degradation portion of this research consisted of three tests: two 121$^{o}$C 72-hour tests, and one 93$^{o}$C 10% wet 6-hour test. The objectives for these tests were to study the effects of oil degradation under different circumstances, and to produce degraded oil for use in the bearing tests. For all three of these tests, both temperature and viscosity were recorded with respect to time. For figures that differentiate between a “left” and “right” beaker, the viscometer was always placed in the “right” beaker. Additionally, the temperature data expressed in the figures of this thesis (unless otherwise stated) is the average of three different thermocouples placed equidistant from each other at a shallow, medium, and low depth in the oil. It is also important to note that glass beakers were used for the initial 121$^{o}$C test (2/18/2019) and the 10% wet test (3/7/2019), whereas stainless steel beakers were used for the second 121$^{o}$C test (7/8/2019). Lastly, the entire set of relevant graphical and pictorial data collected from these tests can be found in Appendix A.

Going in chronological order, the first test would be the initial 121$^{o}$C degradation, which was performed on 2/18/2019. Figure 5.1, on the next page, shows the average oil temperature over time. The two beakers had a very similar, steady trend, and maintained roughly ±1$^{o}$C from the target temperature. The right beaker was slightly more stable, due to the added thermal inertia from the presence of the viscometer. Heat up was completed at about the 5-hour mark, temperature was maintained for 72 hours, and the degradation was completed at the 77-hour mark.

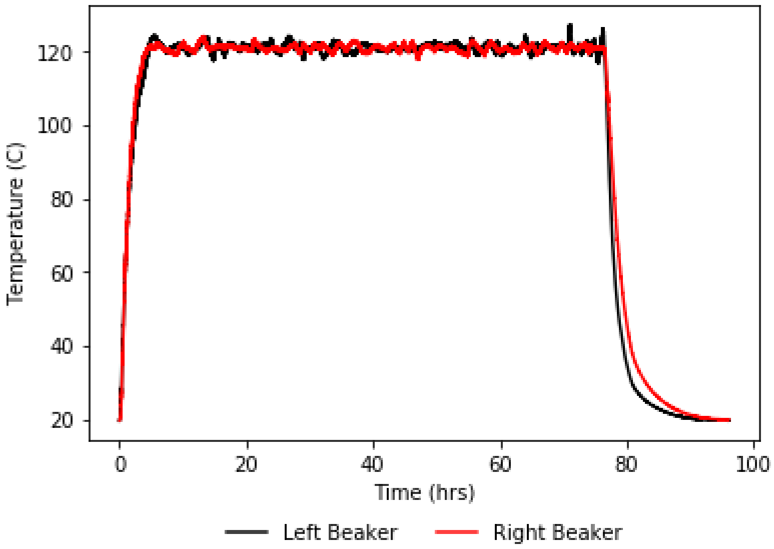


**Figure 5.1:** Average oil temperature over time for the initial 121ºC oil degradation.

Figure 5.2, on the next page, illustrates the relationship between viscosity and temperature, for both fresh and degraded oil. Here, the "fresh" curve was generated by isolating the viscosity during the heat up process (20ºC – 121ºC), and the "degraded" curve was generated by isolating the viscosity during the cooldown process (121ºC – 20ºC). As was expected, the viscosity decreased as the oil temperature increased. At room temperature (~22ºC) the oil had a viscosity of about 85 cSt, and at 121ºC the oil had a viscosity of about 4 cSt; this equates to a 95% decrease. The jaggedness of the "fresh" curve can be explained by convection currents; these currents were visible throughout the test. During heat up, the currents carry hot, less viscous oil up into the cold region of the beaker where it impinges upon the viscometer. The hot oil is then replaced with cooler, more viscous oil, resulting in the jagged curve. The "degraded" curve is smoother because, at the end of the test, the temperature profile (and therefore the viscosity) is much more uniform; this effect shows up in all of the viscosity plots. The two curves are fairly similar, but at first glance it would appear that there is some difference between the fresh and degraded oil; this is actually not the case, and will be explained later.

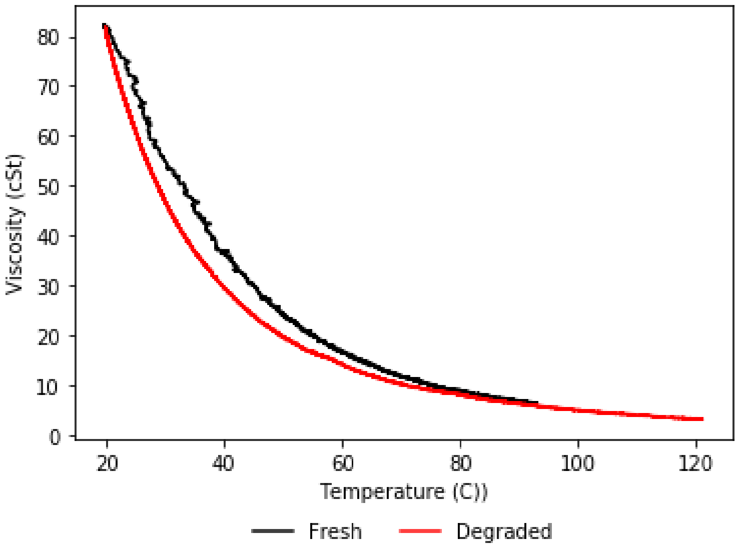


**Figure 5.2:** This plot shows the relationship between viscosity and temperature before and after the degradation, for the initial 121$^{o}$C oil degradation.

A color comparison between fresh oil and the degraded oil produced from this test is shown in Figure 5.3, on the next page. Over the course of the 72-hour degradation the oil color noticeably changes from a very light/clear color, to a yellow/gold color. However, color change in oils can happen at extremely low levels of degradation, and does not necessarily indicate a significant chemical change.

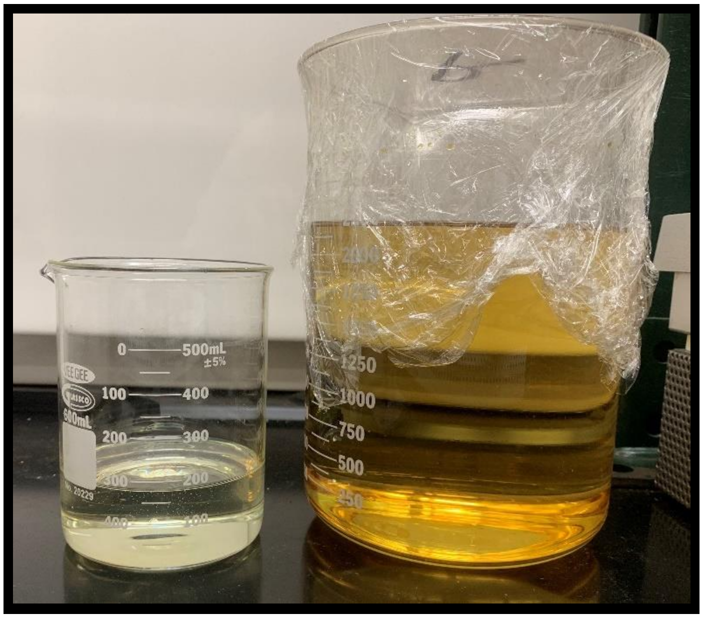


**Figure 5.3:** Shown here is a comparison between oil prior to (left) and after (right) the initial 121°C oil degradation.

The next test performed was the 93°C 10% wet 6-hour degradation. The focus of this test was less about the oil degradation itself, and more about the effects of adding water to the oil while it degraded. There were two noteworthy occurrences during this test. First, a flashing event occurred in the stratified water early on in the test. The water acted as an insulator to the oil and heated up very quickly, but was unable to boil due to the fluid pressure applied by the oil. As a result, the temperature of the water briefly exceeded 110°C before a portion of it flashed into steam, ejecting some of the mixture from the beaker. The hazardous nature of this event was the reason why a shorter test time (6 hours vs 72 hours) was used, and why the test has not been attempted again. Figure 5.4, on the next page, shows the individual thermocouple readings over time.

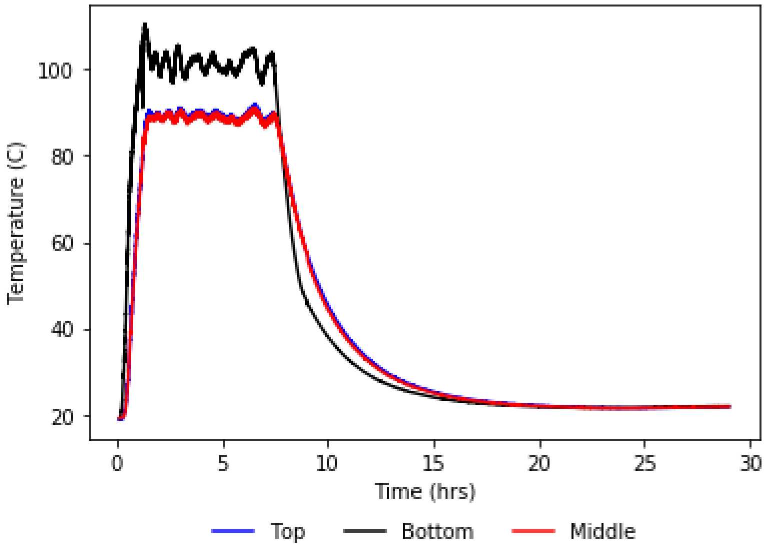


**Figure 5.4:** Temperature over time for the 93°C 10% wet oil degradation.

The “top” and “middle” thermocouples were both placed in the oil; the “bottom” thermocouple was placed in the water. For the duration of the test, there was a significant difference in temperature between the water and the oil (>10°C). Interestingly enough, the flashing event actually shows up in the figure. Looking at the water temperature around the 1-hour mark, there is a sharp spike down to 90°C, immediately followed by a sharp spike up to 110°C. This phenomenon also presents itself in the viscosity vs temperature plot of Figure 5.5, on the next page.

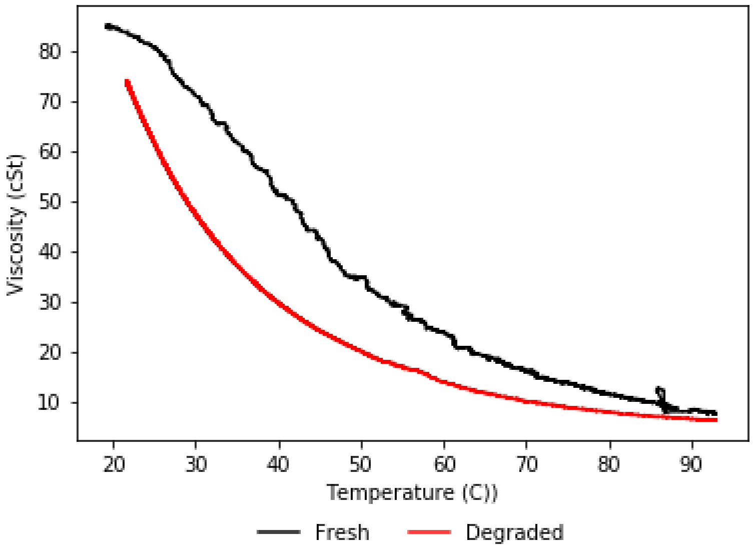


**Figure 5.5:** This plot shows the relationship between viscosity and temperature before and after the degradation, for the 93°C 10% wet oil degradation.

Just past the 85°C mark for the “fresh” curve, the flash event artifact can clearly be seen. Additionally, the “fresh” curve for this test is slightly more linear than that of the 121°C dry test, undoubtably due to the addition of water. This leads into the second interesting occurrence of this test.

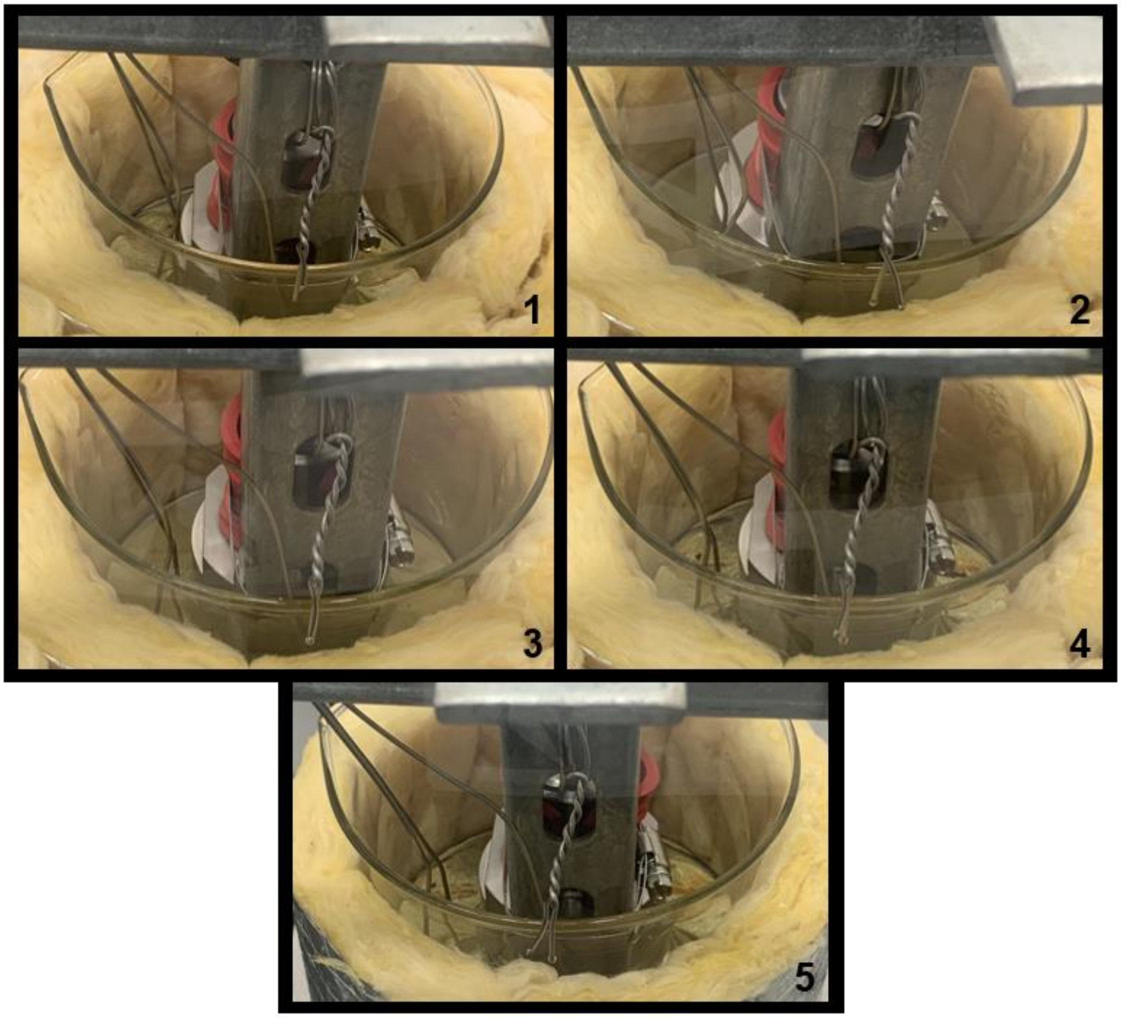


**Figure 5.6:** Progression of the oil/water mixture's appearance over the course of the test. The timestamps in order of numbering are 0:51:00, 1:08:00, 2:19:00, 4:06:00, and 6:18:00.

Over the course of this test, the oil/water mixture changed very little in color, but the opacity made several changes. Figure 5.6, above, shows the progression of the mixture's appearance throughout the experiment. At the start of the experiment (1), the oil was clear. However, as heat was applied to the mixture (2), the water began to boil into the oil, forming a water-in-oil emulsion. Though, as the heating process continued (3, 4, and

5), the oil became clear again; this is because the mixture developed a more uniform temperature gradient. As the temperature of the water and the oil increased, their respective viscosities (and in turn, their interfacial viscosity) decreased. This led to an increase in water droplet coalescence and emulsion separation.[9, 10] However, once the test had reached its completion and the temperatures began to decrease, these rates also decreased, leading to the return of the emulsion. Over the course of the next week, gravity acted on the mixture to separate the suspended water from the oil, and return it to its original state; Figure 5.7, below, gives a comparison of the initial end state of the mixture and its appearance one week later.

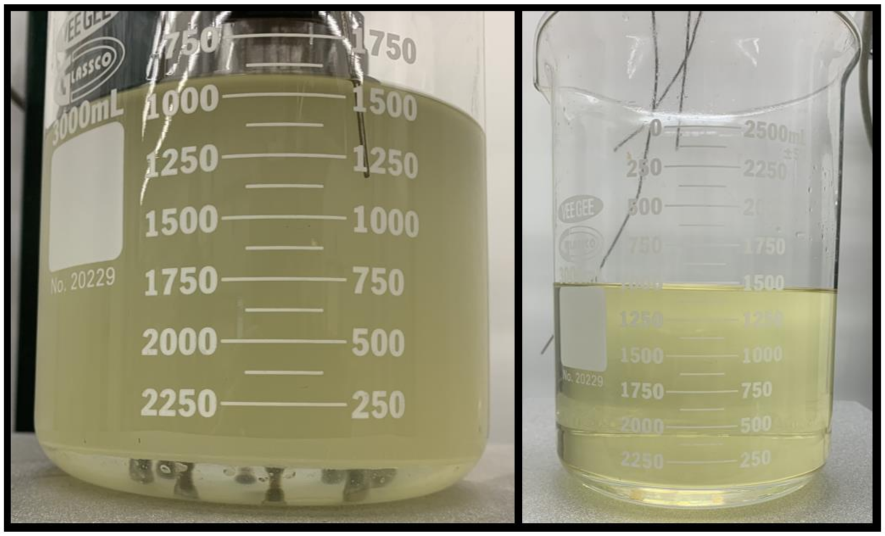


**Figure 5.7:** Shown here is the oil/water mixture the day after the test (left), and one week later (right).

Returning to the discussion of viscosity (Figure 5.5), the appearance of the emulsion during heat up accounts for the large separation between the “fresh” and “degraded” curves. Though counterintuitive, it is possible for an emulsion to have a higher effective viscosity than the respective viscosities of the components. Therefore, during heat up, when the emulsion was initially present, a larger, more linear viscosity profile is seen. During the cool down, before the emulsion reappeared, the viscometer was essentially reading pure oil; this resulted in the “normal” viscosity curve.

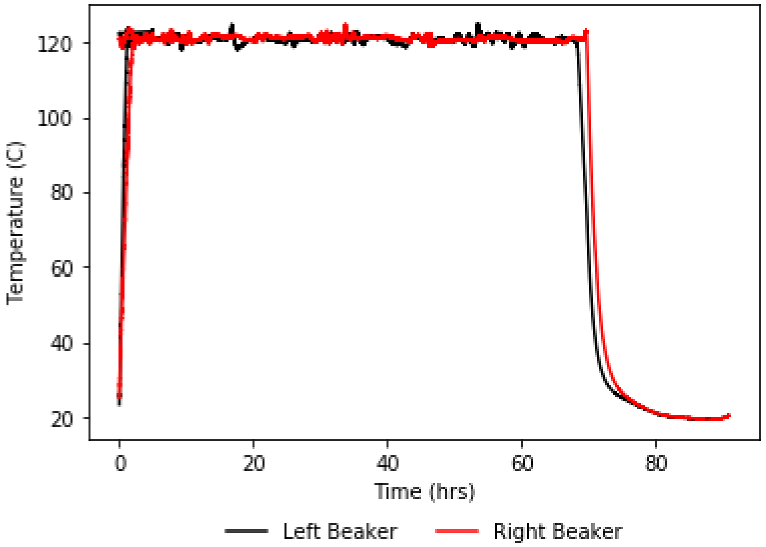


**Figure 5.8:** Average oil temperature over time for the second 121ºC oil degradation.

The final oil test was the second 121ºC degradation, performed on 7/08/2019. An important thing to note about this test is that, unlike the preceding degradations, this one utilized stainless-steel beakers. Figure 5.8, on the previous page, shows the average oil temperature over time. Again, both beakers had a similar, steady trend, and maintained roughly ±2$^o$C from the target 121$^o$C. As was expected, the heat up process was noticeably faster, finishing in approximately 2.5 hours (about twice as fast as with the glass beakers). This is because the stainless-steel conducts heat from the base of the beaker, up the walls, and provides a more uniform heating profile.

The effect of the stainless-steel beakers on viscosity, though, was unexpected. Figure 5.9, below, shows the relationship between viscosity and temperature, for both the degraded and undegraded oil. Unlike the first 72-hour degradation, the viscosity curves for this test are almost identical. This serves as very strong evidence that a chemical change does not occur after a 72-hour degradation at 121$^o$C, and that some other phenomenon is responsible for the separated viscosity curves.

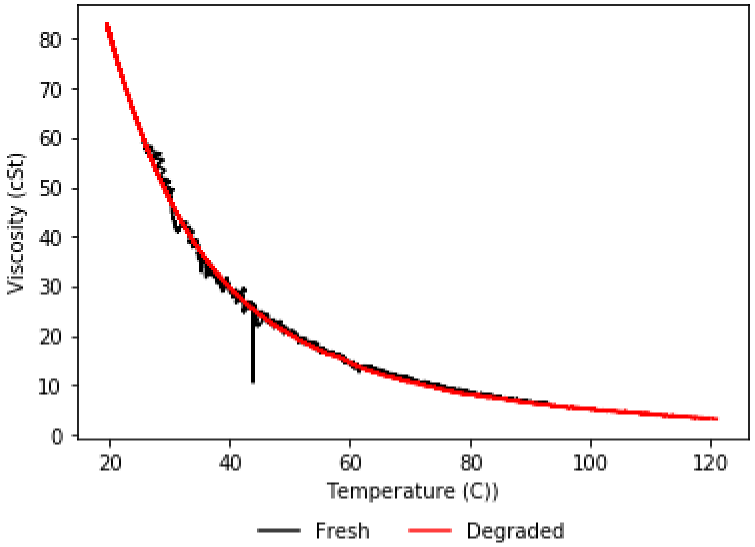


**Figure 5.9:** This plot shows the relationship between viscosity and temperature before and after the degradation, for the second 121$^o$C oil degradation.

Figure 5.10, below, compares the viscosity curves of the two 72-hour tests. First, the two "degraded" curves are nearly indistinguishable. This, combined with the fact that the "fresh" curve from the test utilizing stainless steel beakers is also very similar, indicates that at this level of degradation, little to no chemical change in the oil occurs. The most likely cause for this "separation" is that the glass beakers lack the "wall heating" effect of the stainless-steel beakers, and therefore do not develop a uniform temperature profile until the end of the heat up process. This results in a "cold zone" around the viscometer, which causes it to record viscosity values higher than what they should be.

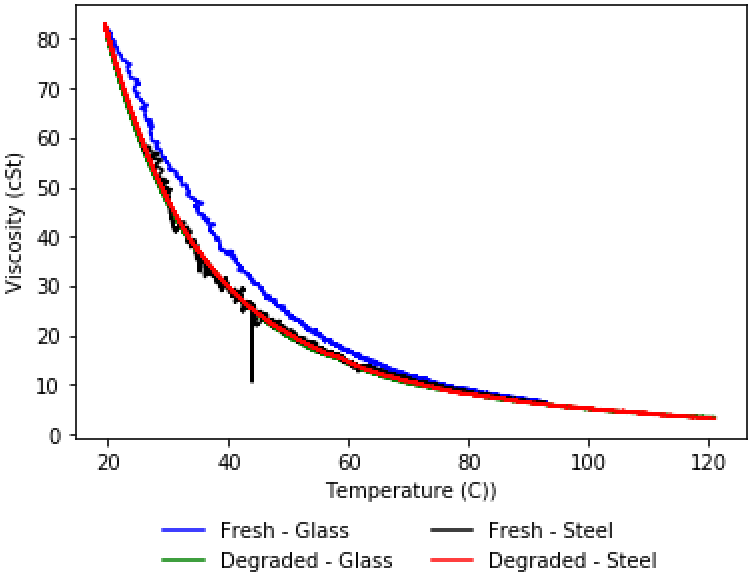


**Figure 5.10:** Comparison between the two sets of viscosity vs temperature curves, for the two 72 hour tests.

Figure 5.11, below, compares the viscosity curves of the initial dry test and the 10% wet test (both with glass beakers). The separation between the "fresh" curves is most likely the result of two factors. First, the stratified water layer acted as an insulator to the oil, providing a similar effect as the steel vs glass beaker case. Second, with the water inhibiting the heating of the oil while being boiled into it, the mixture was well emulsified early into the heat up process. Interestingly enough, though, the two "degraded" curves are identical. At the start of the cooldown, the mixture was separated (as indicated by Figure 5.6), and while the water did cool down faster, there was not a significant temperature difference between the two fluids. Based off the information provided by Figure 5.7 and 5.11, it is also likely that the second emulsion had very little water suspended in it; this would lead the "degraded" wet curve to be as close as it is to the "degraded" dry curve.

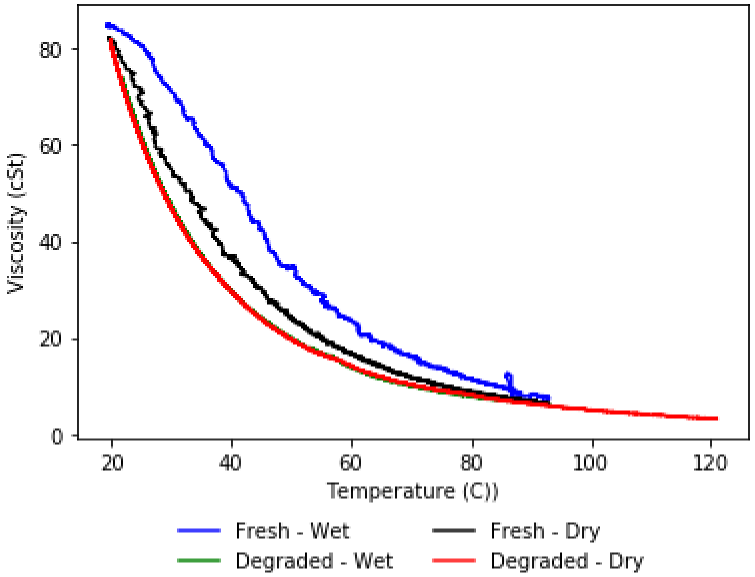


**Figure 5.11:** Comparison between the two sets of viscosity vs temperature curves, for the initial dry test and the 10% wet test.

## 5.2. Bearing Tests

The results for the bearing tests stem from two different kinds of tests: bearing stress tests, and steady state torque tests. The primary objective of the bearing stress tests was to operate the turbine for an extended period of time (3 hours) under different conditions, and analyze the effect (if any) on the bearings. These tests were conducted with fresh and degraded oil, at room temperature and 121ºC, and at 1000 and 3000 rpm, for a total of 8 tests. As for the steady state torque tests, the main interest was how the same parameters (oil quality, oil temperature, and shaft speed) affected the performance of the turbine as a whole. In this case, the parameter of interest was motor torque. For both sets of tests, shaft speed, motor torque, oil temperature, and vibrational frequency were measured. It should be noted that halfway through the testing period the accelerometers (mostly accelerometer 3) began supplying erroneous vibrational data; it was then discovered that some of the wiring had been damaged. All of the accelerometer wiring was replaced on 12/09/2019, so accelerometer data recorded prior to this date should only be used as a rough estimation. Fortunately, enough, this data ended up not being crucial to the investigation. The desire for vibrational data was to investigate potential harmonic frequencies of the turbine, and to externally monitor damage to the bearings; ultimately, neither of these occurred. The full collection of data obtained from the bearing stress tests and the steady state torque tests can be found in Appendix B.

Starting with the bearing stress tests, it is important to remember that the objective of these tests was to characterize the effect of the parameters of interest (oil temperature, oil quality, and shaft speed) by how the bearings were physically affected. Figure 5.12, on the next page, shows a standard shaft speed over time plot for a 3000 rpm test.

Specifically, this test was unheated with fresh oil, and was performed on 11/12/2019. Shaft speed is a controlled parameter, so this data (at least for the bearing stress tests) provides no information, other than to verify that the desired speed was maintained. As a side note, the spikes in speed at the very start and very end of the test are due to a defect in the tachometer; at low speeds (< 100 rpm) the tachometer has a tendency to double or triple count a single rotation.

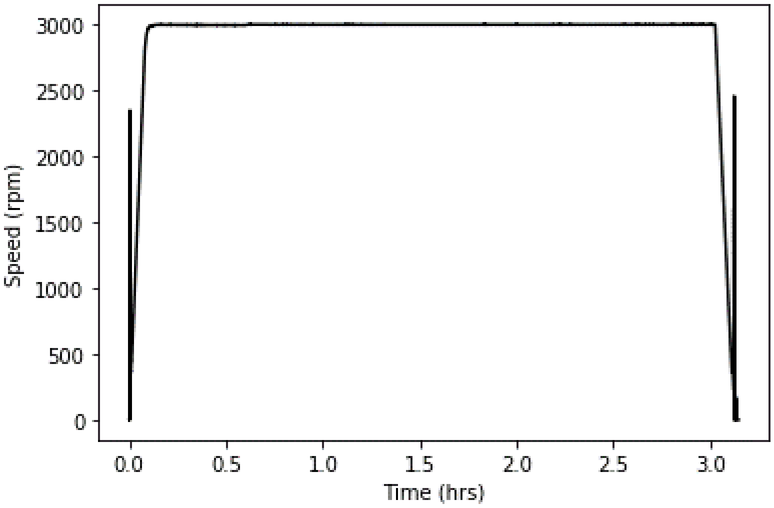


**Figure 5.12:** Rotational shaft speed over time for the unheated, fresh oil, 3000 rpm test.

Though more relevant to the steady state torque tests than the bearing stress tests, Figures 5.13 and 5.14 show motor torque over time for the 1000 and 3000 rpm tests, respectively. For both plots, the initial 8 Nm spike is the result of the motor spinning up the turbine, and the negative spike at the end in from the motor slowing the turbine down (hence, the negative torque). Also in both plots, the two unheated tests are nearly identical, as well as the two heated tests; this would suggest that oil quality (at least at this level of degradation) has no impact on turbine performance. As expected, both sets of heated tests had lower torque values than the unheated tests. This is due to the fact that increasing the oil temperature reduces its viscosity, which in turn alleviates some of the frictional load from the shaft. While torque does steadily decrease over time for all tests (due to added heat from friction), the bulk of the torque drop in the heated tests occurs in the first 30min of the test; this is when most of the heating occurs.

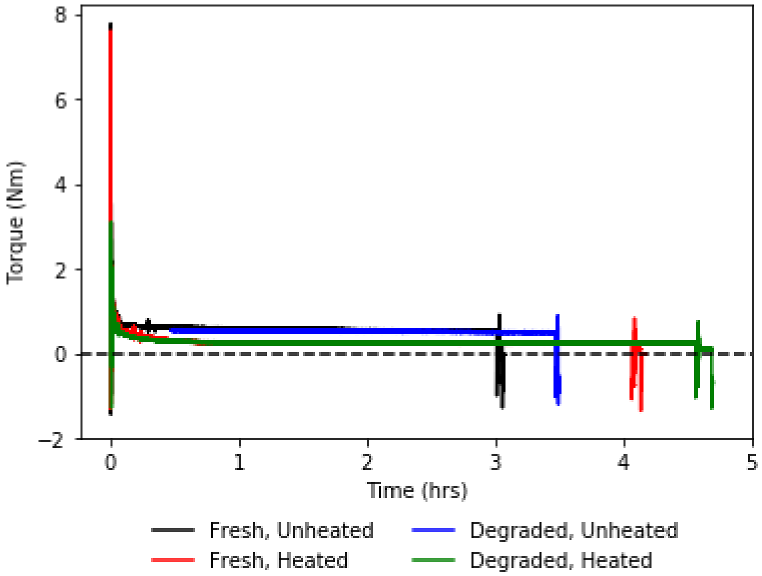


**Figure 5.13:** Motor torque over time for the four 1000 rpm bearing stress tests.

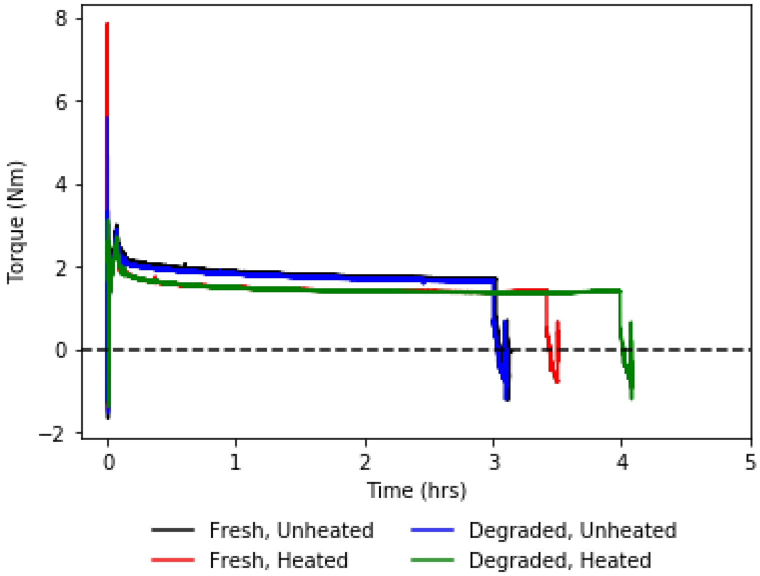


**Figure 5.14:** Motor torque over time for the four 3000 rpm bearing stress tests.

By only looking at the steady state zone of torque (defined as the 0.5 – 2.5 hours timestamp zone), the relationship between torque and temperature can be analyzed. As one might expect, based on the previous results, the only significant factor was shaft speed. In all cases, torque decreased with increasing temperature (similarly as to what was seen in Figures 5.13 and 5.14). However, 3000 rpm tests had a strictly linear trend, whereas 1000 rpm tests were linear up to ~90°C, and then began to exhibit quadratic behavior. Examples of these torque vs temperature relationships can be found in Figure 5.15 and 5.16.

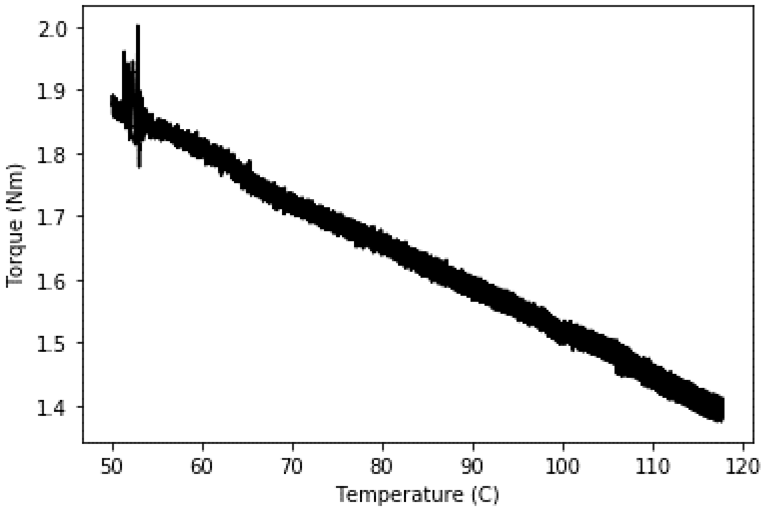


**Figure 5.15:** This plot shows the relationship between motor torque and oil temperature for the heated, degraded oil, 3000 rpm test.

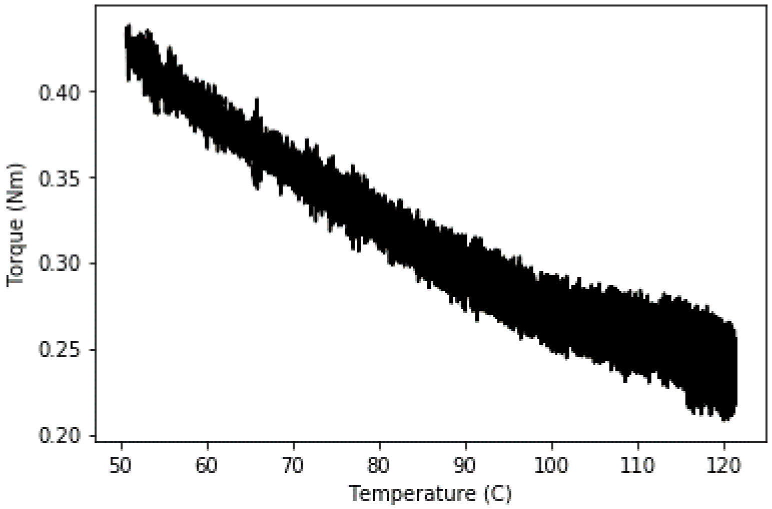


**Figure 5.16:** This plot shows the relationship between motor torque and oil temperature for the heated, degraded oil, 1000 rpm test.

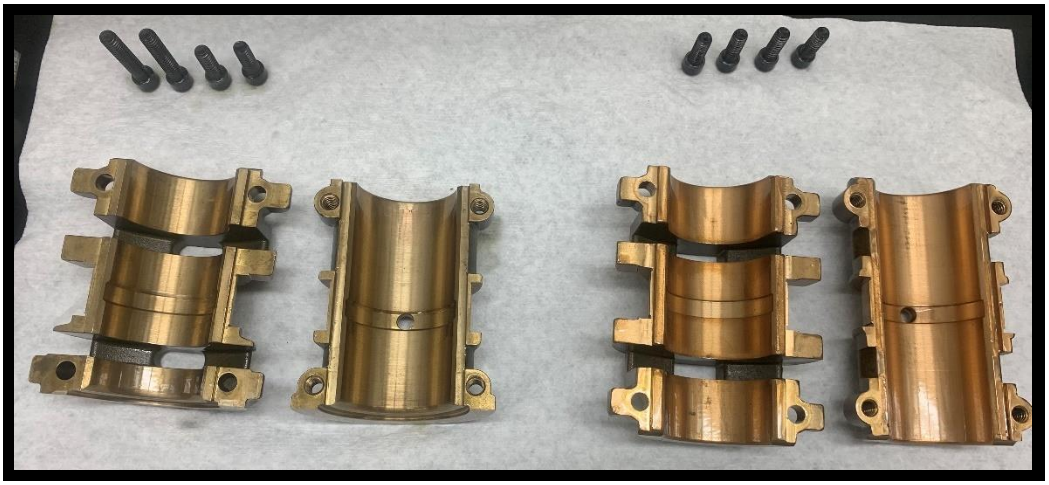


**Figure 5.17:** ZS-1 journal bearings; on the left is the governor side set, and on the right is the motor side set.

Figure 5.17, above, shows the complete journal bearing set of the ZS-1 Terry turbine. Each piece is unique, and installing them incorrectly will cause the turbine to fail. From left to right there is the governor side top bearing, governor side bottom bearing, motor side top bearing, and motor side bottom bearing; the motor side is the drive end, and the governor side is the non-drive end. At the time of this image, the turbine had only gone through shakedown tests with fresh oil, and the bearings were undamaged.

The significant findings from these tests came specifically from the degraded, unheated, 3000 rpm test (performed on 01/06/2020). After performing multiple shakedown tests, all of the fresh oil tests (steady state torque included), and both degraded, unheated tests, the bearings finally showed signs of wear. The damage can be seen in Figure 5.18, on the next page.

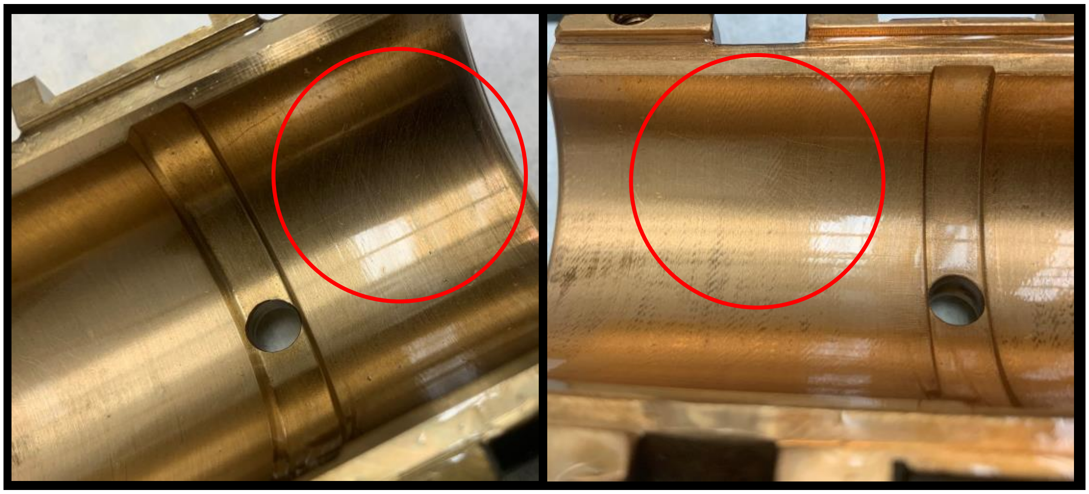


**Figure 5.18:** Shown here is the damage to the bottom bearings (left - governor side, right - motor side) following the unheated, degraded oil, 3000 rpm test.

Slight scoring of the bottom bearings is a standard occurrence. This is because the shaft is supported by the bottom bearings prior to the establishment of the oil wedge. However, the scoring here is at an angle to the direction of rotation of the shaft; this is unconventional damage. The most likely explanation is that the slinger rings picked up small particulates in the oil, and carried them into the bearings; a shaft/bearing interaction is unlikely to have occurred. While the damage is very minor, and no other defects appeared in subsequent tests, these findings are important to note. First, while it appears that mildly degraded oil has no effect on turbine performance, it can clearly cause at least small amounts of damage. This level of wear over an extended period of time will most certainly have an effect on turbine performance. Secondly, the bearings of the ZS Terry turbine are made of brass, whereas the bearings of the GS Terry turbine (the model used in the RCIC system) are made of Babbitt. Babbitt is much softer than brass, and will therefore take on damage much faster.

Moving on to the results of the steady state torque tests, it is important to distinguish that “effect on turbine performance” here is defined as affecting the motor torque. There were four steady state torque tests done in total; one for each combination of temperature and oil quality. Each test involved 13 different shaft speeds, including both forward and backward spin of the turbine. Figures 5.19, 5.20, 5.21, 5.22, and 5.23 show the findings of these tests.

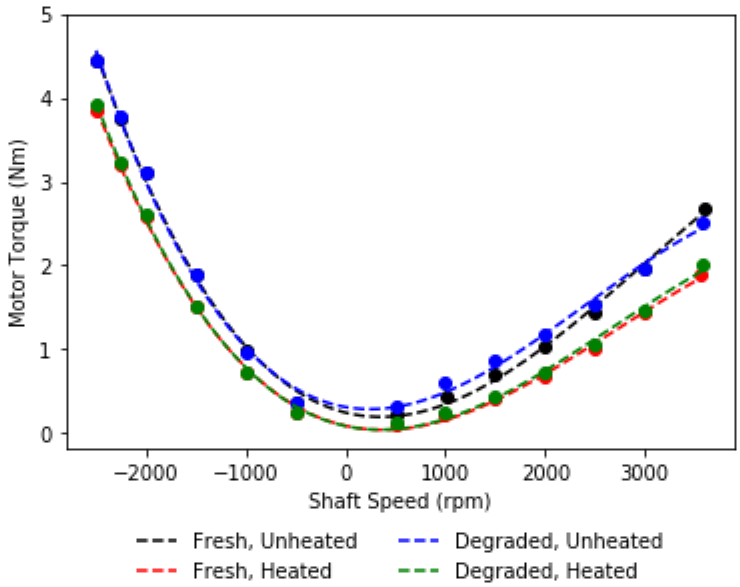


**Figure 5.19:** Comparison of all of the steady state torque tests.

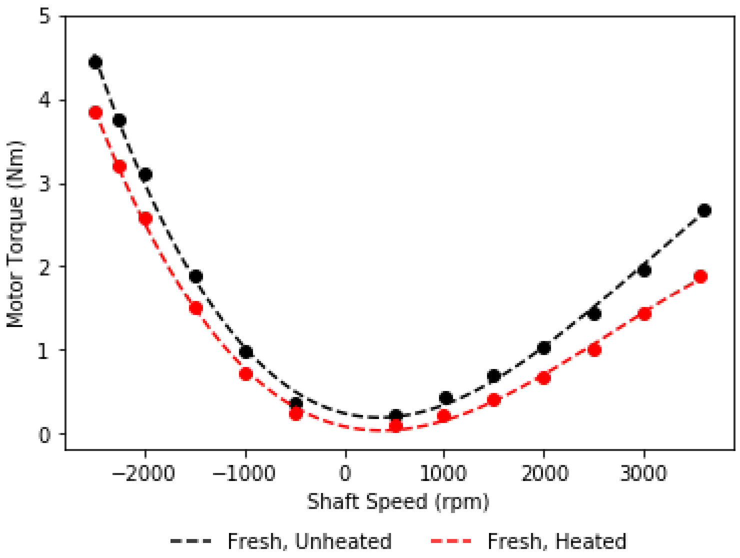


**Figure 5.20:** Comparison of the fresh oil steady state torque tests.

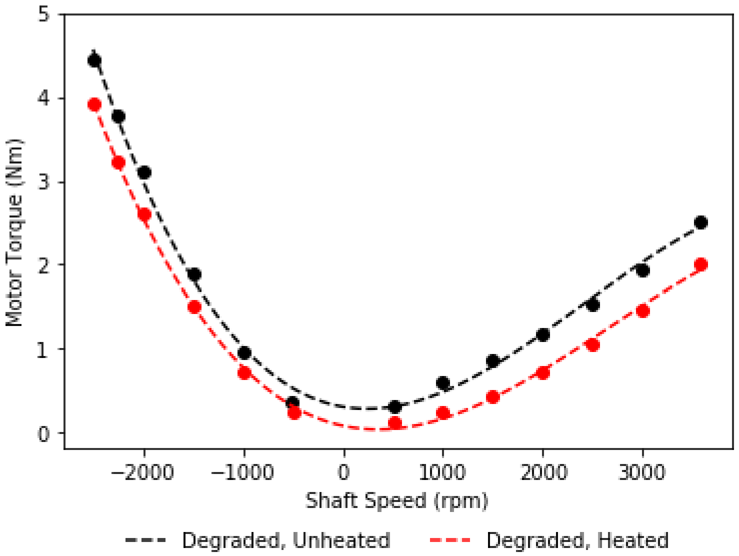


**Figure 5.21:** Comparison of the degraded oil steady state torque tests.

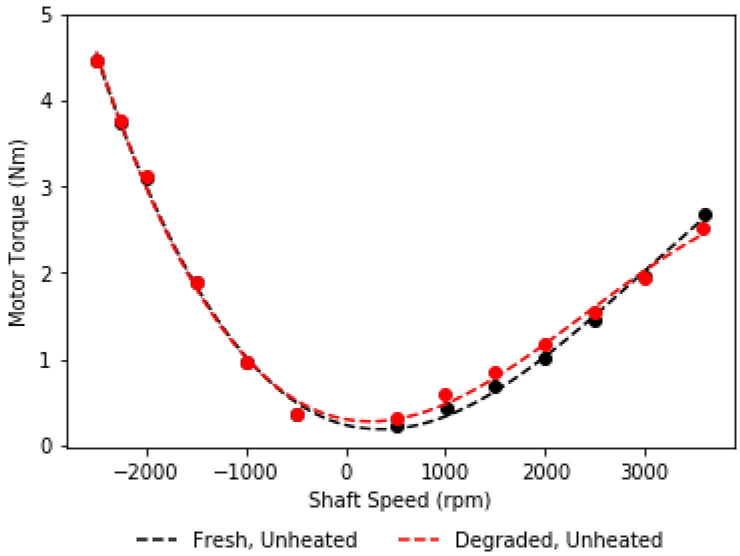


**Figure 5.22:** Comparison of the unheated steady state torque tests.

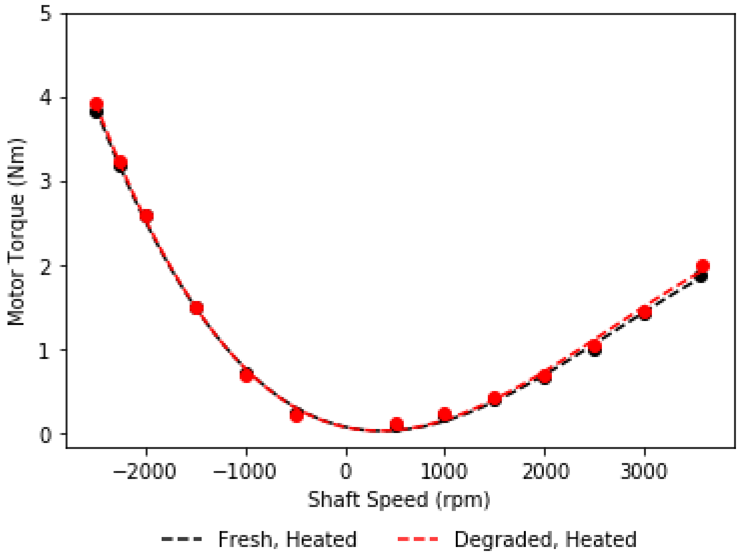


**Figure 5.23:** Comparison of the heated steady state torque tests.

The lines in these plots were generated by computationally fitting regression lines to their respective datasets collected through the steady state torque tests. All of these lines have an $R^2$ value of .99 or greater. The equations for these lines are provided in the “Steady State Torque” section of Appendix B.

Evidently, only oil temperature has an effect on turbine performance. In all cases, heated tests resulted in a lower torque than their unheated counterparts. Oil quality (at least at this level of degradation) appears to have no effect on turbine performance. For almost all of the oil quality comparisons, the torque curves lined up perfectly. The exception to this statement is the unheated comparison for the forward spin case. The reason for this is that the points resulting from both tests were generated in opposing order. The fresh, unheated test was performed in reverse chronological order, starting with 3600 rpm, and ending with 500 rpm. The degraded, unheated test was performed in chronological order, starting with 500 rpm, and ending with 3600 rpm. Because the oil heats up during turbine operation, the torque values at the beginning and end of these curves differ. Had the data from the two tests been collected in the same order, the two curves would have most likely lined up.

### 5.3. Uncertainty Analysis

Outside of observations, the data recorded for these for these tests was provided entirely by automated instrumentation. Therefore, the most prolific source of uncertainty was that of the data collection equipment.

For the oil degradation tests, oil temperature and viscosity were the measured parameters. Because the viscometer's uncertainty is a percentage of its reading, the maximum uncertainty here can be found at the highest recorded value, 85 cSt. With an associated uncertainty of ±4%, the numerical uncertainty at this value equates to ±3.4 cSt. The thermocouples have a flat uncertainty (up to 250$^{o}$C), so the cumulative uncertainty of the temperature average can be calculated using error propagation across the mean; the progression towards the solution starts with Equation 5.1.

$$T_{avg} = \frac{T_1+T_2+T_3+\ \dots T_n}{n} \qquad (5.1)$$

$$T_{avg} = \frac{T_1}{n}+\frac{T_2}{n}+\frac{T_3}{n}+\ \dots\ \frac{T_n}{n} \qquad (5.2)$$

Equation 5.1 is the standard form of a mean value calculation. Here, $T_{avg}$ is the average temperature value, $T_n$ is the temperature reading in the n$^{th}$ thermocouple, and $n$ is the number of thermocouples. By distributing the $\frac{1}{n}$ term across the temperature values, the equivalent Equation 5.2 is formed. With this form in mind, uncertainty can be calculated using additive error propagation with multiplication of a constant; the base form of this is shown by Equation 5.3.

$$\delta T_{avg} = \sqrt{\left(\frac{\delta T_1}{n}\right)^2+\left(\frac{\delta T_2}{n}\right)^2+\left(\frac{\delta T_3}{n}\right)^2+\ \dots\ \left(\frac{\delta T_n}{n}\right)^2} \qquad (5.3)$$

Here, $\delta T_{avg}$ is the uncertainty in the temperature average, and $\delta T_n$ is the uncertainty associated with the nth thermocouple. Under the assumption that all of the thermocouples have equal uncertainty, Equation 5.3 simplifies down to Equation 5.4.

$$\delta T_{avg} = \frac{\delta T}{\sqrt{n}} \qquad (5.4)$$

In the oil tests, three thermocouples were used per beaker. Using Equation 5.4 to propagate the ±0.5$^{o}$C thermocouple uncertainty results in a cumulative uncertainty of ±0.289 $^{o}$C.

For the bearing tests, oil temperature, shaft speed, motor torque, and vibrational data were the key factors. In this case, four thermocouples were used to calculate an average oil temperature. Using Equation 5.4 again results in an aggregate uncertainty of ±0.25$^{o}$C. In regards to shaft speed, the highest speed measured was 3600 rpm. The tachometer has an uncertainty of ±0.01%, which, at this speed, translates to ±0.36 rpm. The highest peak torque was always encountered during startup, but only lasted for a few seconds; this value was usually about 8 Nm. The largest steady state torque occurred during reverse spin at 2500 rpm with unheated degraded oil; this value was 5 Nm. Considering the torque meter's uncertainty of ±0.2%, the numerical uncertainty in the peak torque is ±0.016 Nm, and the numerical uncertainty in the maximum steady state torque is ±0.01 Nm. As for the vibrational data, there is a significant amount of uncertainty associated with the accelerometers (particularly accelerometer 3) in tests performed prior to

12/09/2019 for reasons previously discussed. Following that, the accelerometers have maximum amplitude (g) uncertainty of ±2% (for the frequency range considered). This is a relatively low amount of uncertainty, and further analysis is not necessary. The purpose of the accelerometers was to provide numerical evidence that a shaft-bearing interaction occurred, which would have been evident (had one occurred) regardless of the ±2% uncertainty.

Though there is also uncertainty associated with data processing (i.e. conversion from analog to digital signal), this is assumed to be negligibly small. Ultimately, in the case of both the oil degradation and bearing tests, all associated uncertainties (with the exception of the vibrational data from the bearing tests) are assumed inconsequential when compared to the magnitude of the parameters they represent.

# 6. CONCLUSION

## 6.1. Summary

Tests were conducted to evaluate the change in the composition of oil as it thermally degrades over the course of 72 hours, and to investigate the effect thermally degraded oil had on the performance of a ZS-1 Terry turbine.

In regards to the oil tests, a color changes did occur following the degradations, but this can occur at very low levels of oil degradation and does not indicate a significant chemical change. The viscosities before and after the degradations (accounting for convection currents and the effect of glass vs steel beakers) were identical.

As for the bearing tests, two types of experiments were conducted. In the bearing stress tests, there was no damage to the bearings following 3-hour tests with fresh oil (both heated and unheated), as expected. Additionally, a 1000 rpm, unheated, degraded oil test was also performed with no bearing wear. Following the 3000 rpm, unheated, degraded oil test, however, very slight scoring was detected on the bottom half of both bearings. The damage to the bronze ZS-1 bearings was superficial, but could have critical implications into the effect of degraded oil on the softer Babbitt bearings of the industry size GS Terry turbine; further damage did not appear in subsequent tests. In the steady state torque tests, the relationship between motor torque and shaft speed under different test conditions was explored. First, the direction of spin had the strongest effect on torque, with reverse spin requiring more torque. Next, oil temperature had the second largest impact on torque, with the unheated condition resulting in the higher value. Finally, oil quality had the least influence on torque, and the effect was only noticeable during forward spin, where degraded oil resulted in an overall higher torque.

Ultimately, the change in oil composition over the course of a 72-hour 121ºC degradation was found to be negligibly small. Additionally, when the same degraded oil was used as bearing lubricant in a ZS-1 Terry turbine, the effect on torque was noticeable, but minor. Scoring to the bearings did occur when degraded oil was used, but the degree of damage was superficial.

### 6.2. Future Work

The next step in this research will be the full-scale tests with the GS Terry turbine. This investigation will put greater emphasis on oil temperature than oil quality. Therefore, degraded oil will not be used, but a wider range of oil temperatures will be explored. Additionally, these tests will collect temperature readings from inside the bearings themselves, not just the oil.

Following the completion of the full-scale tests with the GS Terry turbine, the ZS Terry turbine testing facility will be reconstructed. At that time, the tests conducted for this thesis will be replicated to verify repeatability.

Furthermore, the complete proposed project test matrix for oil testing (Table 4.1) will be carried out by a Texas A&M petroleum engineering oil analysis lab. These tests will all be 72 hours in length, and will include the aerated and wet tests; initial and final viscosity and nuclear magnetic resonance data will be collected.

# REFERENCES

1. “Terry Steam Turbine Engine”. *New England Wireless & Steam Museum*. 31 Dec 2015, newsm.org/steam/terry-steam-turbine/. Accessed 8 Oct 2019
2. Osborn, Douglas, et al. “Terry Turbopump Expanded Operating Band Research”. *Light Water Reactor Sustainability Newsletter*. U.S. Department of Energy, 2018.
3. “Reactor Core Isolation Cooling (RCIC) System”. *United States Nuclear Regulatory Commission*. 5 Mar 2019. nrcoe.inl.gov/resultsdb/SysStudy/RCIC.aspx. Accessed 24 Mar 2019.
4. “Fukushima Daiichi Accident”. *World Nuclear Association*. Oct 2018. www.world-nuclear.org/information-library/safety-and-security/safety-of-plants/fukushima-accident.aspx. Accessed 4 Mar 2019.
5. Osborn, Douglas, et al. *Terry Turbopump Expanded Operating Band*. SAND2017-4246C. Sandia National Laboratories, 2017.
6. Peck, Diana. *Turbine Oil Degradation and its Effects on Performance of a Terry Turbine in Nuclear Applications*. MS Thesis. Texas A&M University, 2019.
7. Majumdar, B C. *Introduction to Tribology of Bearings*. New Delhi, S. Chand, 2008.
8. Wilkes, Jason, et al. *Compression Machinery for Oil and Gas*. Gilf Professional Publishing, 2019.
9. Abdurahman , H. Nour and Rosli, M. Yunus. “A Continuous Microwave Heating of Water-in-Oil Emulsions: An Experimental Study”. *Journal of Applied Sciences*, vol. 6, iss. 8, pp. 1868-1872. https://scialert.net/fulltextmobile/?doi=jas.2006.1868.1872. Accesed 6 Feb 2020.

10. “Stability of Oil Emulsions”. *PetroWiki*. 22 Oct 2014, https://petrowiki.org/Stability_of_oil_emulsions. Accessed 6 Feb 2020.

11. Harika, Elias, et al. “Measurements of Lubrication Characteristics of a Tilting Pad Thrust Bearing Disturbed by a Water-Contaminated Lubricant”. *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, vol. 227, pg 16-25, 2013.

12. Harika, Elias, et al. “Effects of Water Contamination of Lubricants on Hydrodynamic Lubrication: Rheological and Thermal Modeling”. *Journal of Tribology*, vol. 135, 2013.

# APPENDIX A

# OIL DEGRADATION PLOTS

## A.1. 121ºC Oil Degradation - 2/18/2019

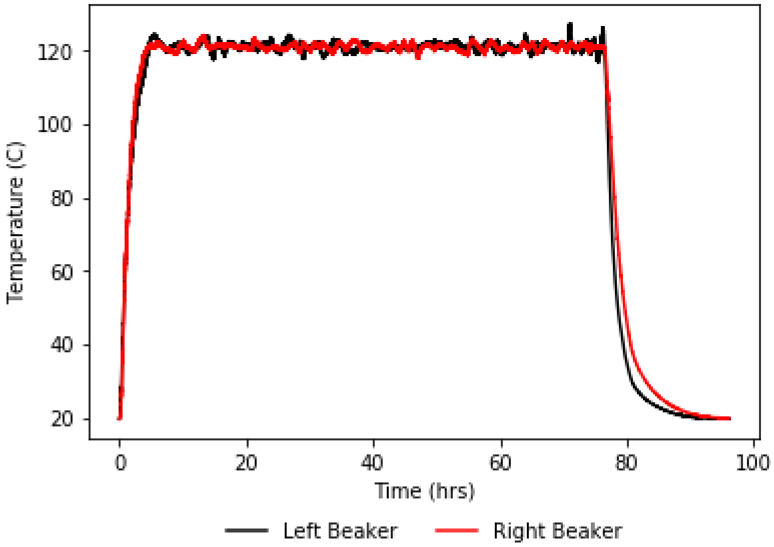


**Figure A-1:** Average oil temperature over time for the initial 121ºC oil degradation.

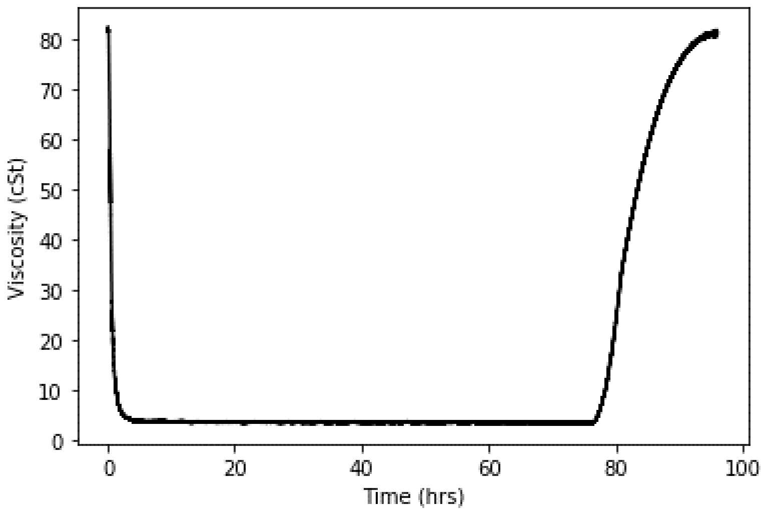


**Figure A-2:** Viscosity over time in the right beaker, for the initial 121°C oil degradation.

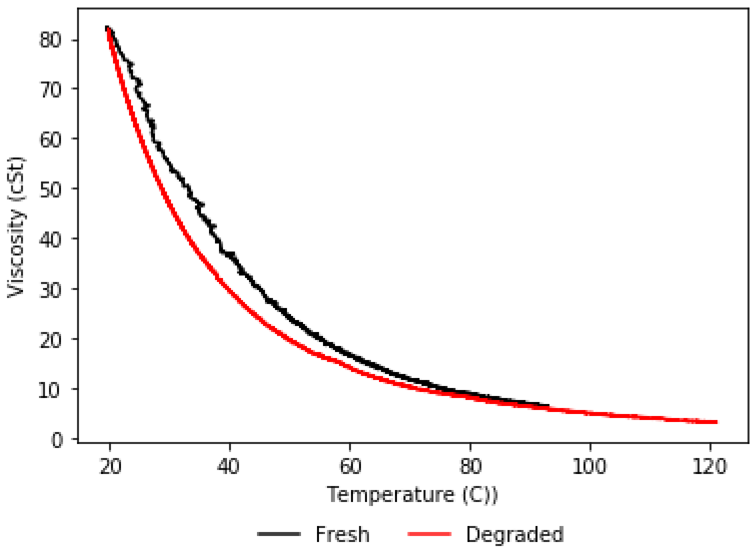


**Figure A-3:** This plot shows the relationship between viscosity and temperature before and after the degradation, for the initial 121°C oil degradation.

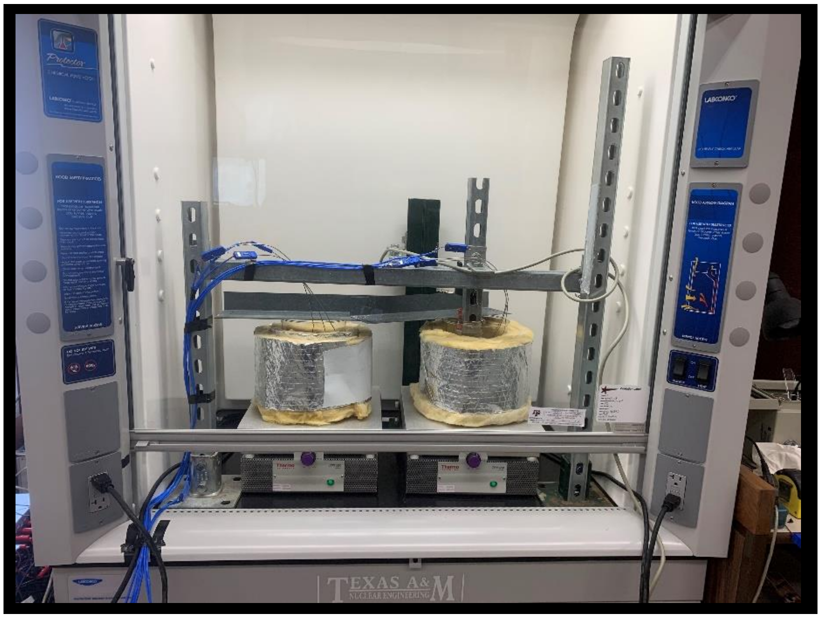


**Figure A-4:** Front view of the experimental setup for the initial 121$^{o}$C oil degradation.

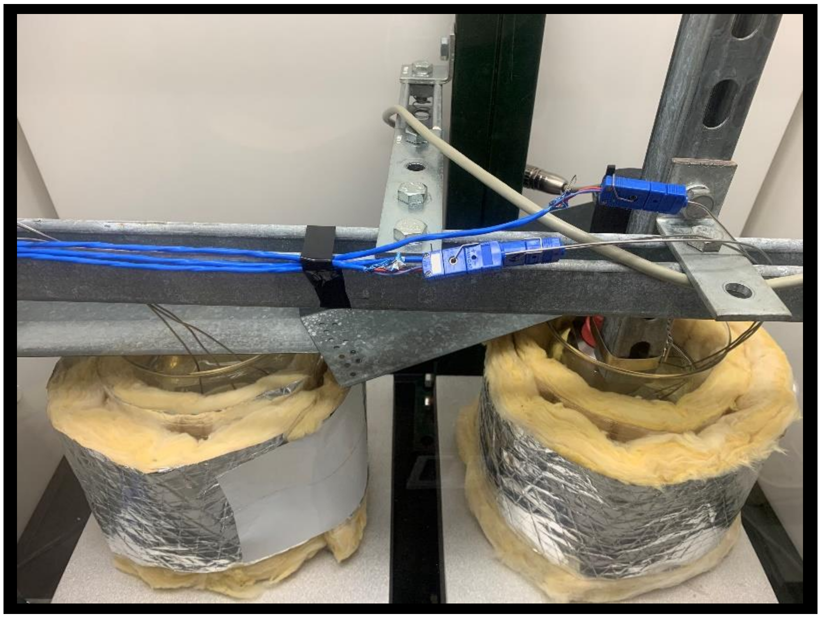


**Figure A-5:** Top view of the experimental setup for the initial 121$^{o}$C oil degradation.

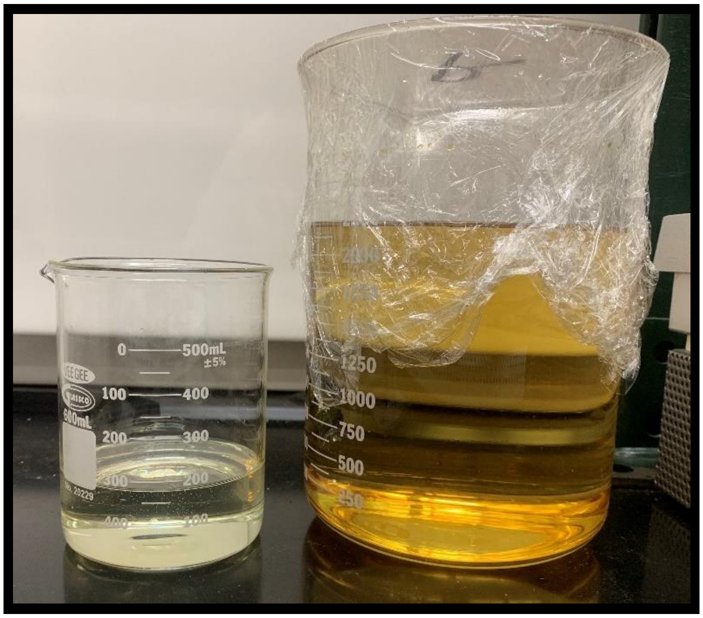


**Figure A-6:** Comparison between oil prior to (left) and after (right) the initial 121$^o$C oil degradation.

### A.2. 93$^o$C 10% Wetness Oil Degradation - 3/7/2019

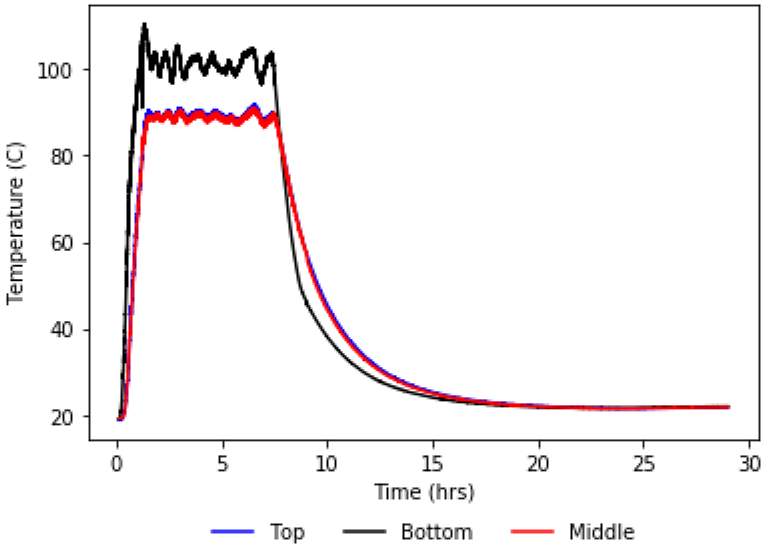


**Figure A-7:** Average temperature over time for the 93$^o$C 10% wet oil degradation.

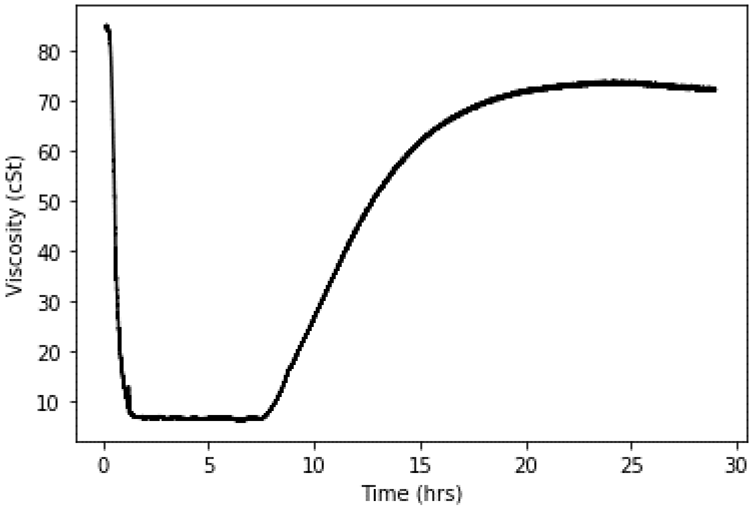


**Figure A-8:** Viscosity over time for the 93$^o$C 10% wet oil degradation.

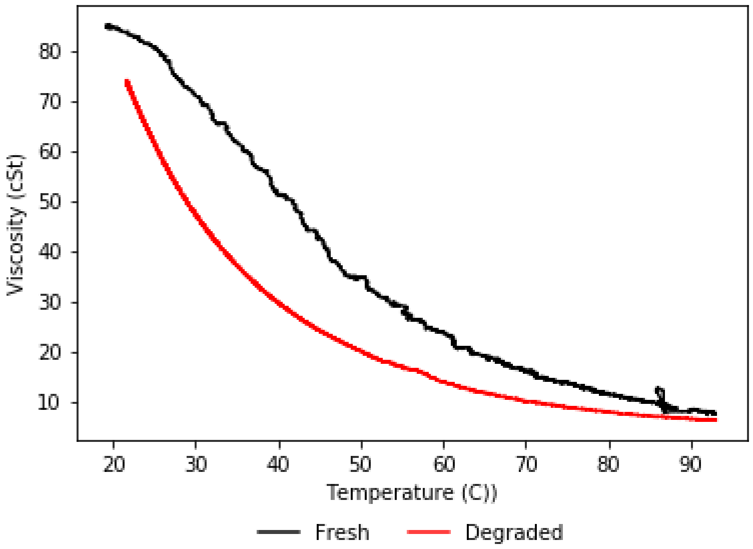


**Figure A-9:** This plot shows the relationship between viscosity and temperature before and after the degradation, for the 93ºC 10% wet oil degradation.

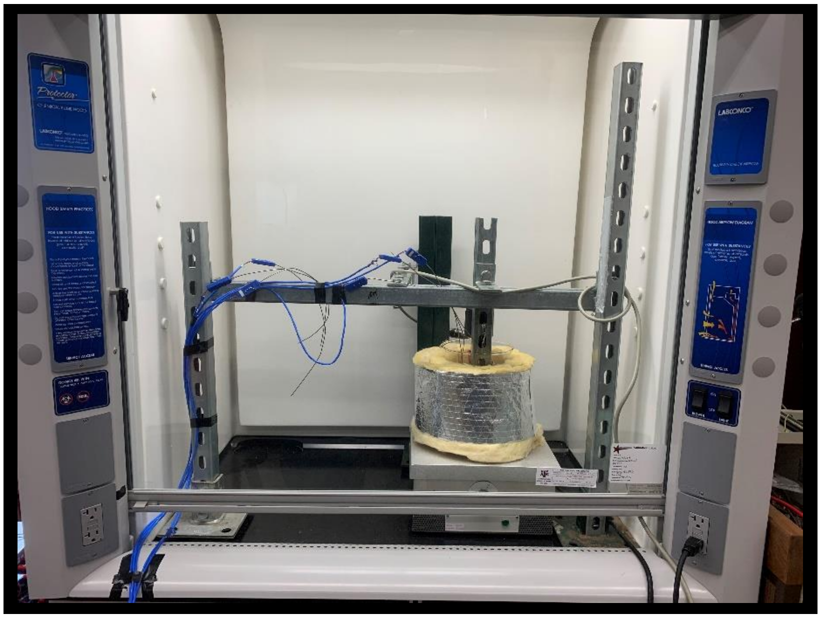


**Figure A-10:** Front view of the experimental setup for the 93ºC 10% wet oil degradation.

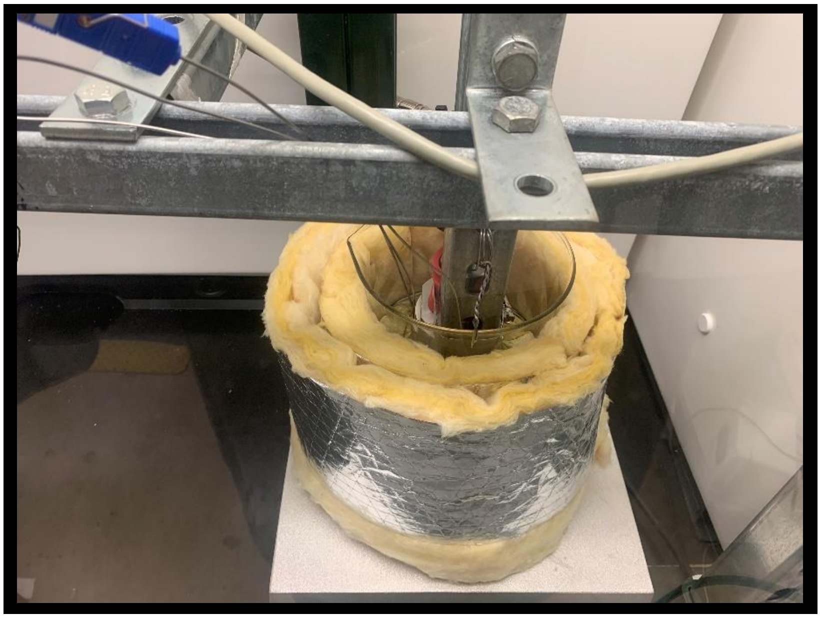


**Figure A-11:** Oil at the start of the 93$^{o}$C 10% wet oil degradation.

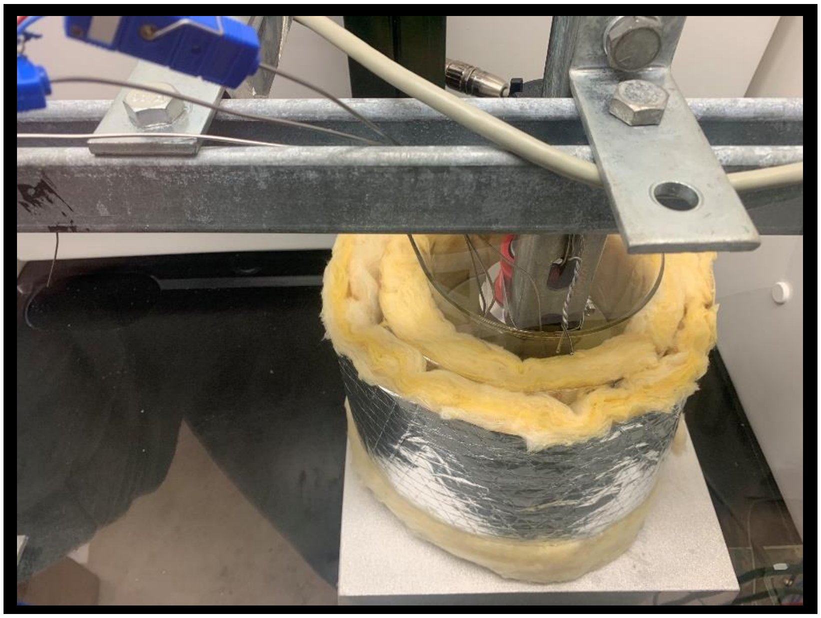


**Figure A-12:** Oil at the 1:08:00 timestamp for the 93$^{o}$C 10% wet oil degradation.

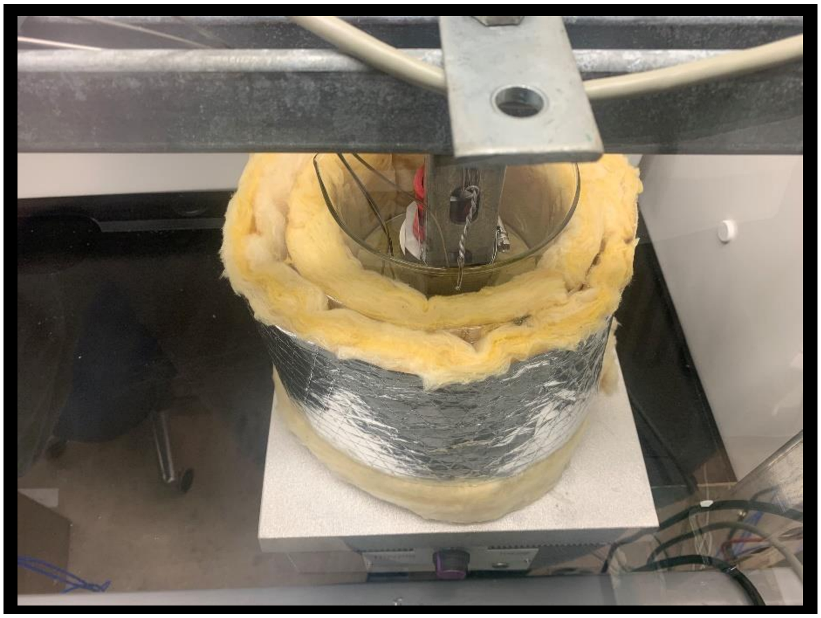


**Figure A-13:** Oil at the 2:19:00 timestamp for the 93$^{o}$C 10% wet oil degradation.

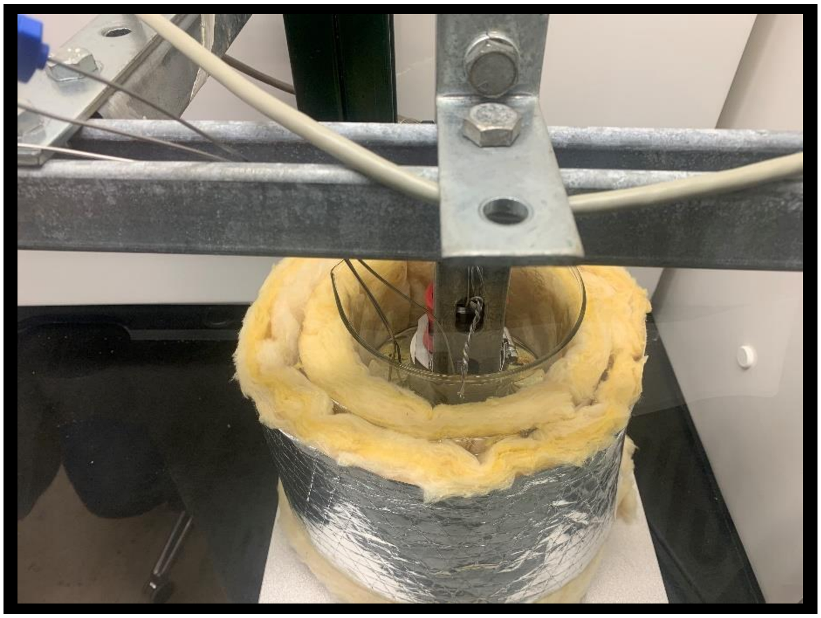


**Figure A-14:** Oil at the 4:06:00 timestamp for the 93$^{o}$C 10% wet oil degradation.

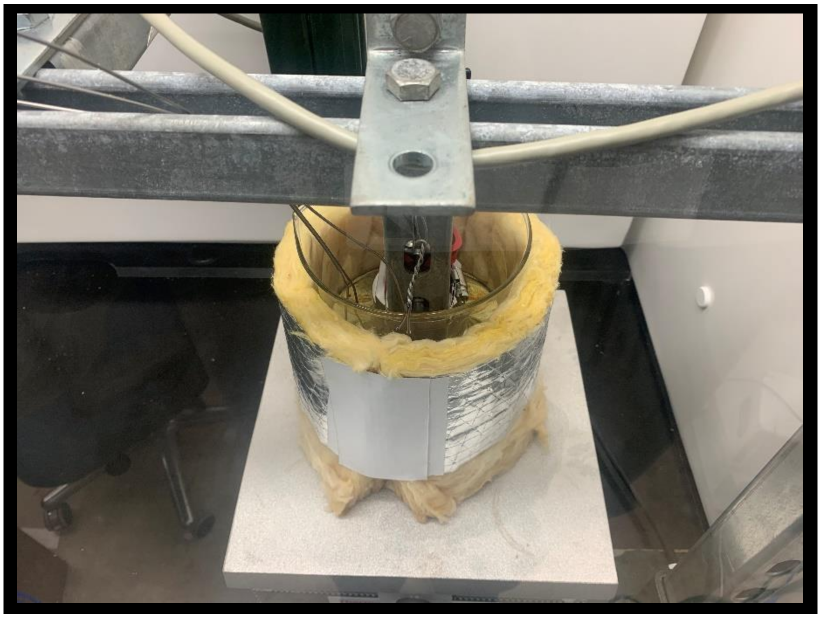


**Figure A-15:** Oil at the 6:18:00 timestamp for the 93$^{o}$C 10% wet oil degradation.

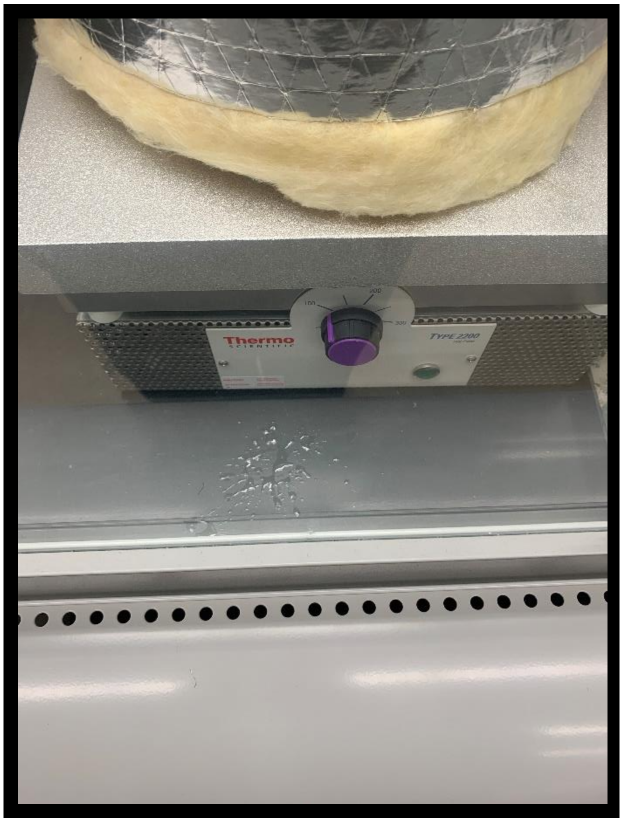


**Figure A-16:** Aftermath of the flashing event that occurred during the 93$^{o}$C 10% wet oil degradation.

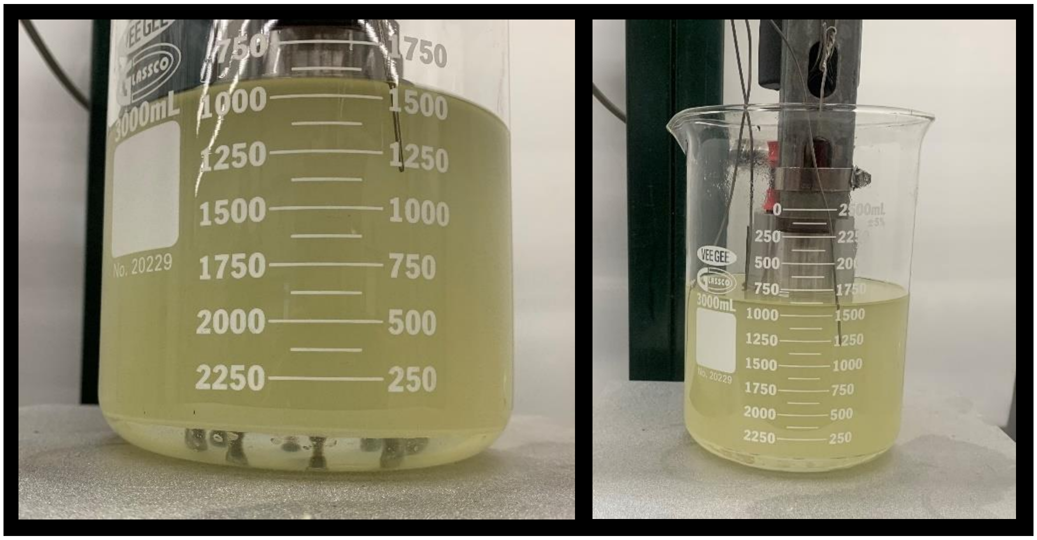


**Figure A-17:** Stratified oil/water mixture dirtectly after the 93$^{o}$C 10% wet oil degradation.

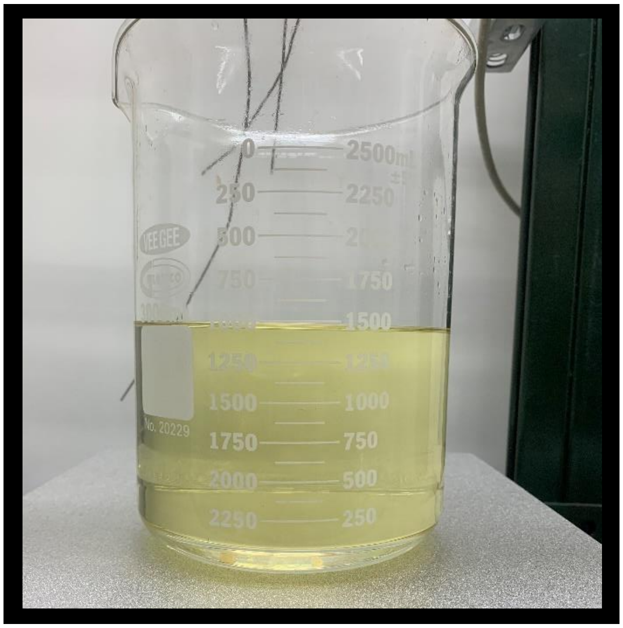


**Figure A-18:** Stratified oil/water mixture one week after the 93$^{o}$C 10% wet oil degradation.

### A.3. 121°C Oil Degradation - 7/8/2019

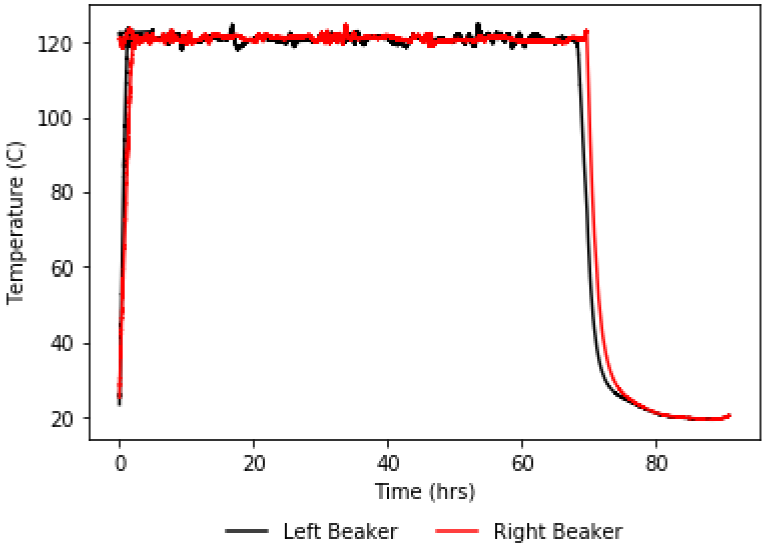


**Figure A-19:** Average oil temperature over time for the second 121°C oil degradation.

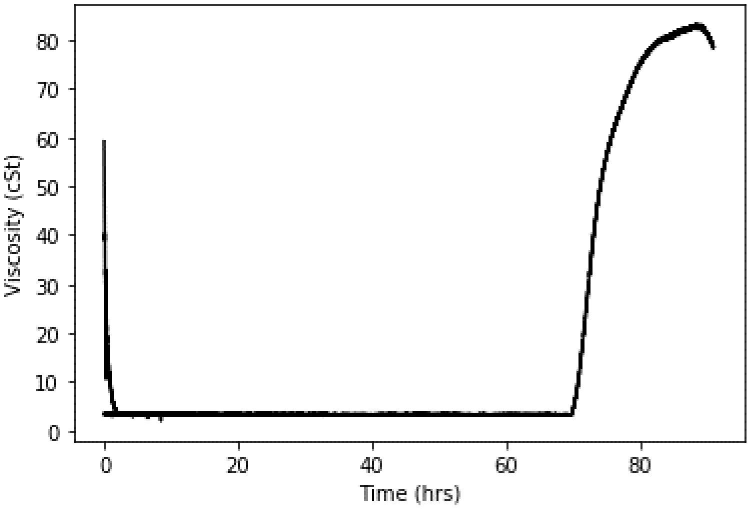


**Figure A-20:** Viscosity over time in the right beaker, for the second 121°C oil degradation.

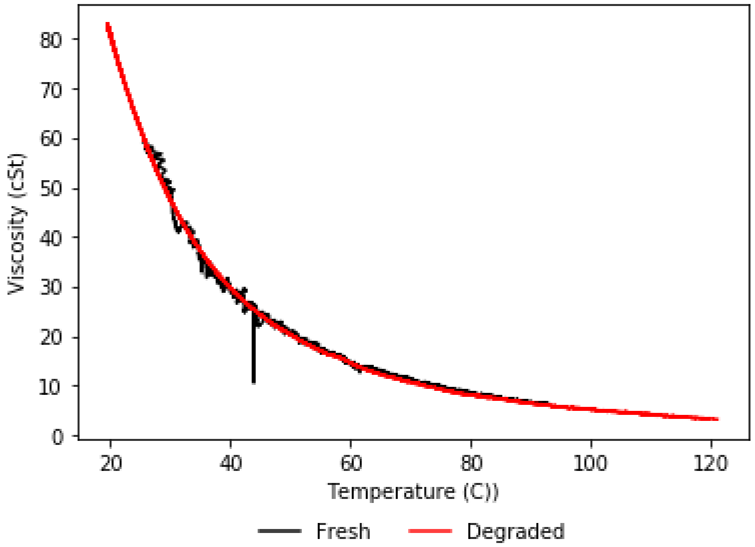


**Figure A-21:** This plot shows the relationship between viscosity and temperature before and after the degradation, for the second 121ºC oil degradation.

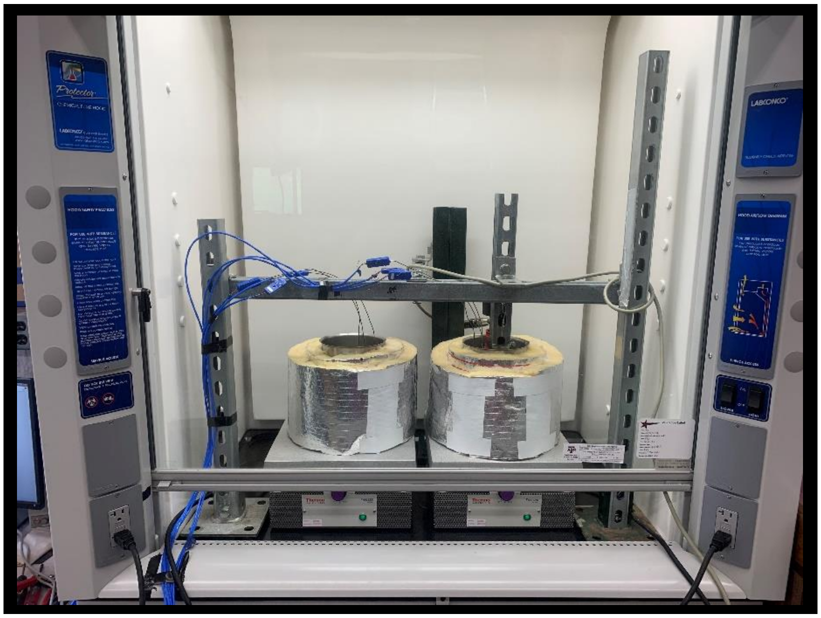


**Figure A-22:** Front view of the experimental setup for the second 121ºC oil degradation.

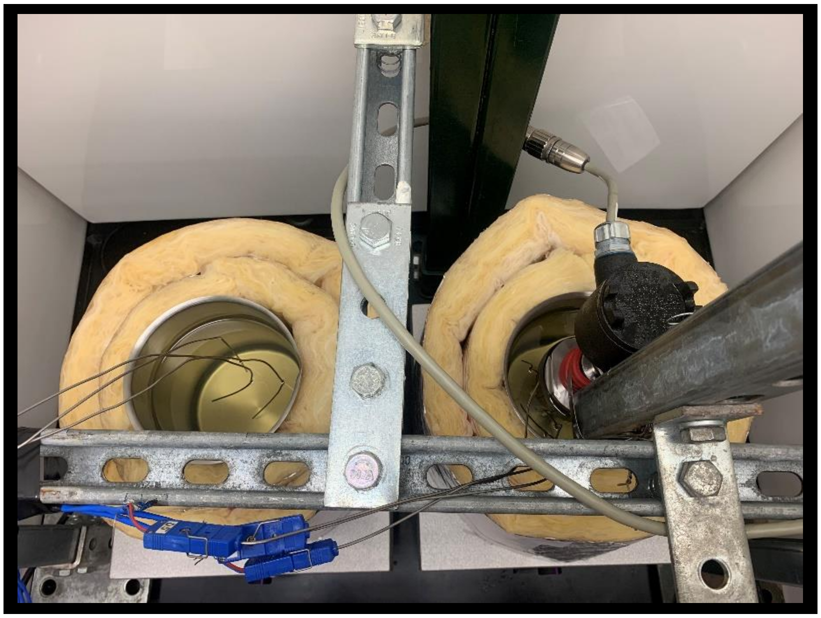


**Figure A-23:** Oil at the start of the second 121ºC oil degradation.

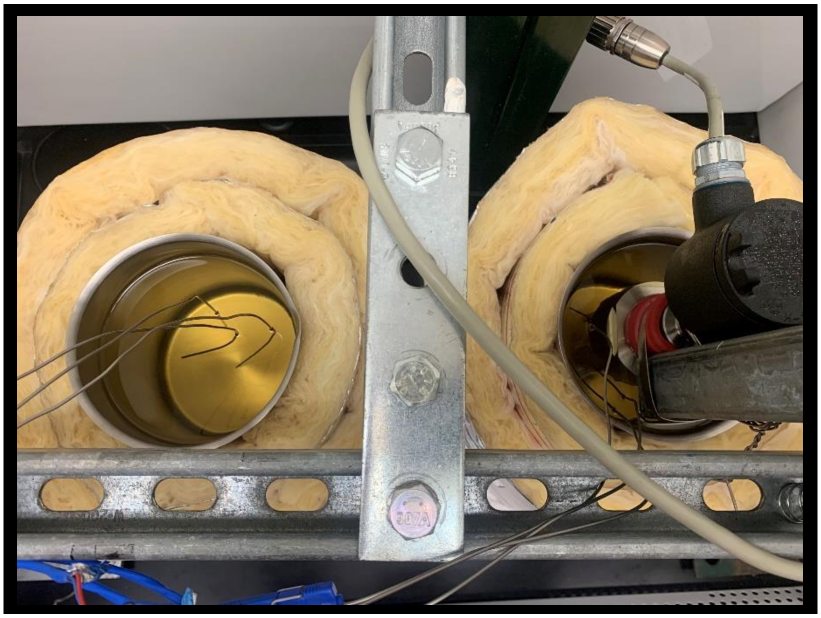


**Figure A-24:** Oil at the 13:15:00 timestamp for the second 121ºC oil degradation.

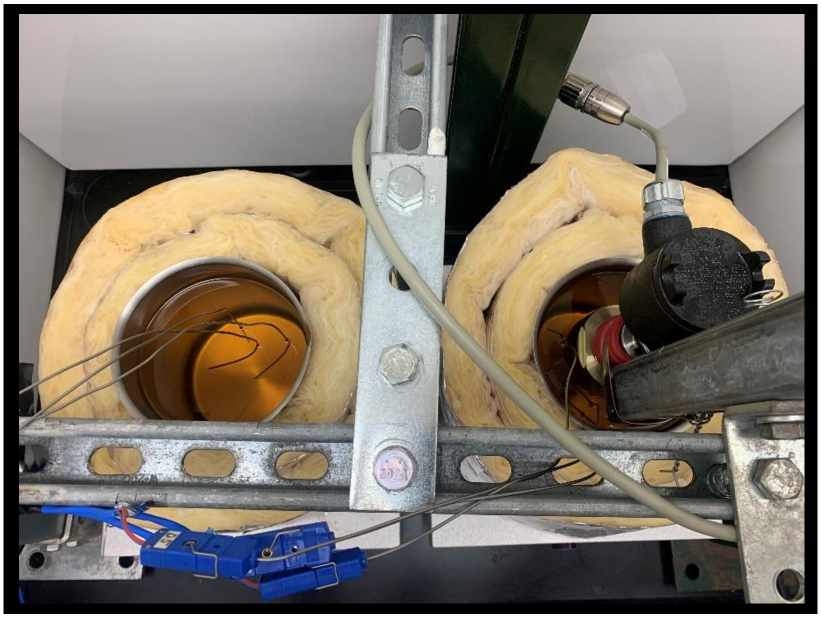


**Figure A-25:** Oil at the 34:35:00 timestamp for the second 121ºC oil degradation.

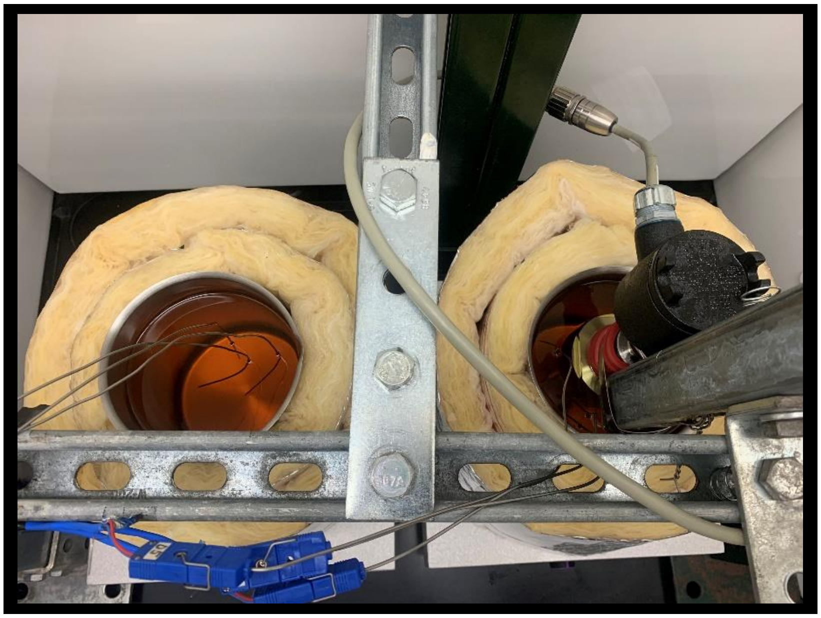


**Figure A-26:** Oil at the 58:37:00 timestamp for the second 121ºC oil degradation.

# APPENDIX B

# BEARING TEST PLOTS

## B.1. Shakedown

### *B.1.1. Multiple Speed Tests - 2/22/2019*

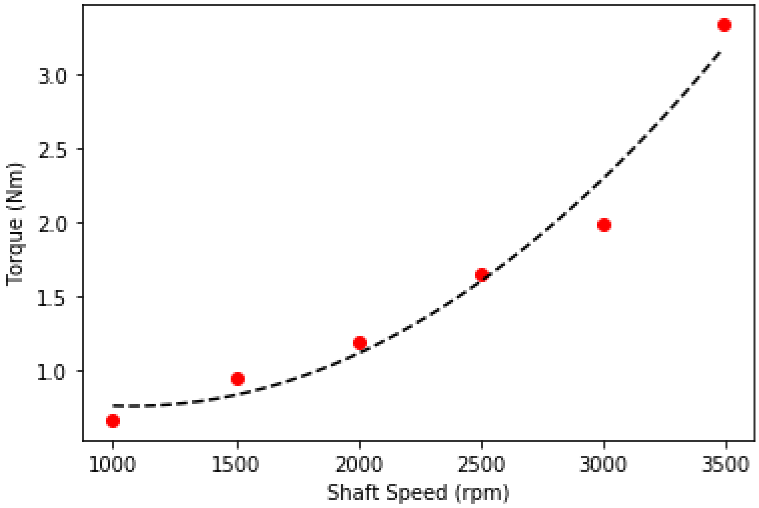


**Figure B-1:** Relationship between speed and torque for forward spin only, for the multiple speed shakedown. The equation for this line is 4.144e-7x$^2$ – 8.894e-4x + 1.236, with an R$^2$ of 0.968.

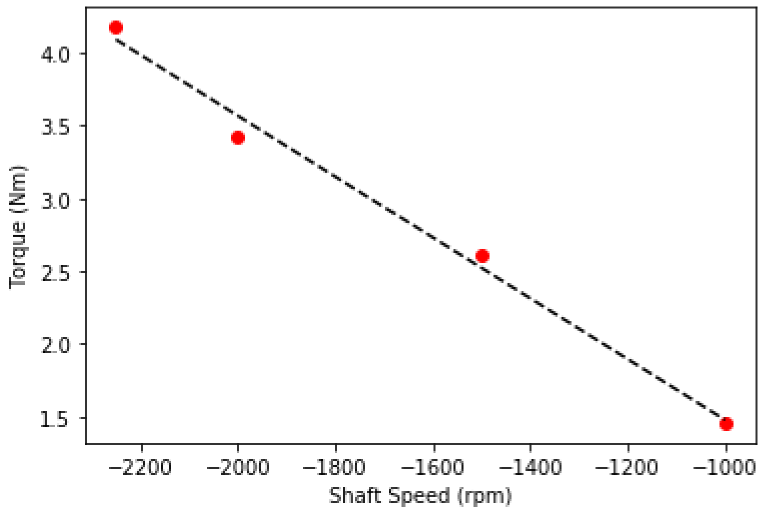


**Figure B-2:** Relationship between speed and torque for reverse spin only, for the multiple speed shakedown. The equation for this line is -2.089e-3x – 0.611, with an $R^2$ of 0.991.

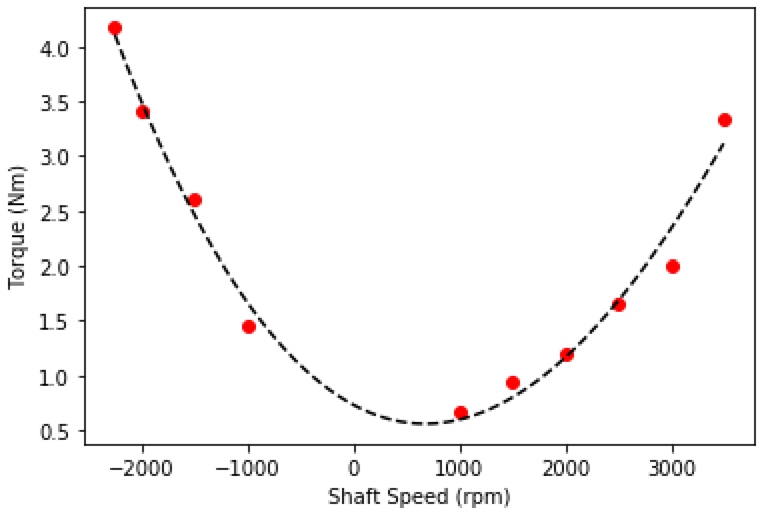


**Figure B-3:** Full curve relationship between speed and torque, for the multiple speed shakedown. The equation for this line is -1.61e-11$x^3$ + 4.006e-7$x^2$ – 5.189e-4x + 0.731, with an $R^2$ of

### *B.1.2. 6 Hour Shakedown - 4/3/2019*

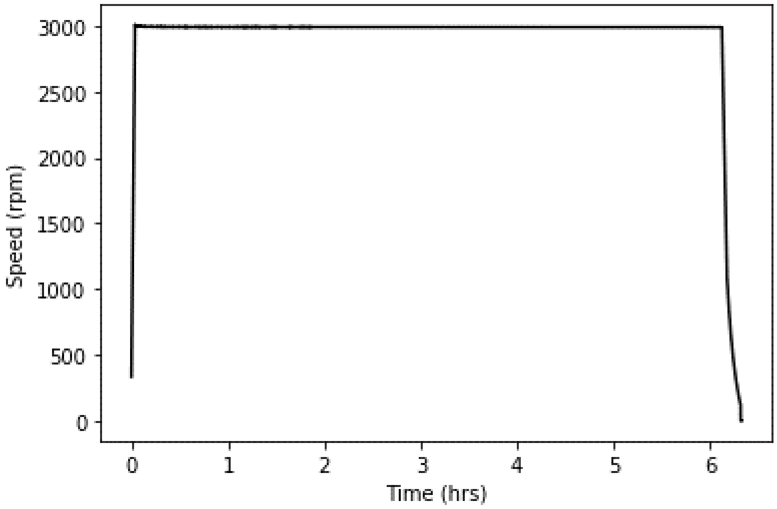


**Figure B-4:** Rotational shaft speed over time for the 6 hour shakedown.

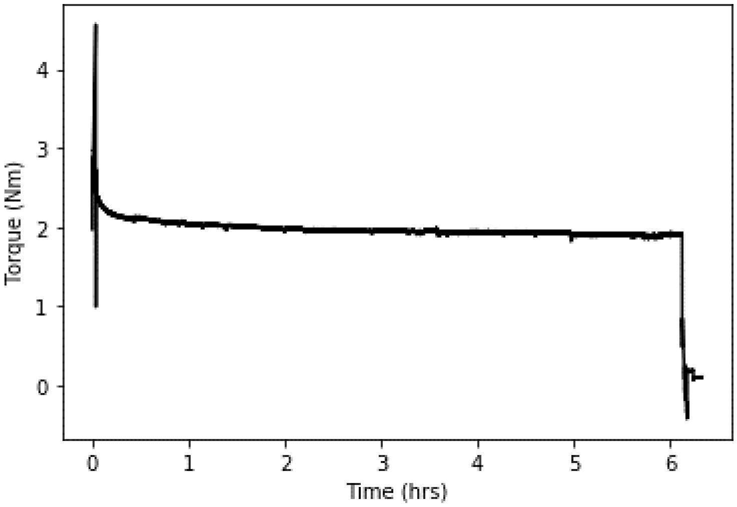


**Figure B-5:** Motor torque over time for the 6 hour shakedown.

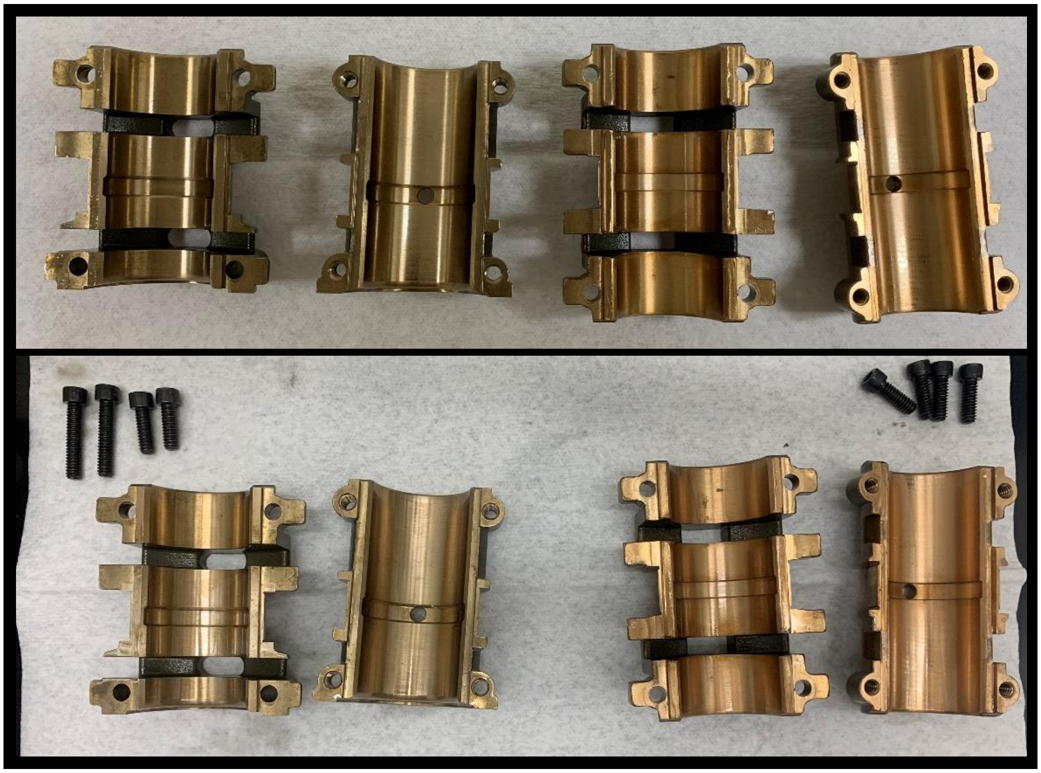


**Figure B-6:** ZS1 journal bearings before (top) and after (bottom) testing, for the 6 hour shakedown.

### *B.1.3. Heater Tape Shakedown - 7/26/2019*

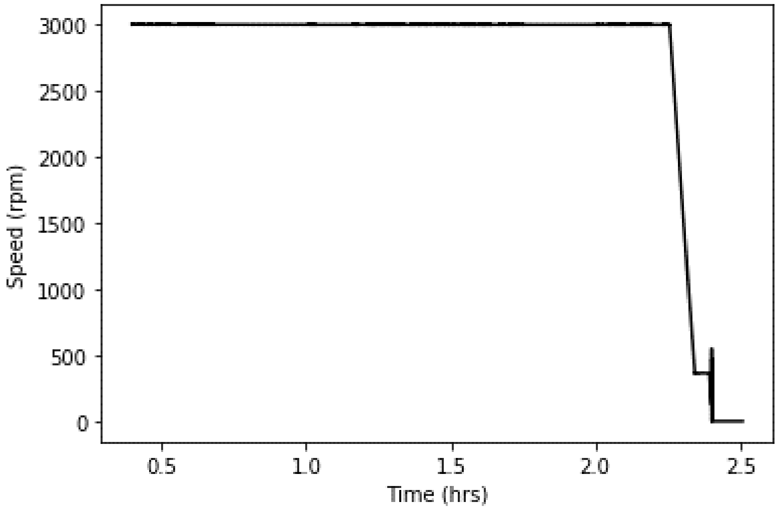


**Figure B-7:** Rotational shaft speed over time for the heater tape shakedown.

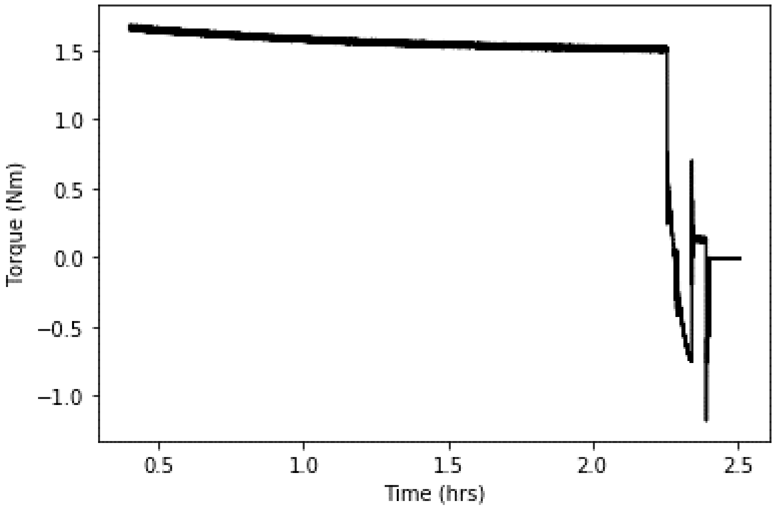


**Figure B-8:** Motor torque over time for the heater tape shakedown.

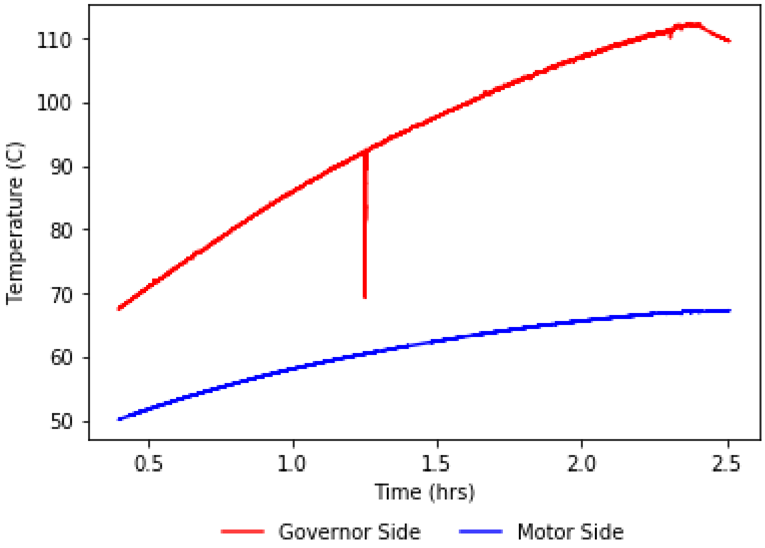


**Figure B-9:** Average oil temperature over time for the heater tape shakedown.

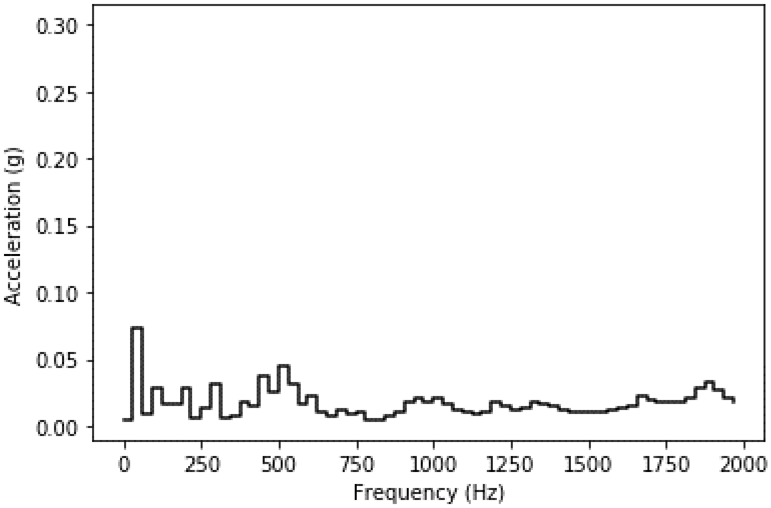


**Figure B-10:** Average acceleration magnitude per frequency bin for accelerometer 1, for the heater tape shakedown.

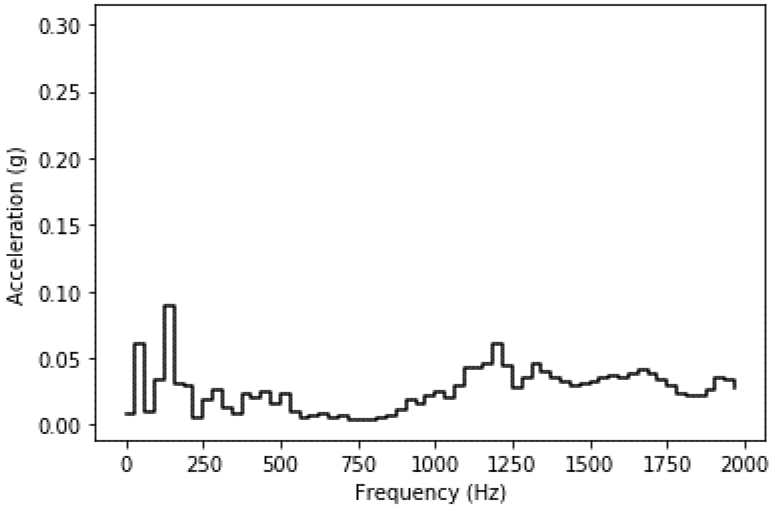


**Figure B-11:** Average acceleration magnitude per frequency bin for accelerometer 2, for the heater tape shakedown.

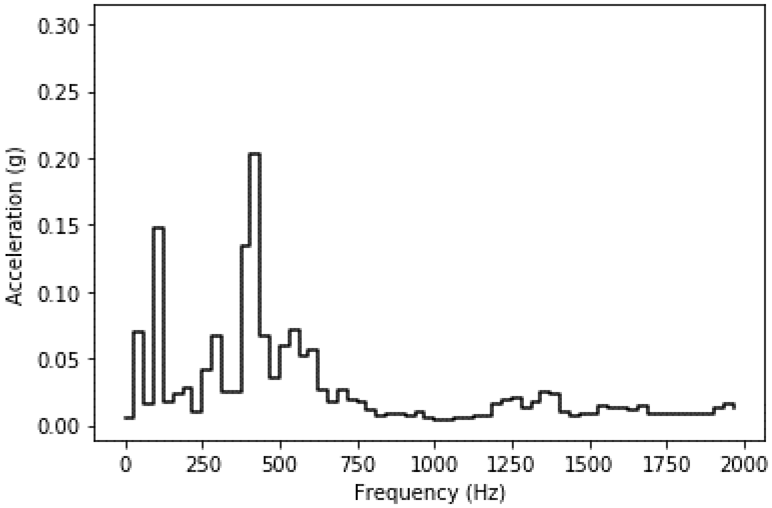


**Figure B-12:** Average acceleration magnitude per frequency bin for accelerometer 3, for the heater tape shakedown.

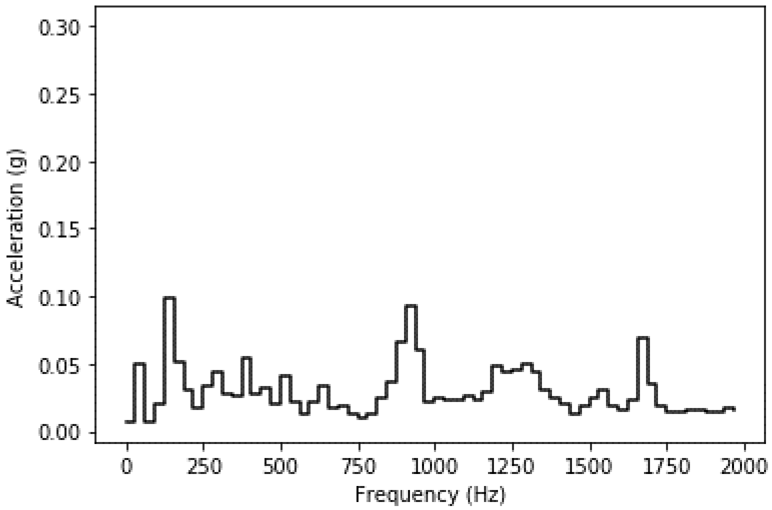


**Figure B-13:** Average acceleration magnitude per frequency bin for accelerometer 4, for the heater tape shakedown.

### *B.1.4. Immersion Heater Calibration - 10/21/2019*

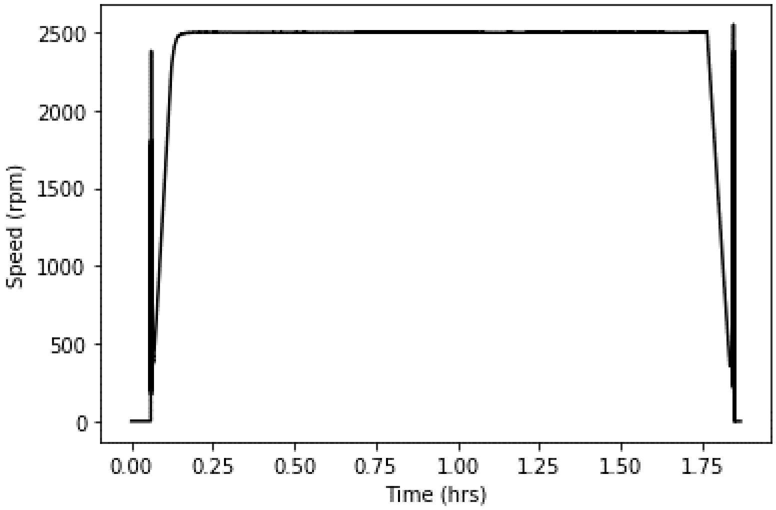


**Figure B-14:** Rotational shaft speed over time for the immersion heater calibration.

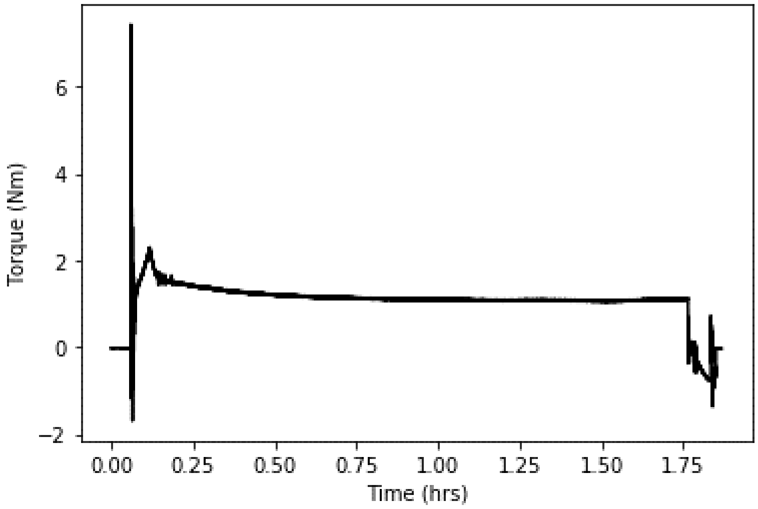


**Figure B-15:** Motor torque over time for the immersion heater calibration.

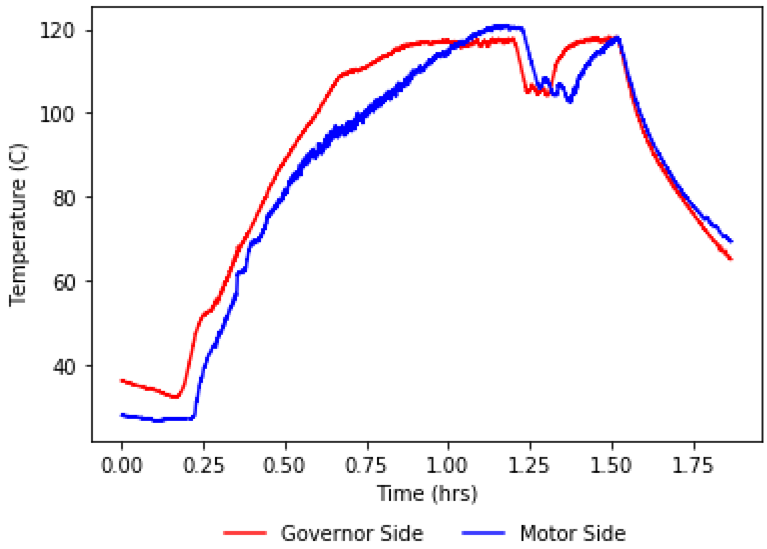


**Figure B-16:** Average oil temperature over time for the immersion heater calibration.

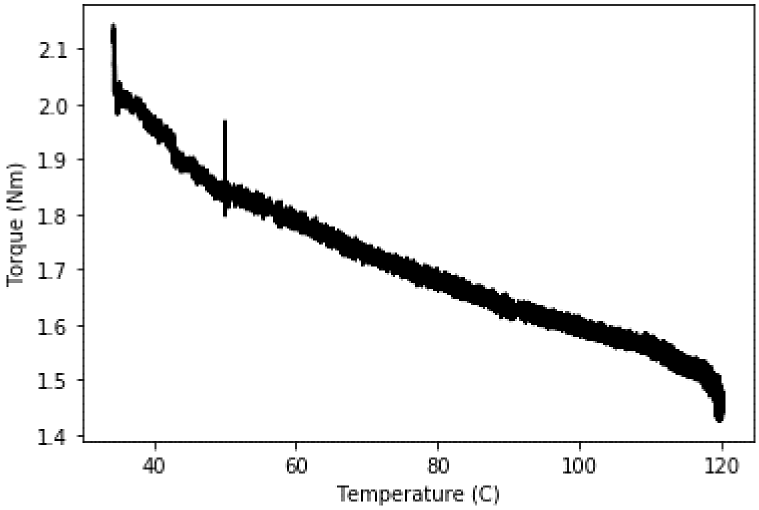


**Figure B-17:** Relationship between motor torque and oil temperature for the immersion heater calibration.

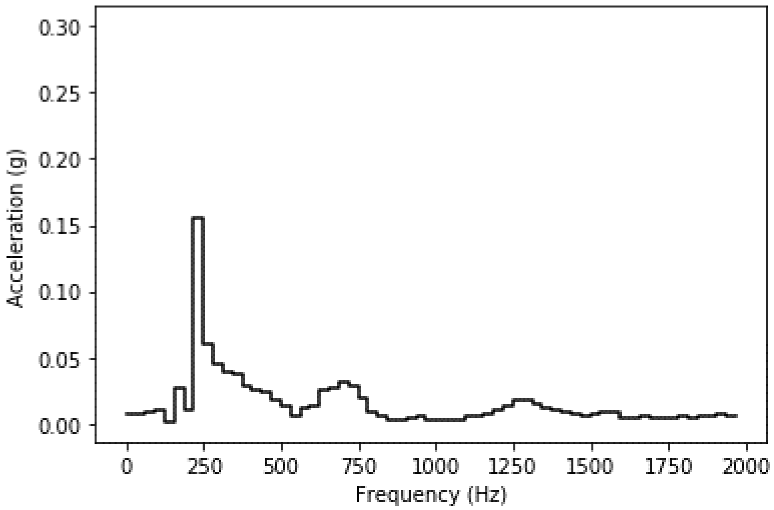


**Figure B-18:** Average acceleration magnitude per frequency bin for accelerometer 1, for the immersion heater calibration.

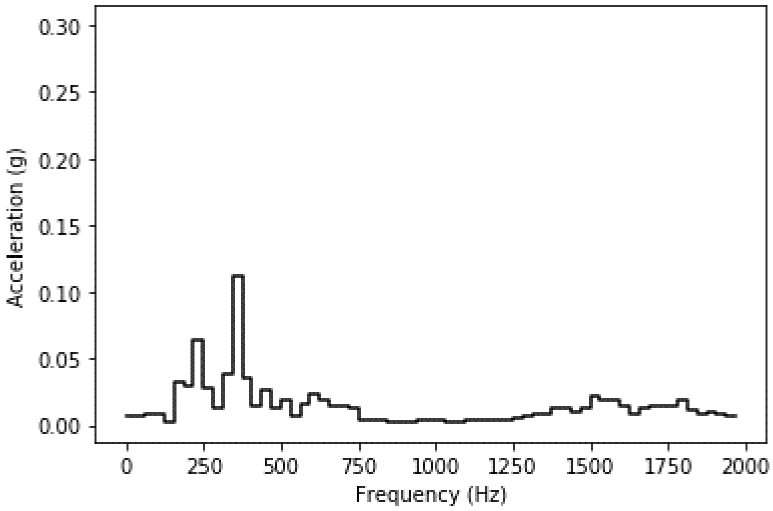


**Figure B-19:** Average acceleration magnitude per frequency bin for accelerometer 2, for the immersion heater calibration.

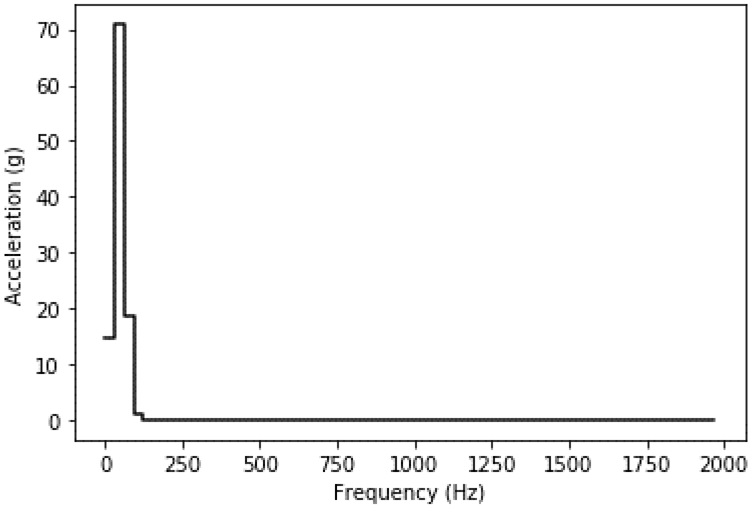


**Figure B-20:** Average acceleration magnitude per frequency bin for accelerometer 3, for the immersion heater calibration.

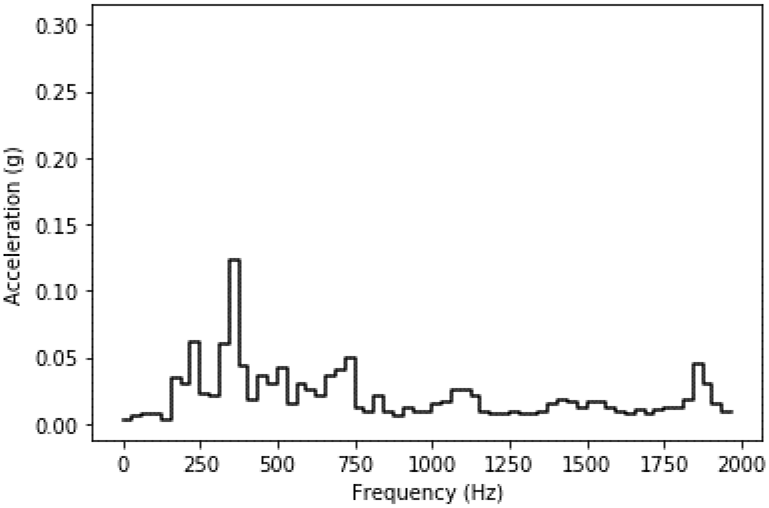


**Figure B-21:** Average acceleration magnitude per frequency bin for accelerometer 4, for the immersion heater calibration.

### *B.1.5. Immersion Heater Shakedown - 10/24/2019*

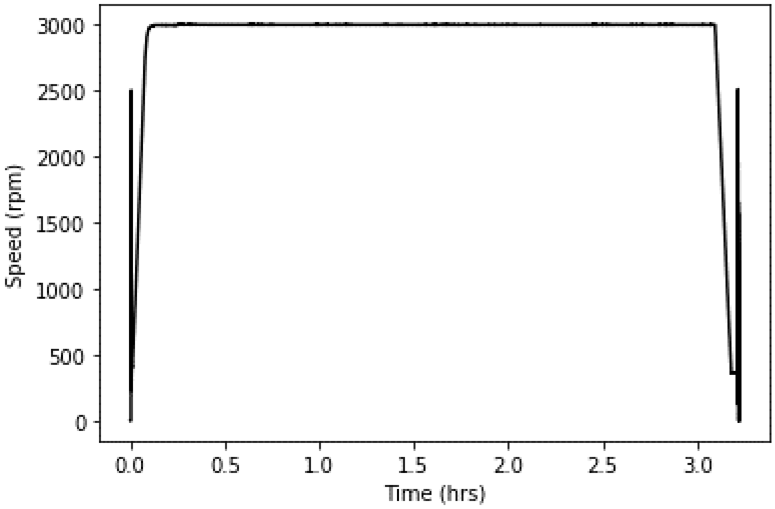


**Figure B-22:** Rotational shaft speed over time for the immersion heater shakedown.

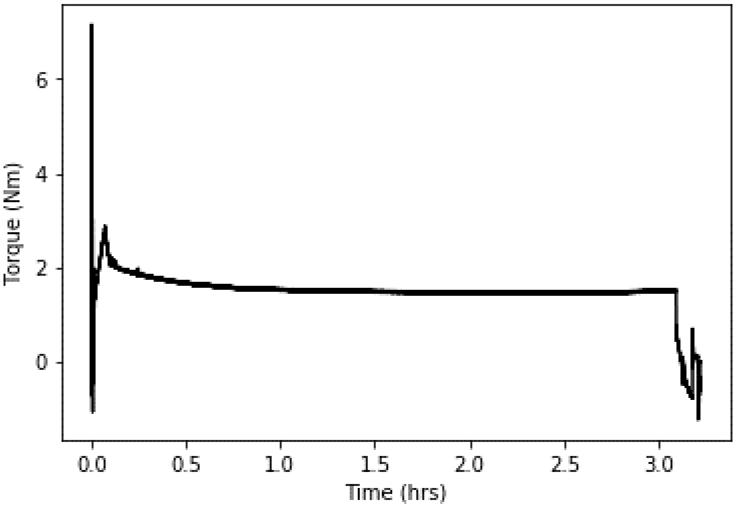


**Figure B-23:** Motor torque over time for the immersion heater shakedown.

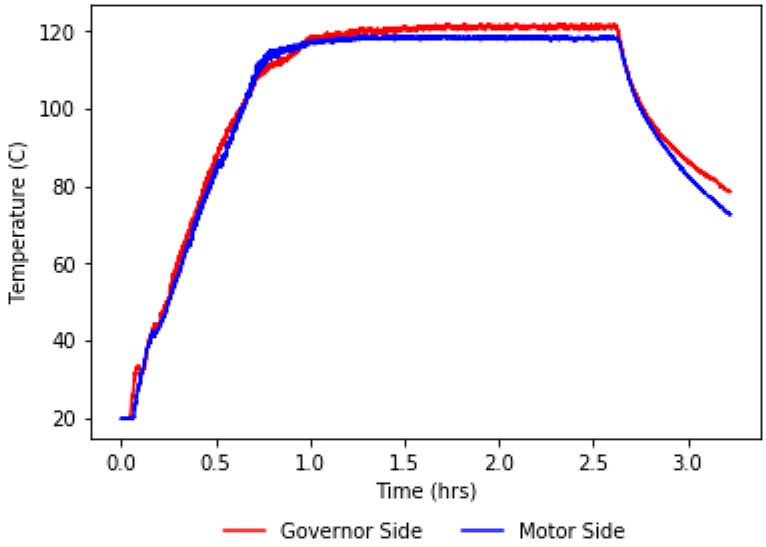


**Figure B-24:** Average oil temperature over time for the immersion heater shakedown.

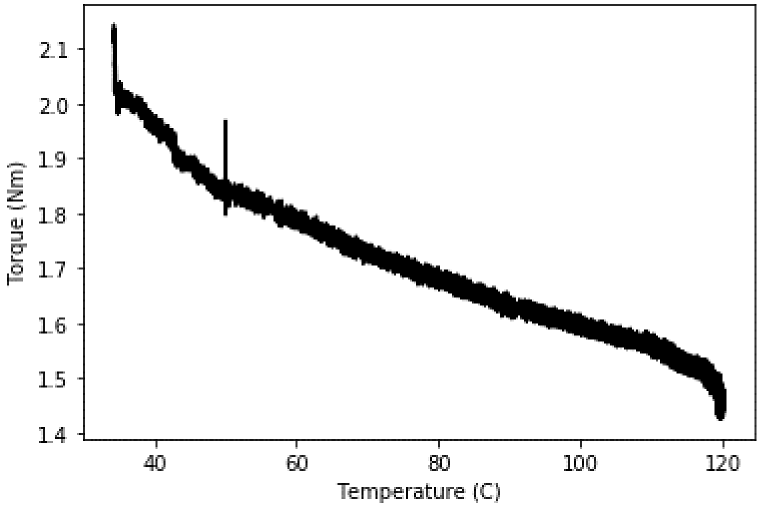


**Figure B-25:** Relationship between motor torque and oil temperature for the immersion heater shakedown.

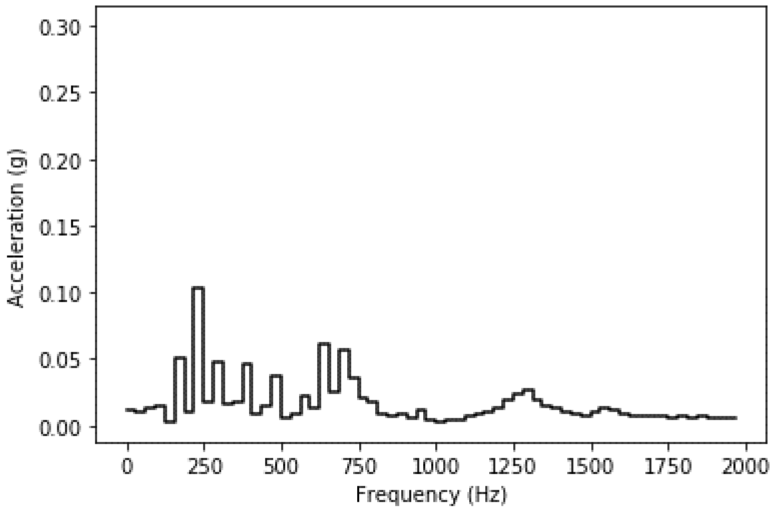


**Figure B-26:** Average acceleration magnitude per frequency bin for accelerometer 1, for the immersion heater shakedown.

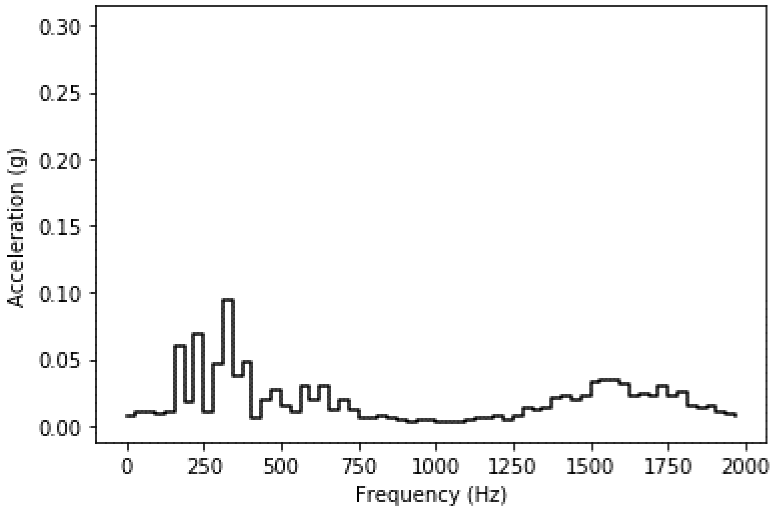


**Figure B-27:** Average acceleration magnitude per frequency bin for accelerometer 2, for the immersion heater shakedown.

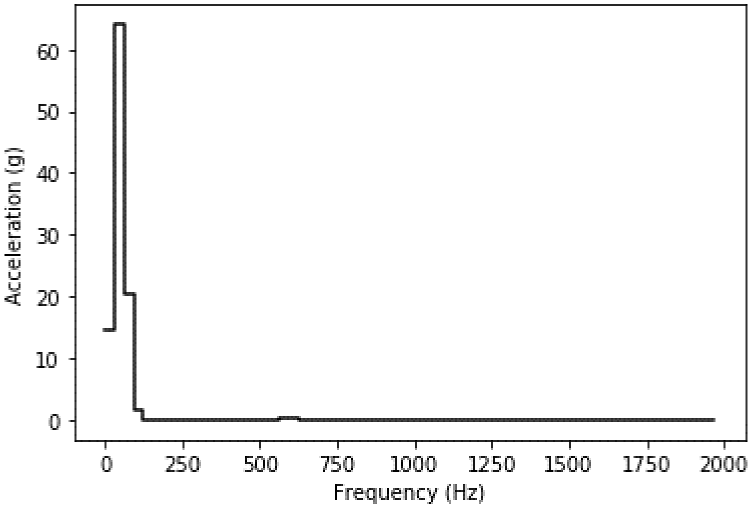


**Figure B-28:** Average acceleration magnitude per frequency bin for accelerometer 3, for the immersion heater shakedown.

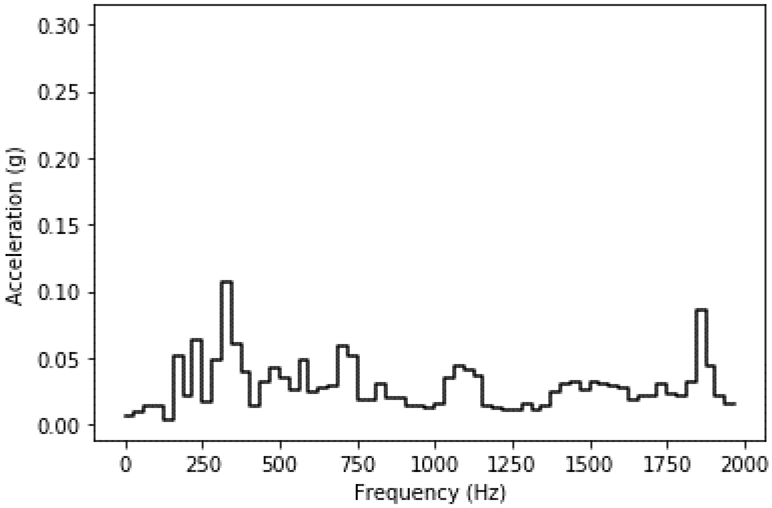


**Figure B-29:** Average acceleration magnitude per frequency bin for accelerometer 4, for the immersion heater shakedown.

## B.2. Tests

### *B.2.1. Bearing Stress Tests*

*For these tests, it was assumed that the bearing state did not change between tests. Therefore, before and after beating photos were only taken for the first test; subsequent tests only have after photos. Additionally, it was discovered that some of the accelerometer wire (primarily those connected to accelerometer 3) had suffered damage and were providing faulty readings. All of the accelerometer wiring was replaced on 12/09/2019.*

#### B.2.1.1. Fresh Oil, No Heat, 1000 rpm - 11/8/2019

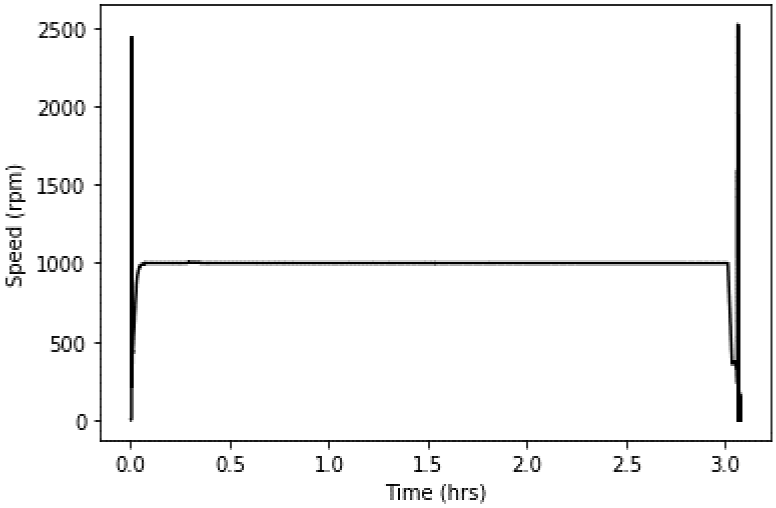


**Figure B-30:** Rotational shaft speed over time for the unheated, fresh oil, 1000 rpm test.

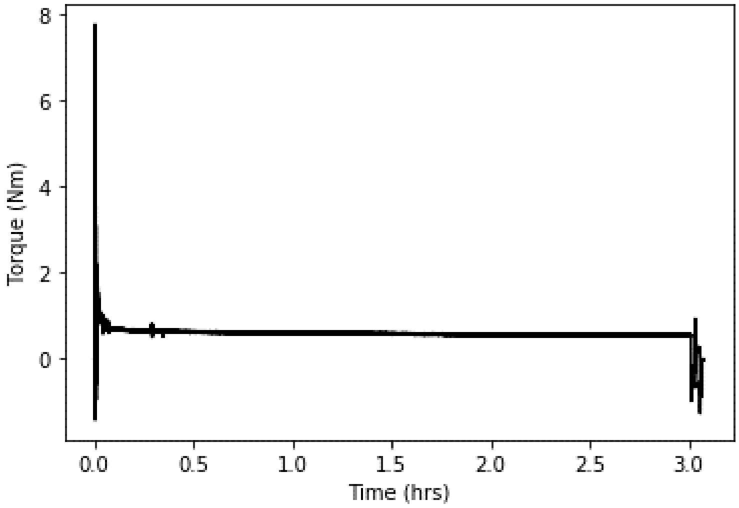


**Figure B-31:** Motor torque over time for the unheated, fresh oil, 1000 rpm test.

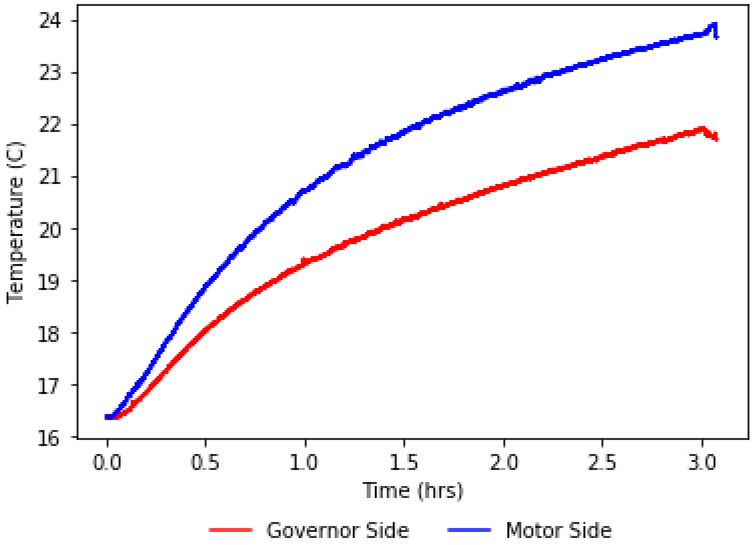


**Figure B-32:** Average oil temperature over time for the unheated, fresh oil, 1000 rpm test.

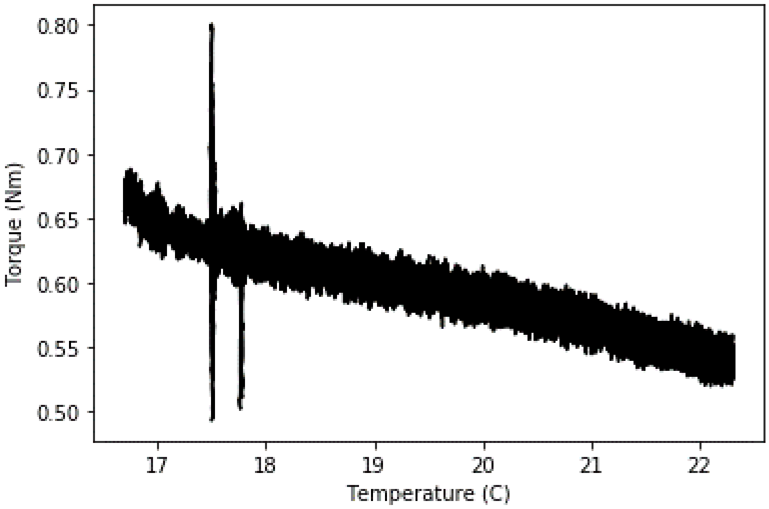


**Figure B-33:** This plot shows the relationship between motor torque and oil temperature for the unheated, fresh oil, 1000 rpm test.

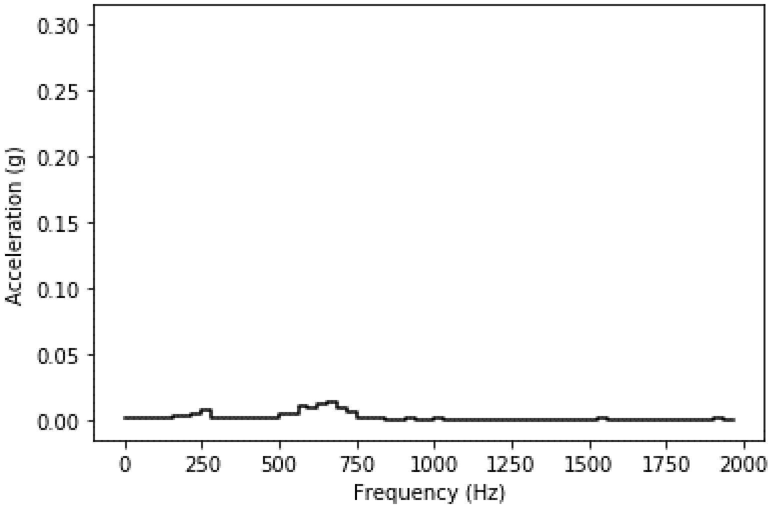


**Figure B-34:** Average acceleration magnitude per frequency bin for accelerometer 1, for the unheated, fresh oil, 1000 rpm test.

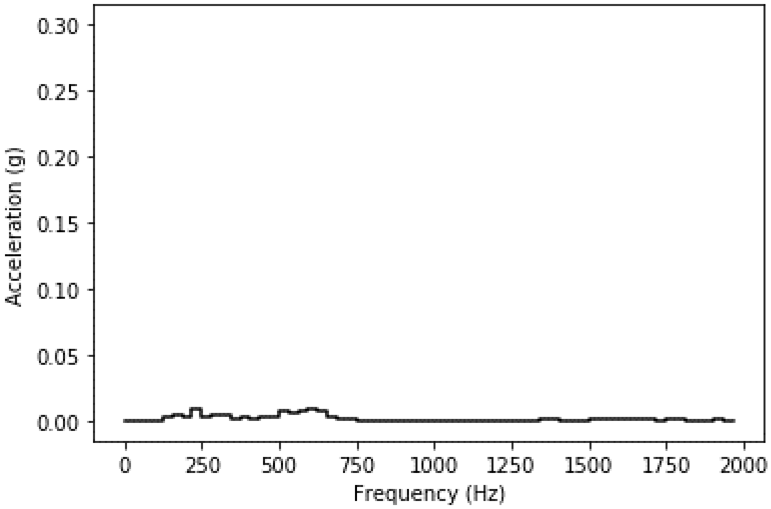


**Figure B-35:** Average acceleration magnitude per frequency bin for accelerometer 2, for the unheated, fresh oil, 1000 rpm test.

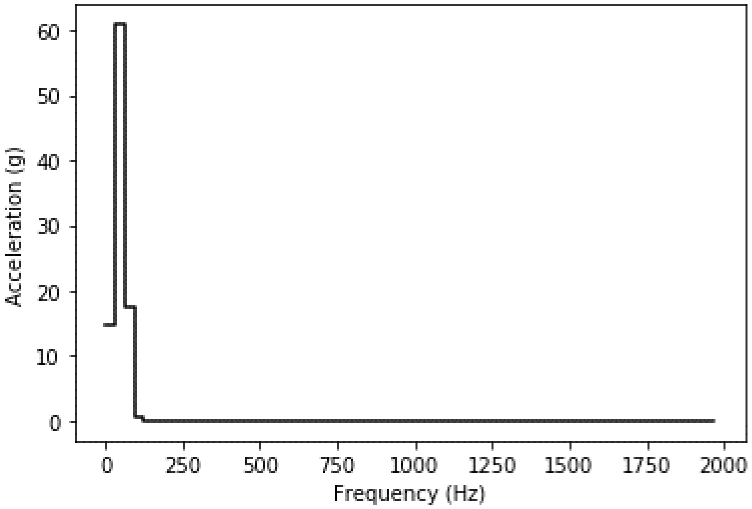


**Figure B-36:** Average acceleration magnitude per frequency bin for accelerometer 3, for the unheated, fresh oil, 1000 rpm test.

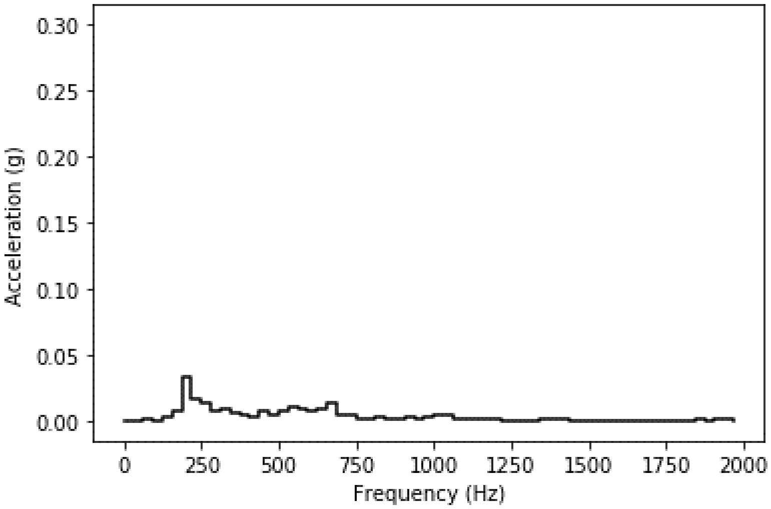


**Figure B-37:** Average acceleration magnitude per frequency bin for accelerometer 4, for the unheated, fresh oil, 1000 rpm test.

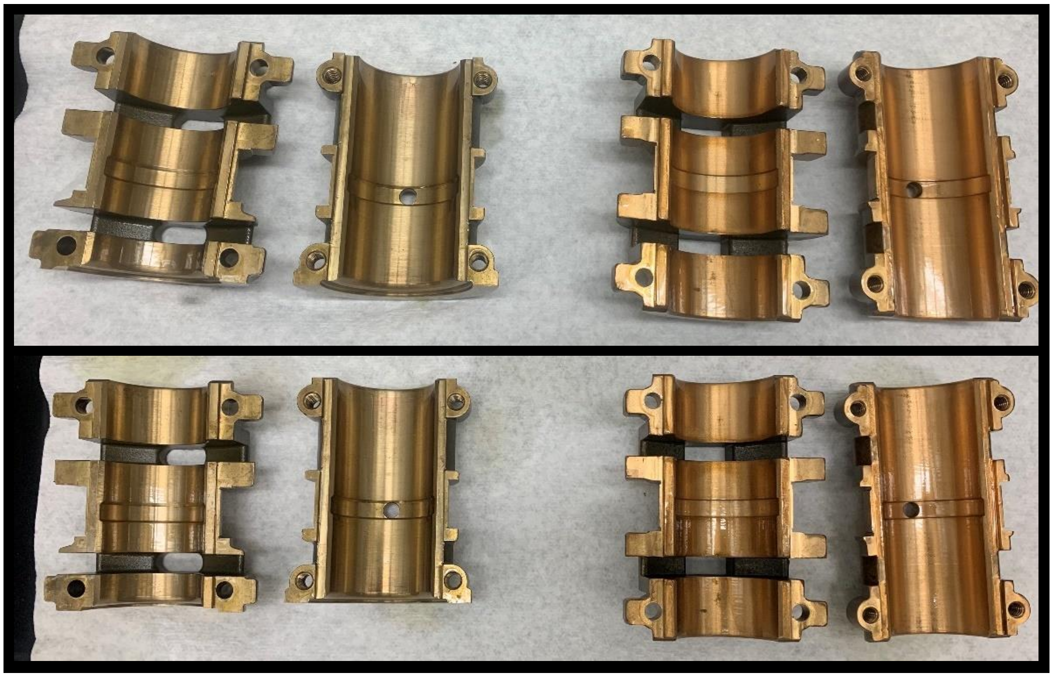


**Figure B-38:** ZS-1 journal bearings before (top) and after (bottom) the unheated, fresh oil, 1000 rpm test.

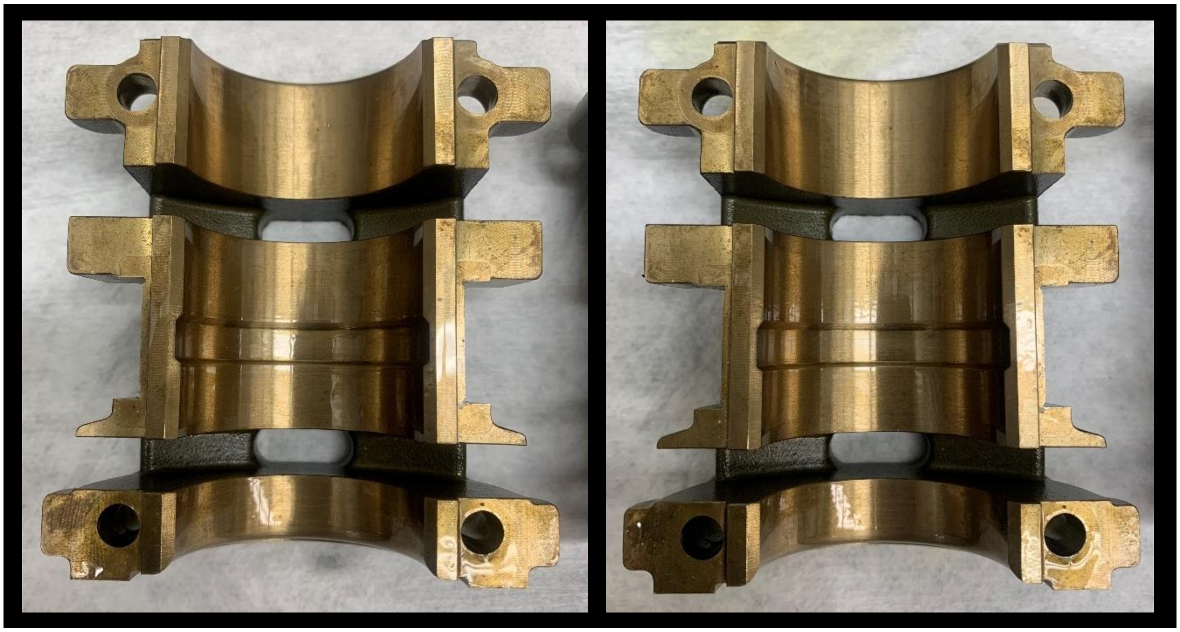


**Figure B-39:** Governor side top bearing before (left) and after (right) the unheated, fresh oil, 1000 rpm test.

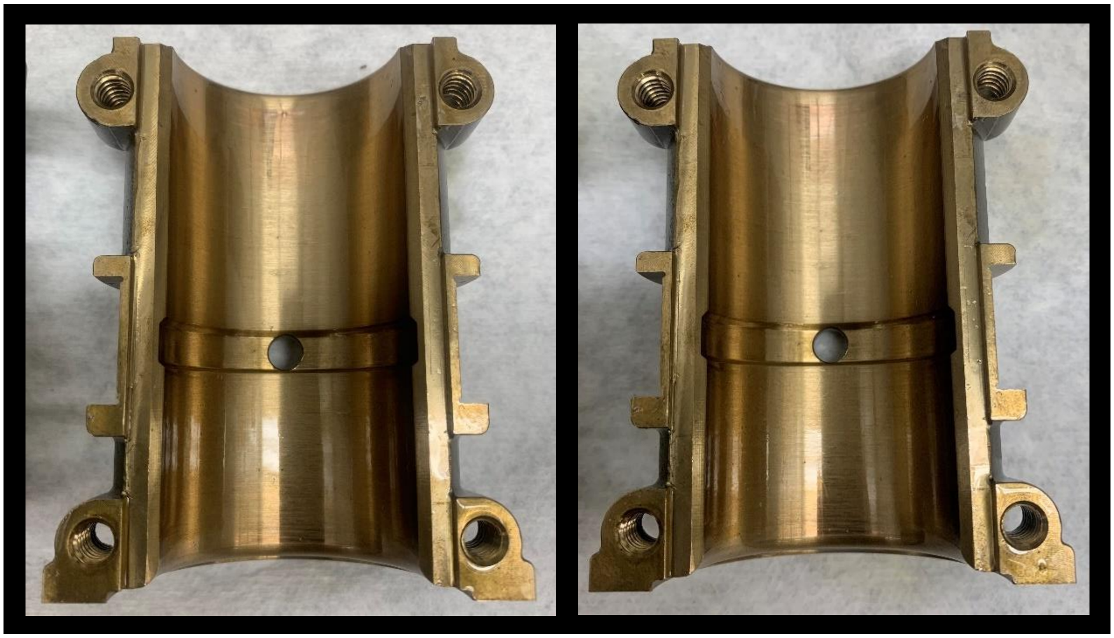


**Figure B-40:** Governor side bottom bearing before (left) and after (right) the unheated, fresh oil, 1000 rpm test.

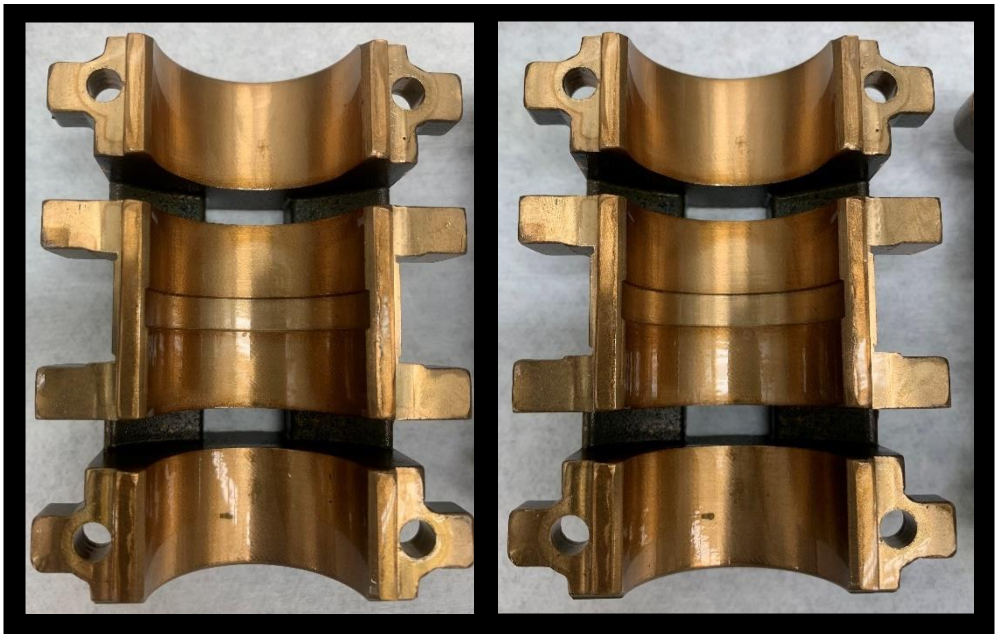


**Figure B-41:** Motor side top bearing before (left) and after (right) the unheated, fresh oil, 1000 rpm test.

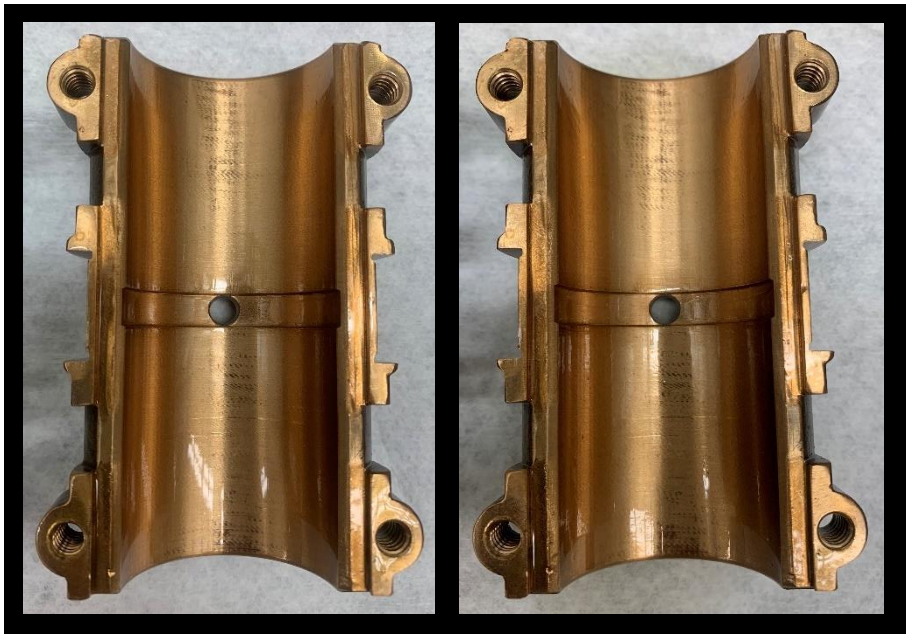


**Figure B-42:** Motor side bottom bearing before (left) and after (right) the unheated, fresh oil, 1000 rpm test.

### B.2.1.2. Fresh Oil, No Heat, 3000 rpm - 11/12/2019

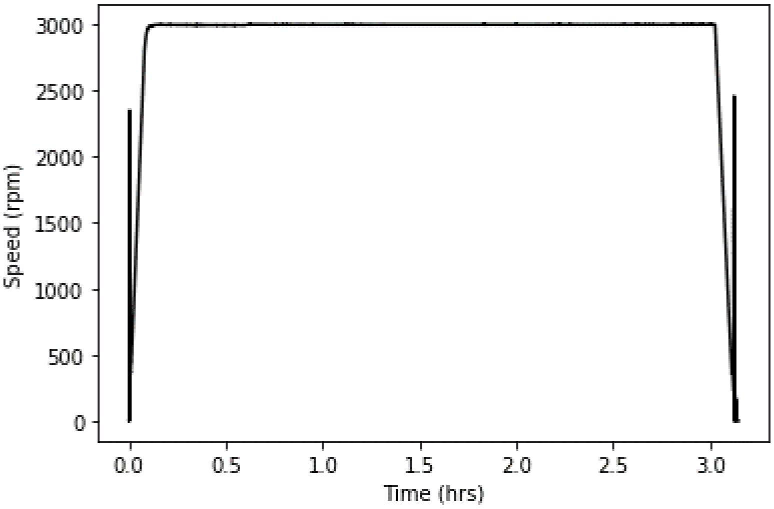


**Figure B-43:** Rotational shaft speed over time for the unheated, fresh oil, 3000 rpm test.

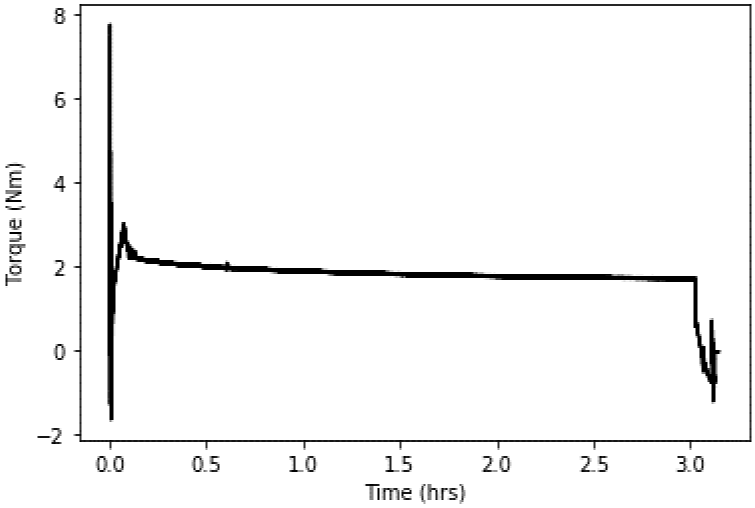


**Figure B-44:** Motor torque over time for the unheated, fresh oil, 3000 rpm test.

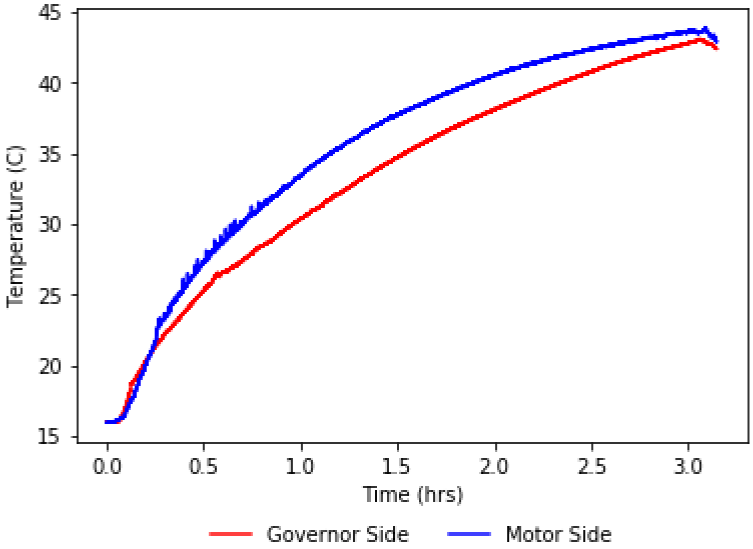


**Figure B-45:** Average oil temperature over time for the unheated, fresh oil, 3000 rpm test.

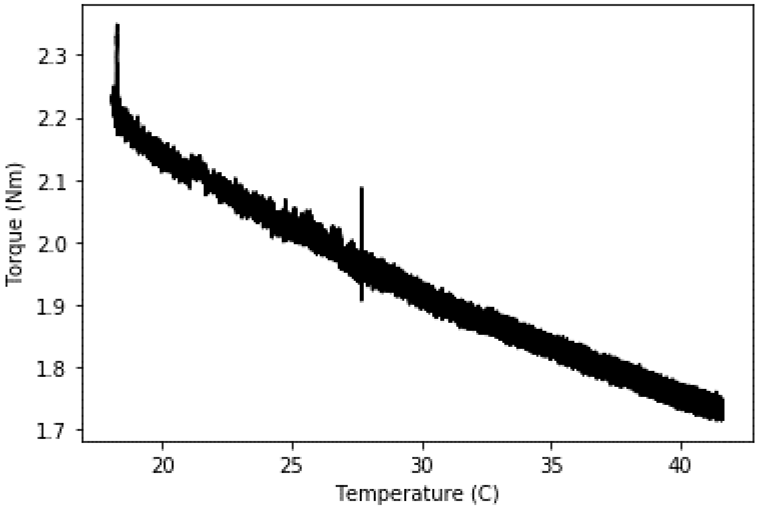


**Figure B-46:** This plot shows the relationship between motor torque and oil temperature for the unheated, fresh oil, 3000 rpm test.

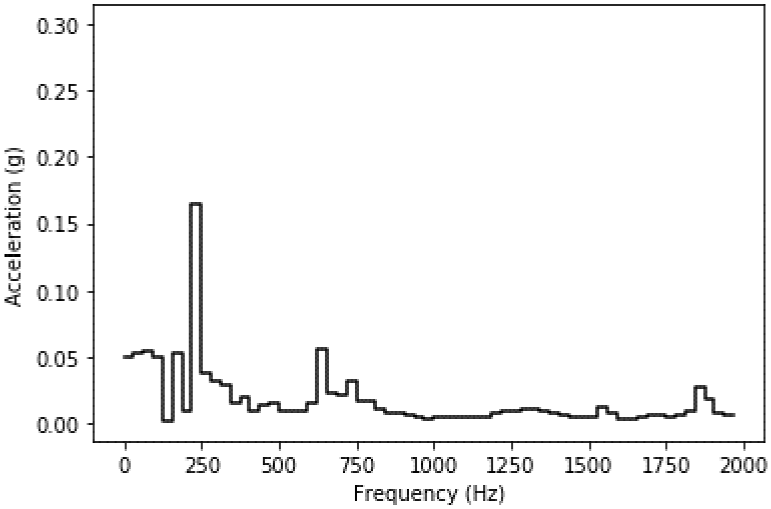


**Figure B-47:** Average acceleration magnitude per frequency bin for accelerometer 1, for the unheated, fresh oil, 3000 rpm test.

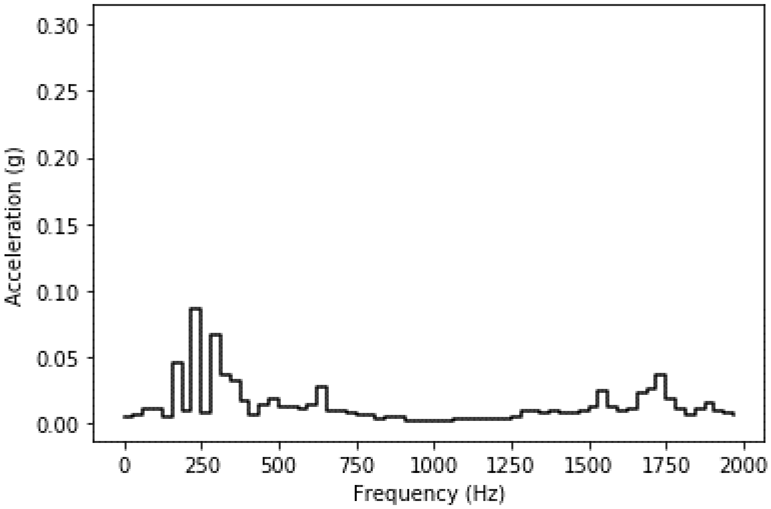


**Figure B-48:** Average acceleration magnitude per frequency bin for accelerometer 2, for the unheated, fresh oil, 3000 rpm test.

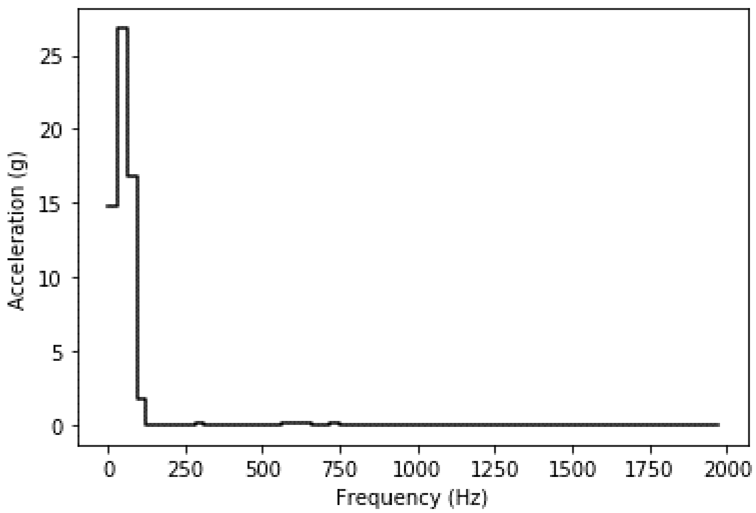


**Figure B-49:** Average acceleration magnitude per frequency bin for accelerometer 3, for the unheated, fresh oil, 3000 rpm test.

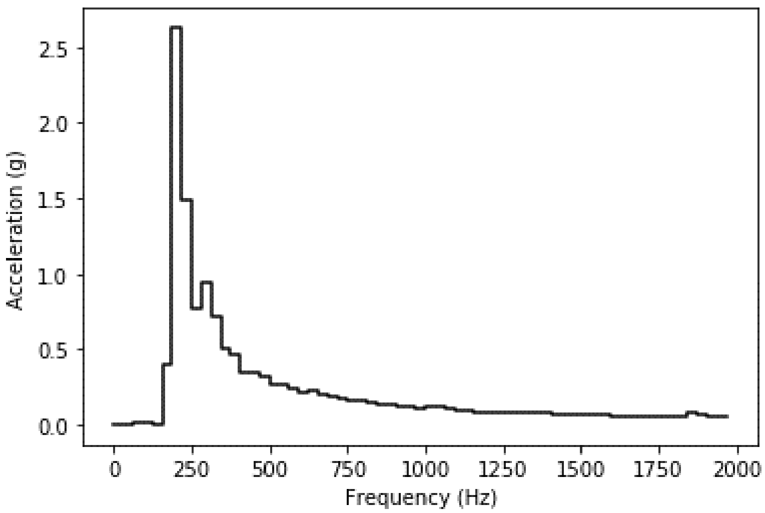


**Figure B-50:** Average acceleration magnitude per frequency bin for accelerometer 4, for the unheated, fresh oil, 3000 rpm test.

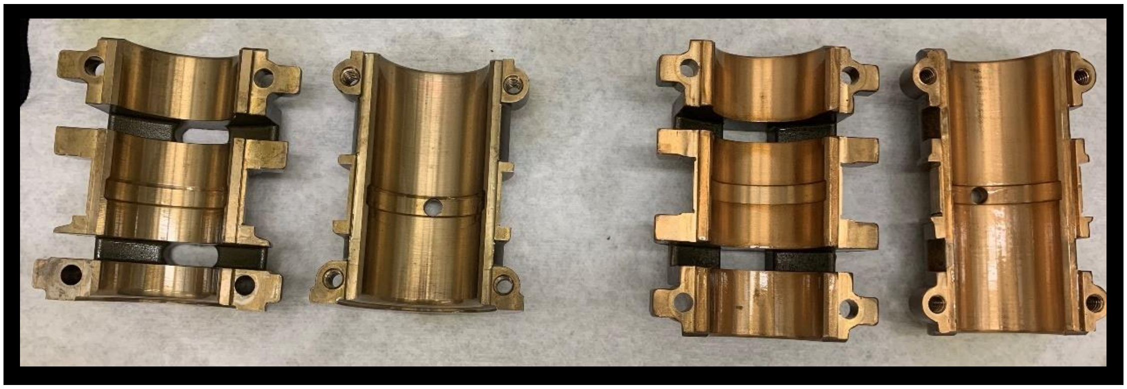


**Figure B-51:** ZS-1 journal bearings after the unheated, fresh oil, 3000 rpm test.

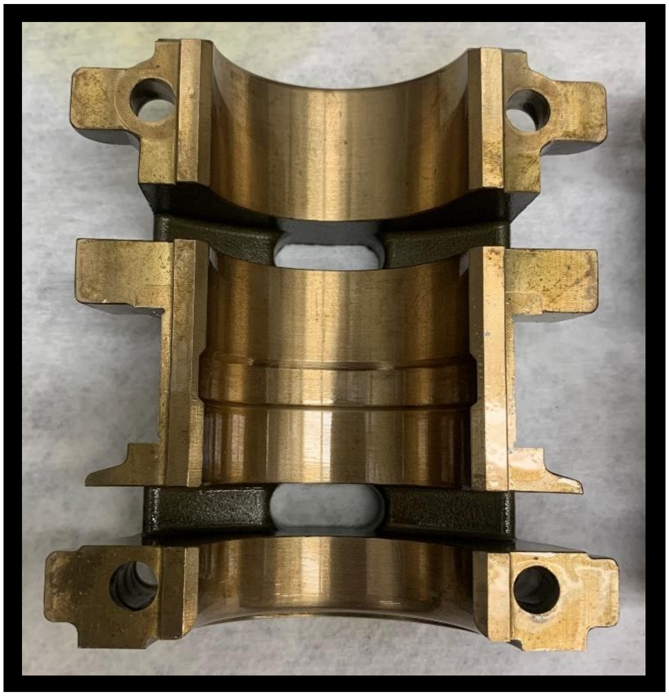


**Figure B-52:** Governor side top bearing after the unheated, fresh oil, 3000 rpm test.

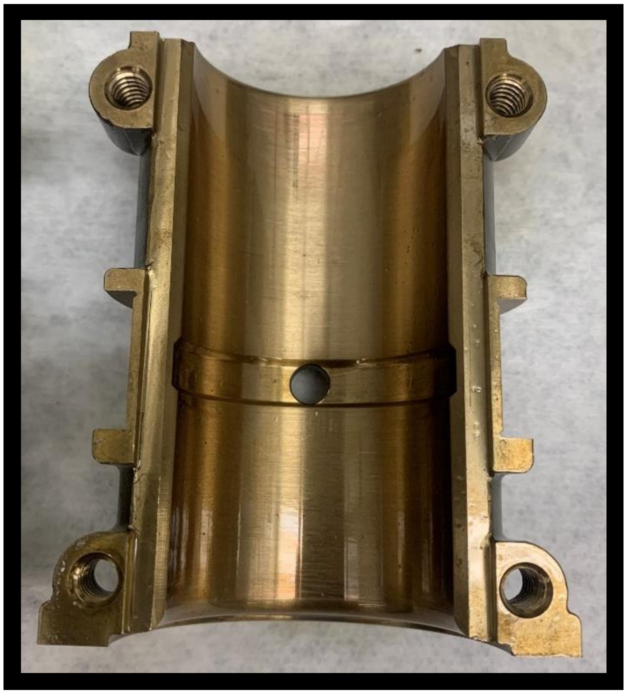


**Figure B-53:** Governor side bottom bearing after the unheated, fresh oil, 3000 rpm test.

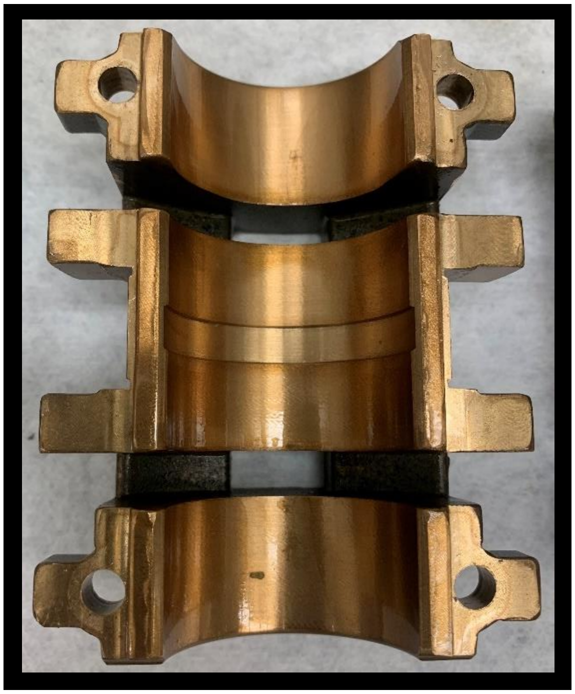


**Figure B-54:** Motor side top bearing after the unheated, fresh oil, 3000 rpm test.

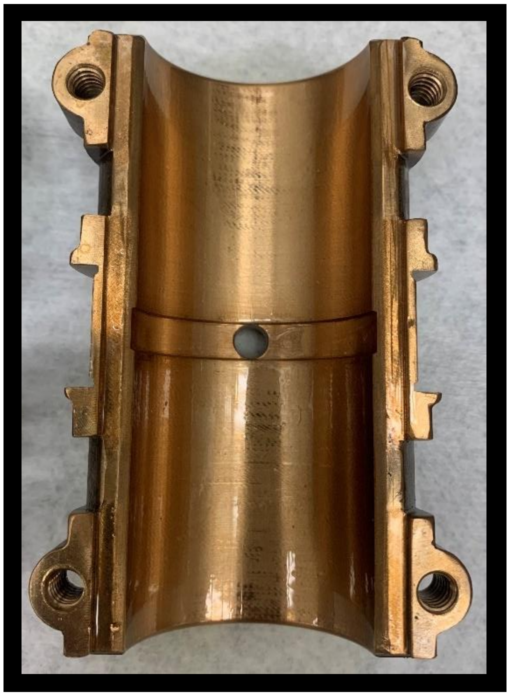


**Figure B-55:** Motor side bottom bearing after the unheated, fresh oil, 3000 rpm test.

B.2.1.3. Fresh Oil, 121ºC, 1000 rpm - 11/13/2019

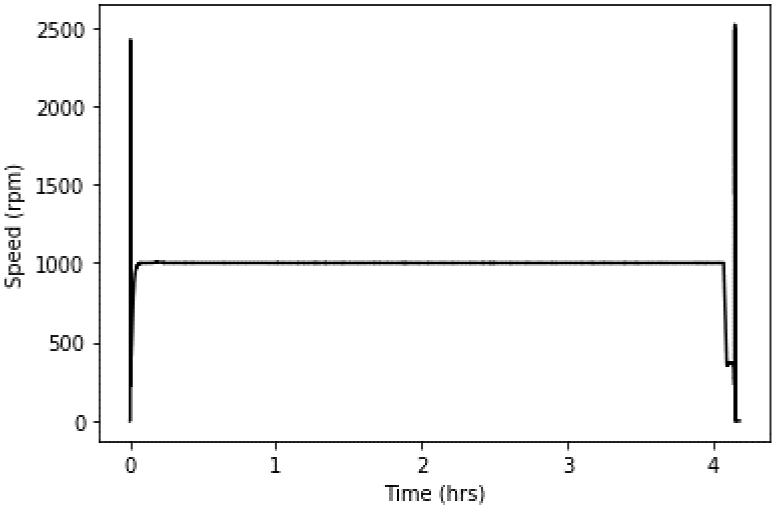


**Figure B-56:** Rotational shaft speed over time for the heated, fresh oil, 1000 rpm test.

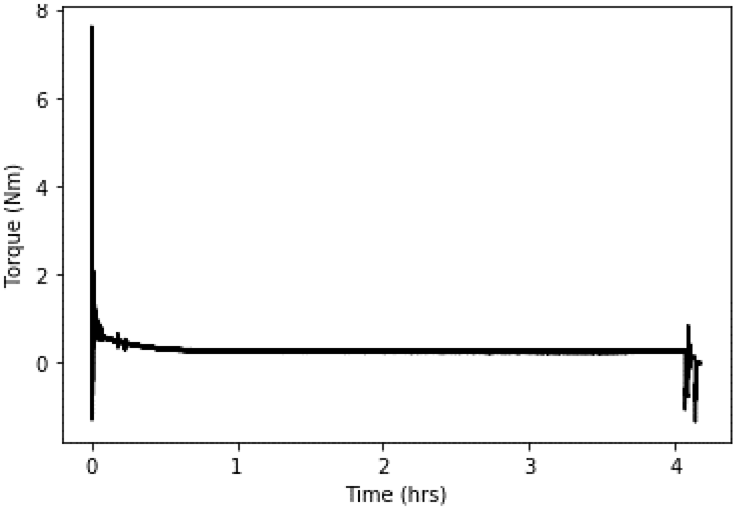


**Figure B-57:** Motor torque over time for the heated, fresh oil, 1000 rpm test.

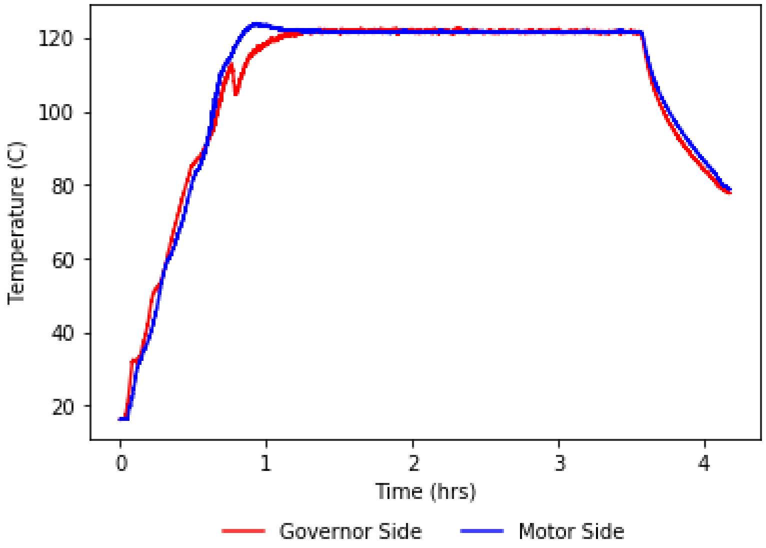


**Figure B-58:** Average oil temperature over time for the heated, fresh oil, 1000 rpm test.

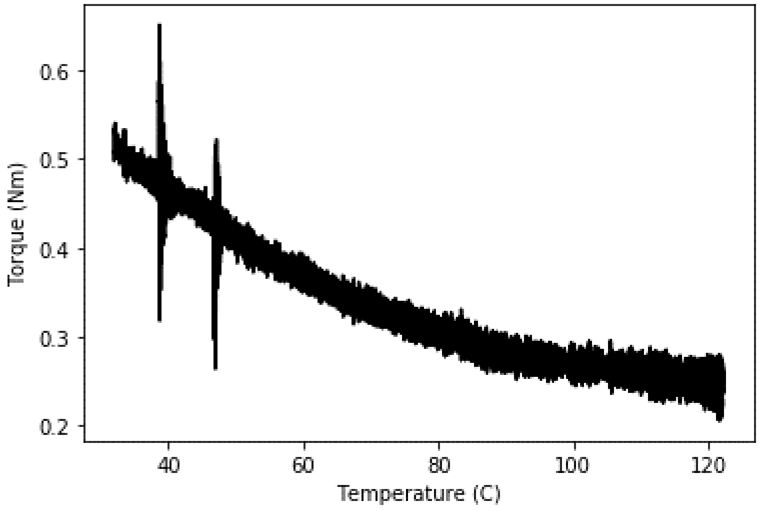


**Figure B-59:** This plot shows the relationship between motor torque and oil temperature for the heated, fresh oil, 1000 rpm test.

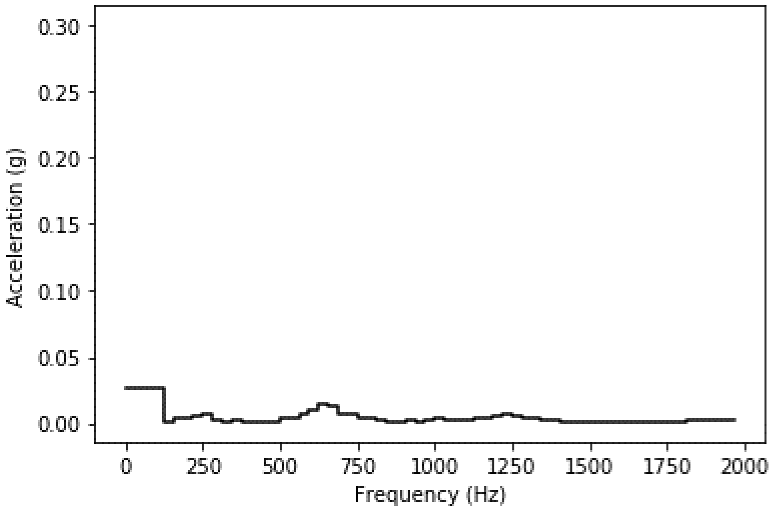


**Figure B-60:** Average acceleration magnitude per frequency bin for accelerometer 1, for the heated, fresh oil, 1000 rpm test.

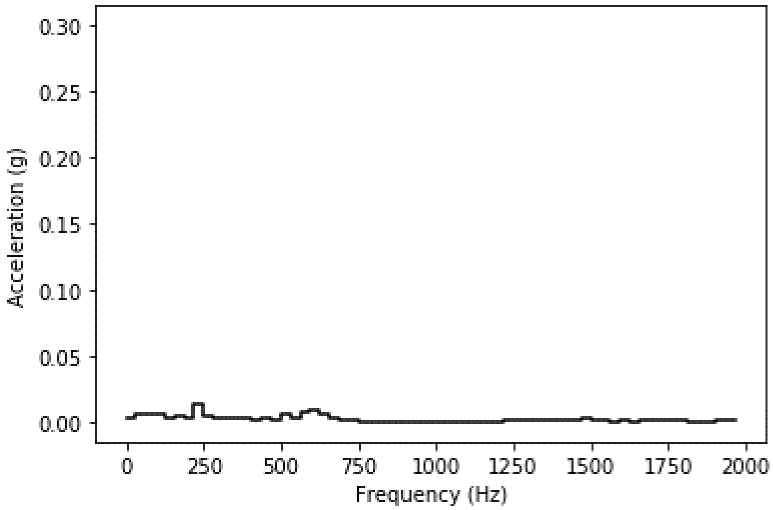


**Figure B-61:** Average acceleration magnitude per frequency bin for accelerometer 2, for the heated, fresh oil, 1000 rpm test.

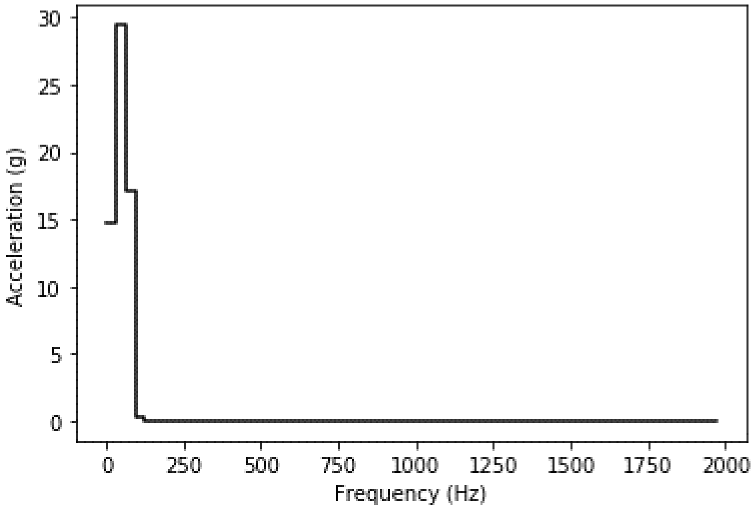


**Figure B-62:** Average acceleration magnitude per frequency bin for accelerometer 3, for the heated, fresh oil, 1000 rpm test.

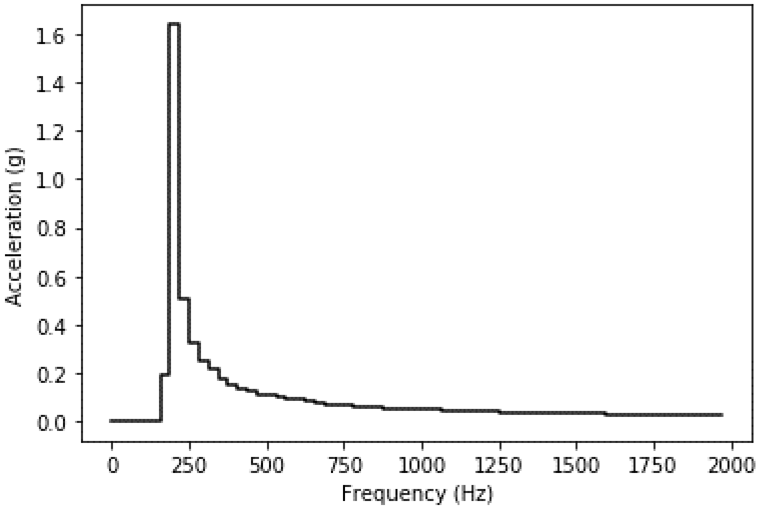


**Figure B-63:** Average acceleration magnitude per frequency bin for accelerometer 4, for the heated, fresh oil, 1000 rpm test.

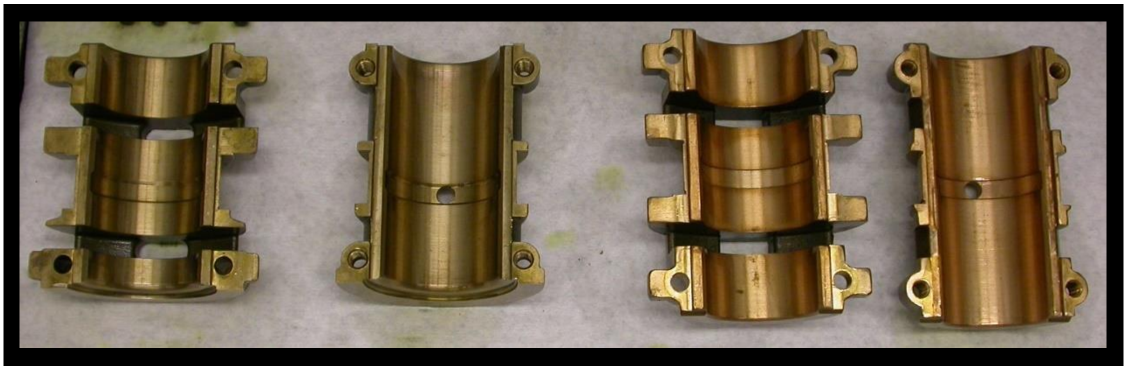


**Figure B-64:** ZS-1 journal bearings after the heated, fresh oil, 1000 rpm test.

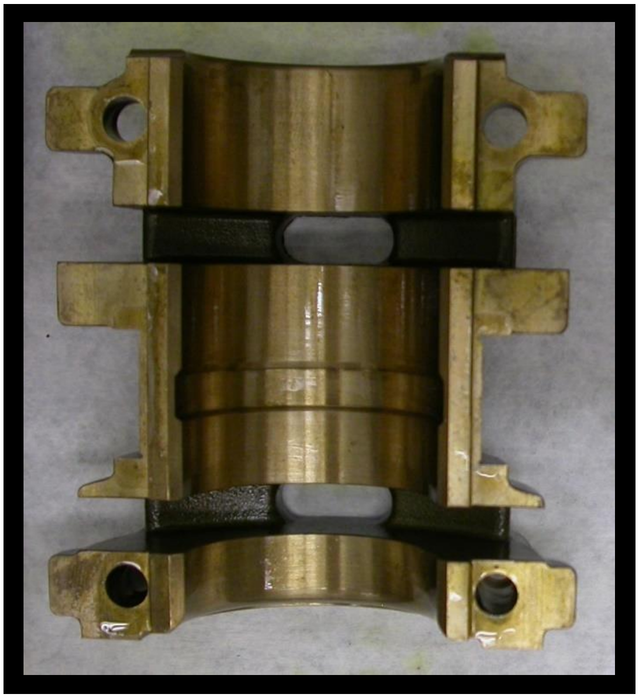


**Figure B-65:** Governor side top bearing after the heated, fresh oil, 1000 rpm test.

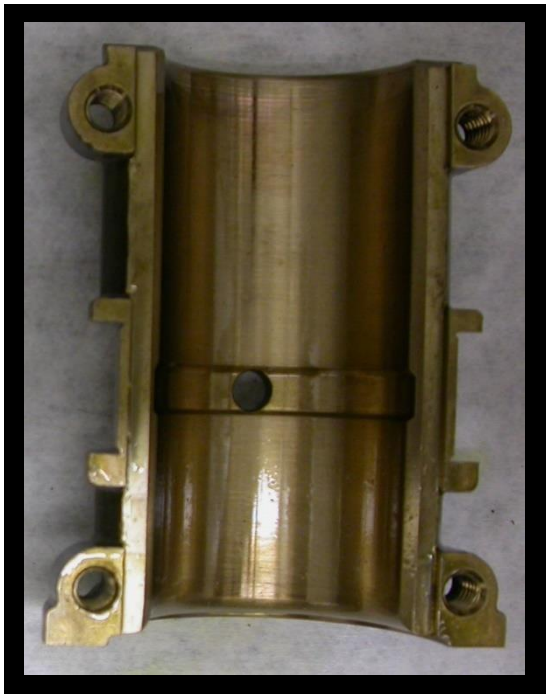


**Figure B-66:** Governor side bottom bearing after the heated, fresh oil, 1000 rpm test.

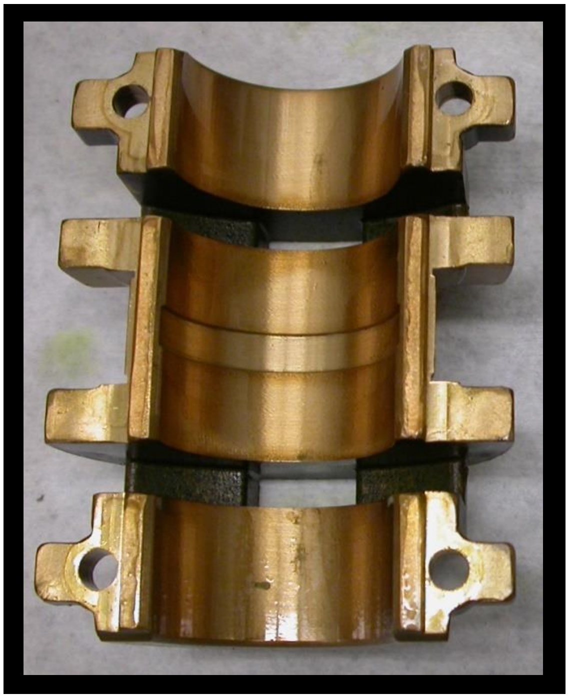


**Figure B-67:** Motor side top bearing after the heated, fresh oil, 1000 rpm test.

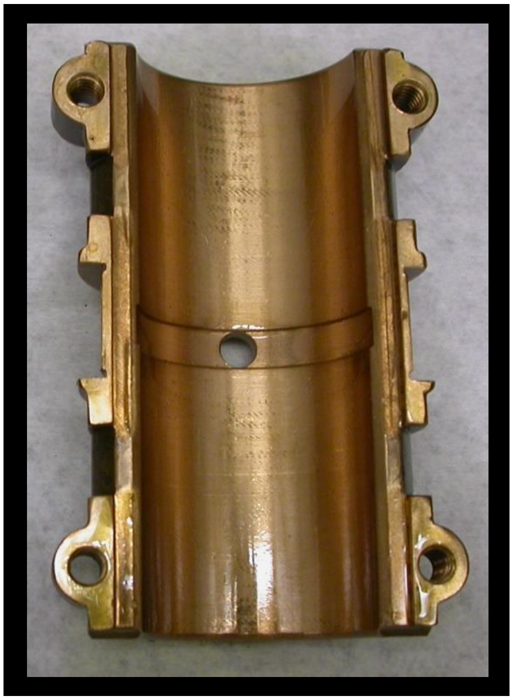


**Figure B-68:** Shown here is the motor side bottom bearing after the heated, fresh oil, 1000 rpm test.

### B.2.1.4. Fresh Oil, 121$^{o}$C, 3000 rpm - 12/03/2019

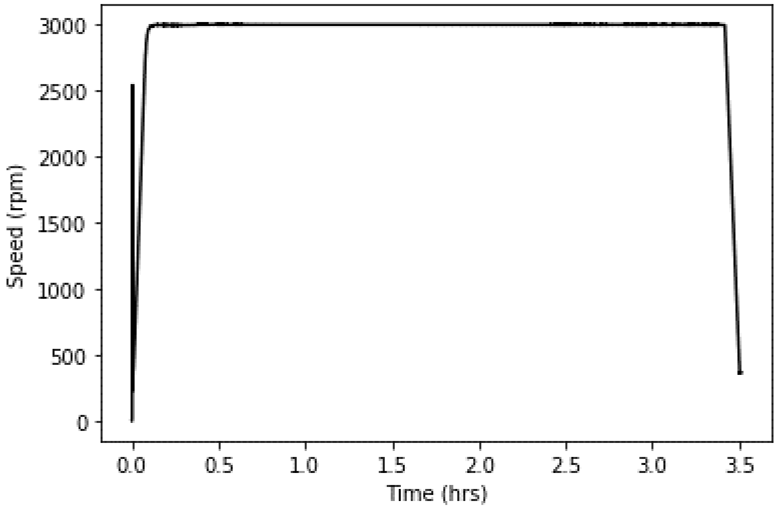


**Figure B-69:** Rotational shaft speed over time for the heated, fresh oil, 3000 rpm test.

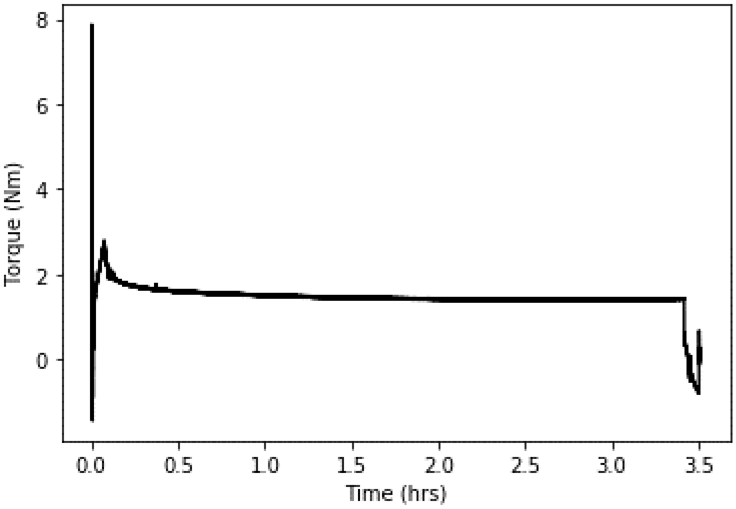


**Figure B-70:** Motor torque over time for the heated, fresh oil, 3000 rpm test.

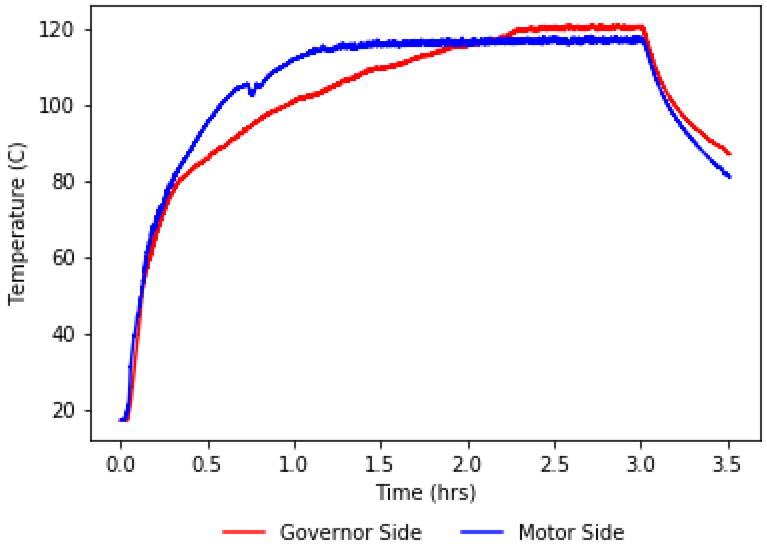


**Figure B-71:** Average oil temperature over time for the heated, fresh oil, 3000 rpm test.

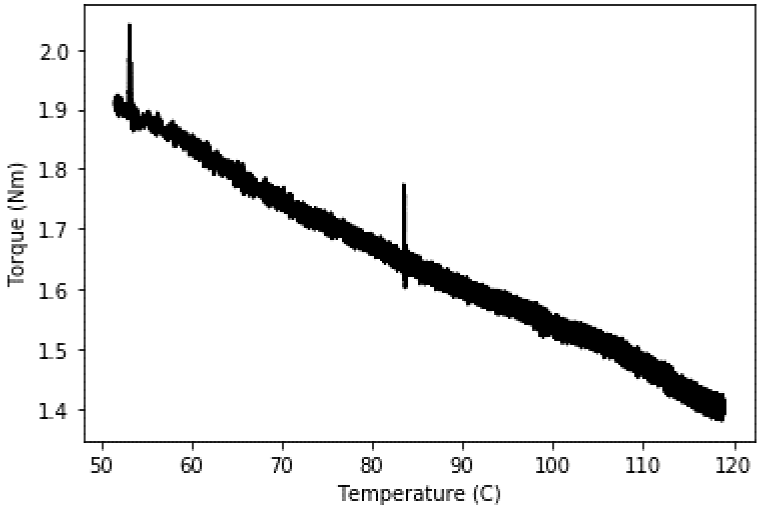


**Figure B-72:** This plot shows the relationship between motor torque and oil temperature for the heated, fresh oil, 3000 rpm test.

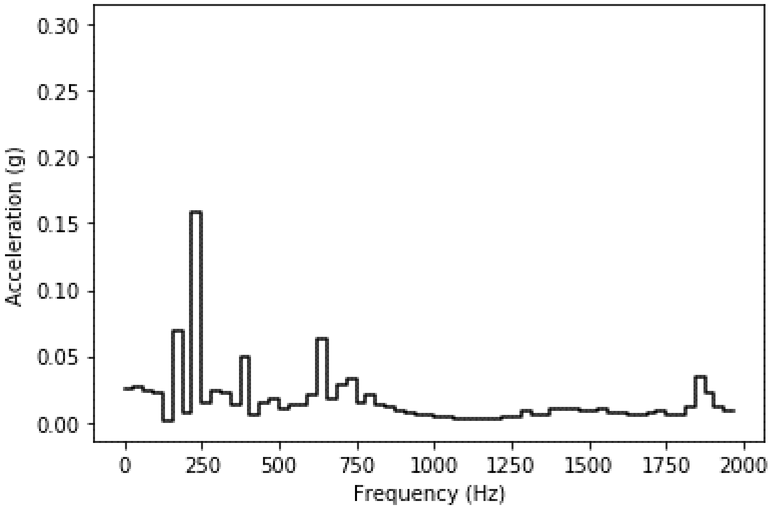


**Figure B-73:** Average acceleration magnitude per frequency bin for accelerometer 1, for the heated, fresh oil, 3000 rpm test.

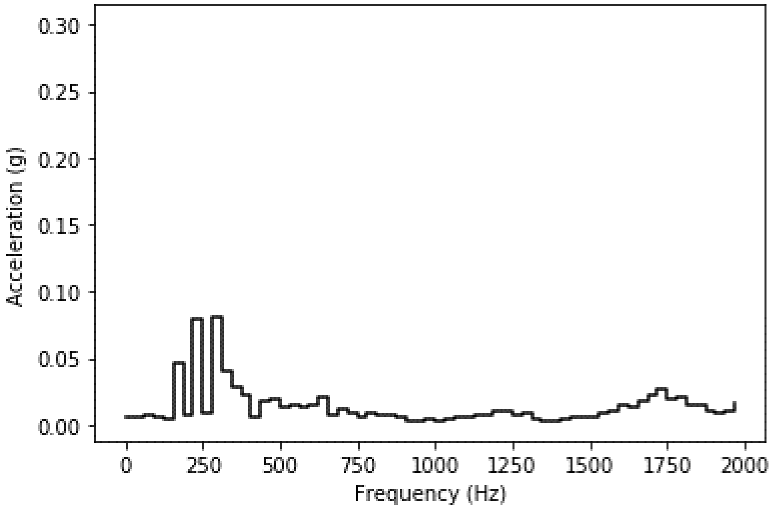


**Figure B-74:** Average acceleration magnitude per frequency bin for accelerometer 2, for the heated, fresh oil, 3000 rpm test.

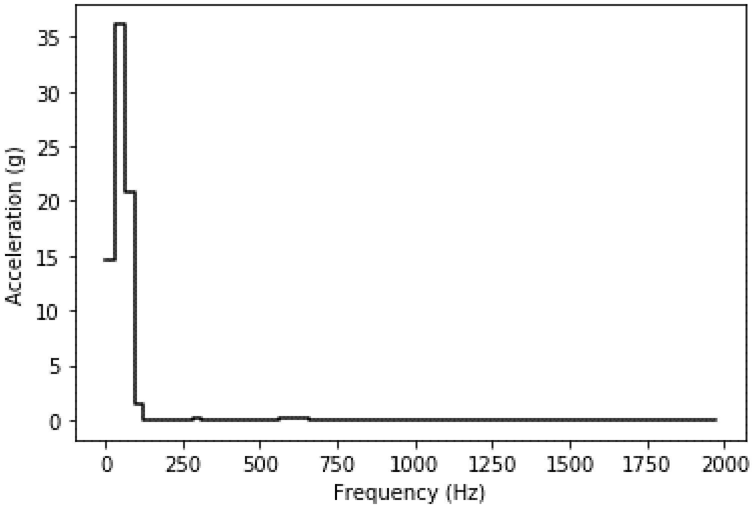


**Figure B-75:** Average acceleration magnitude per frequency bin for accelerometer 3, for the heated, fresh oil, 3000 rpm test.

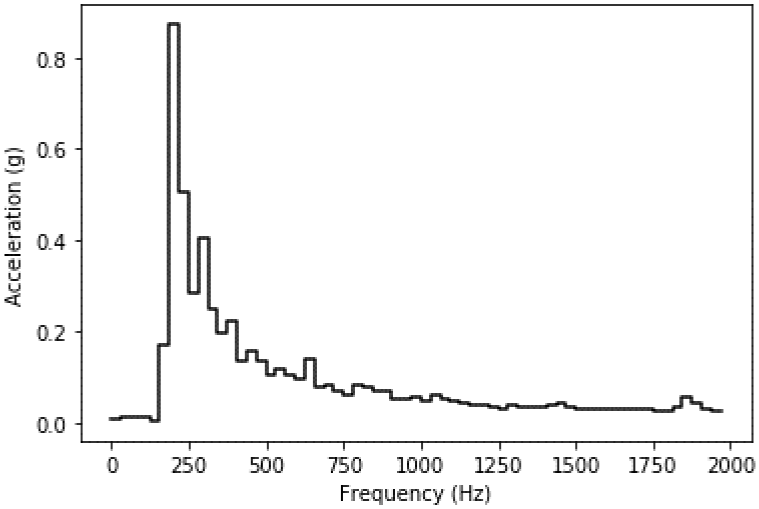


**Figure B-76:** Average acceleration magnitude per frequency bin for accelerometer 4, for the heated, fresh oil, 3000 rpm test.

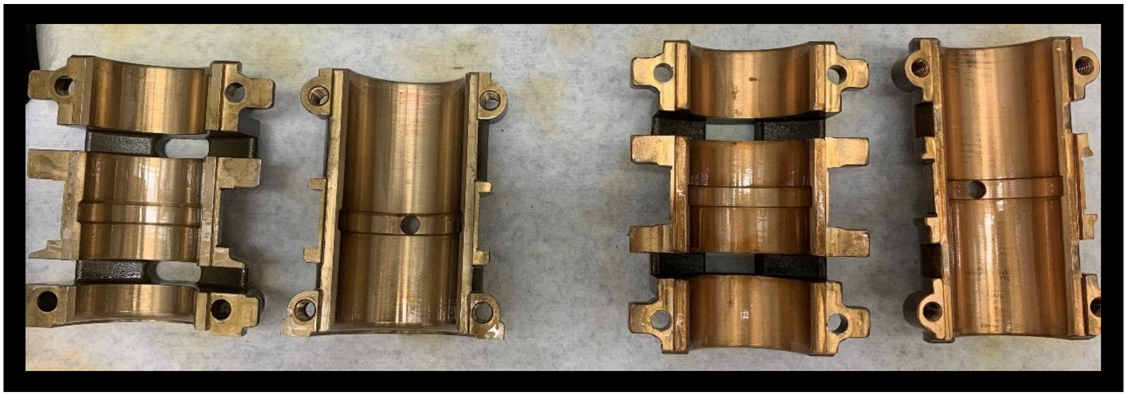


**Figure B-77:** ZS-1 journal bearings after the heated, fresh oil, 3000 rpm test.

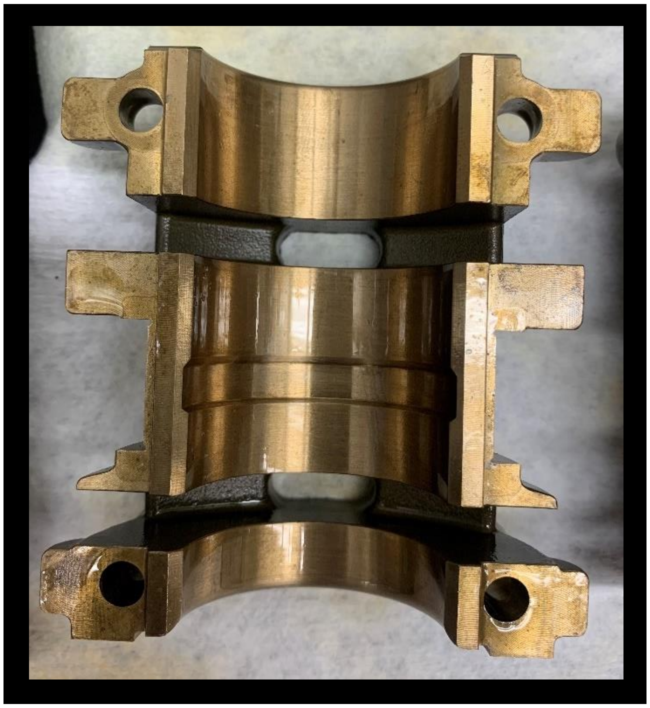


**Figure B-78:** Governor side top bearing after the heated, fresh oil, 3000 rpm test.

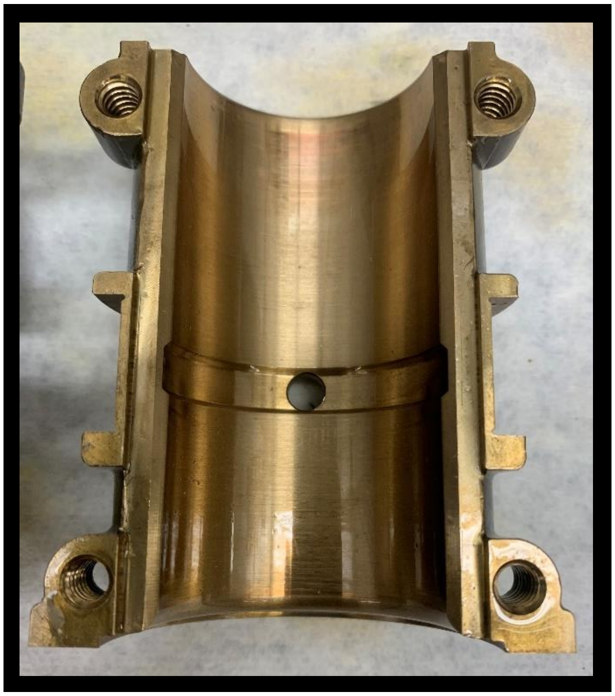


**Figure B-79:** Governor side bottom bearing after the heated, fresh oil, 3000 rpm test.

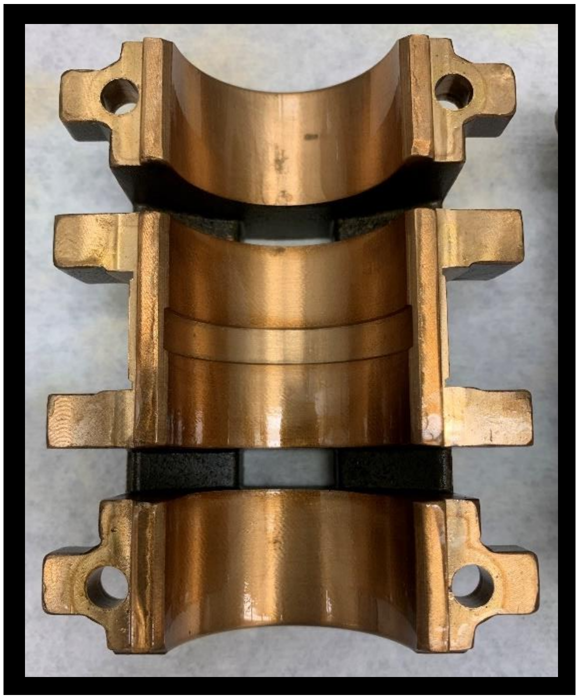


**Figure B-80:** Motor side top bearing after the heated, fresh oil, 3000 rpm test.

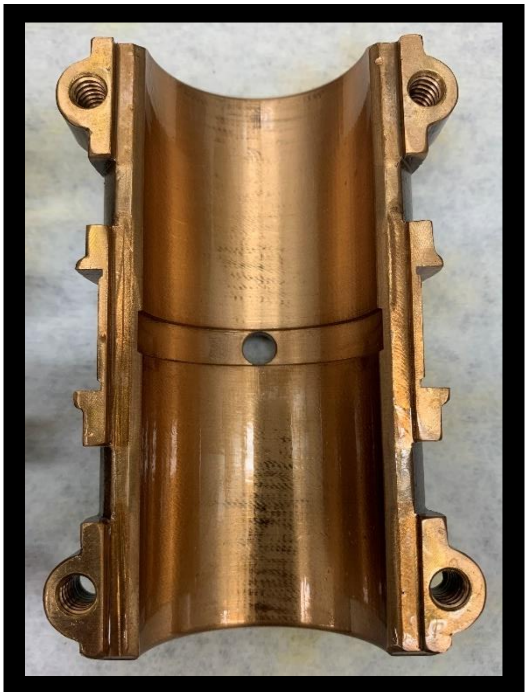


**Figure B-81:** Motor side top bearing after the heated, fresh oil, 3000 rpm test.

B.2.1.5. Degraded Oil, No Heat, 1000 rpm - 12/16/2019

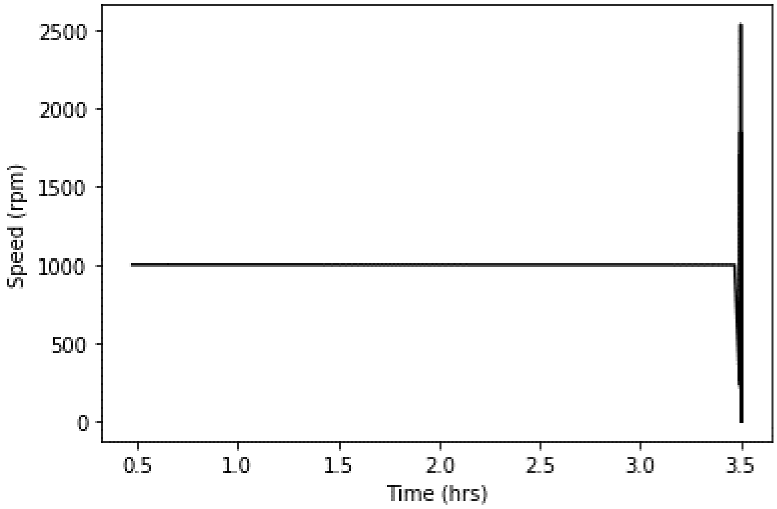


**Figure B-82:** Rotational shaft speed over time for the unheated, degraded oil, 1000 rpm test.

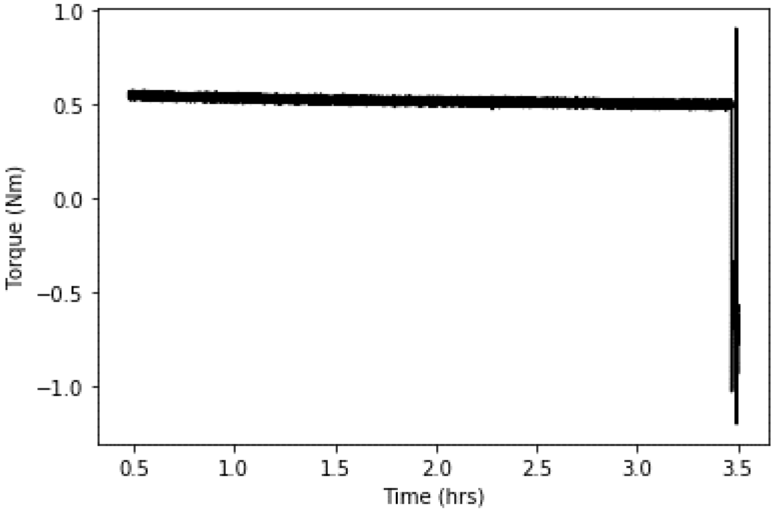


**Figure B-83:** Motor torque over time for the unheated, degraded oil, 1000 rpm test.

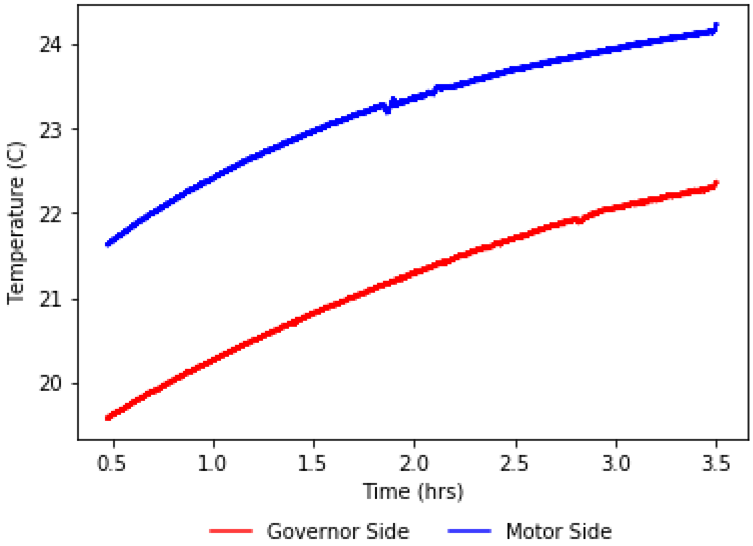


**Figure B-84:** Average oil temperature over time for the unheated, degraded oil, 1000 rpm test.

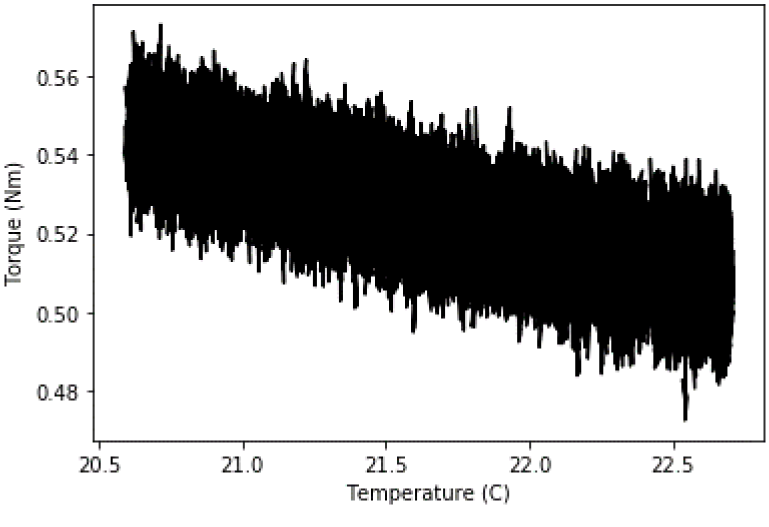


**Figure B-85:** This plot shows the relationship between motor torque and oil temperature for the unheated, degraded oil, 1000 rpm test.

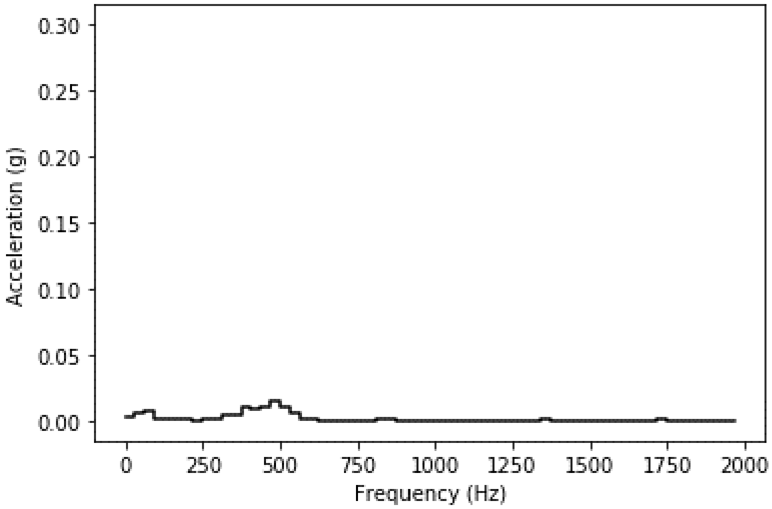


**Figure B-86:** Average acceleration magnitude per frequency bin for accelerometer 1, for the unheated, degraded oil, 1000 rpm test.

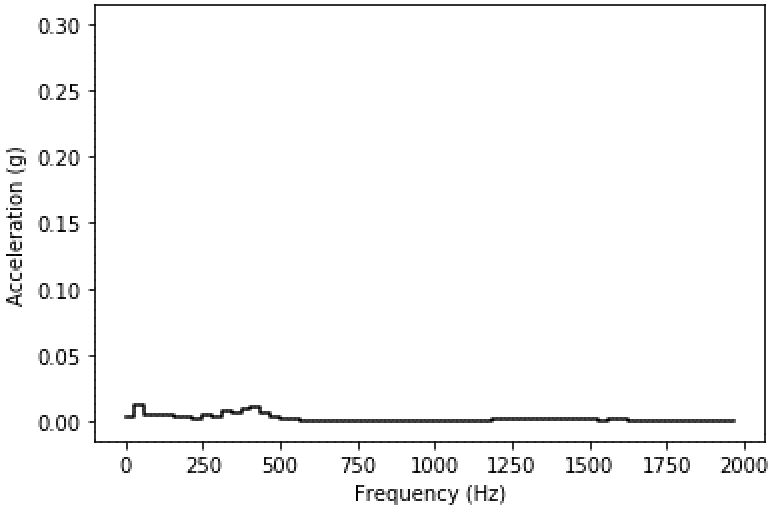


**Figure B-87:** Average acceleration magnitude per frequency bin for accelerometer 2, for the unheated, degraded oil, 1000 rpm test.

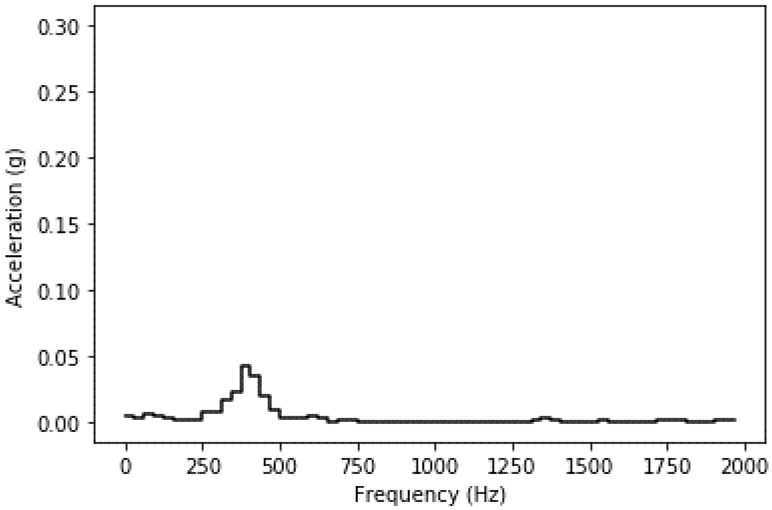


**Figure B-88:** Average acceleration magnitude per frequency bin for accelerometer 3, for the unheated, degraded oil, 1000 rpm test.

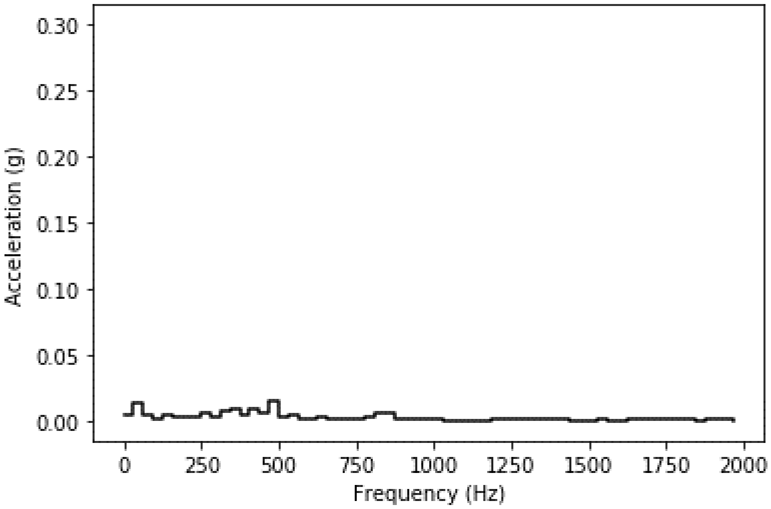


**Figure B-89:** Average acceleration magnitude per frequency bin for accelerometer 4, for the unheated, degraded oil, 1000 rpm test.

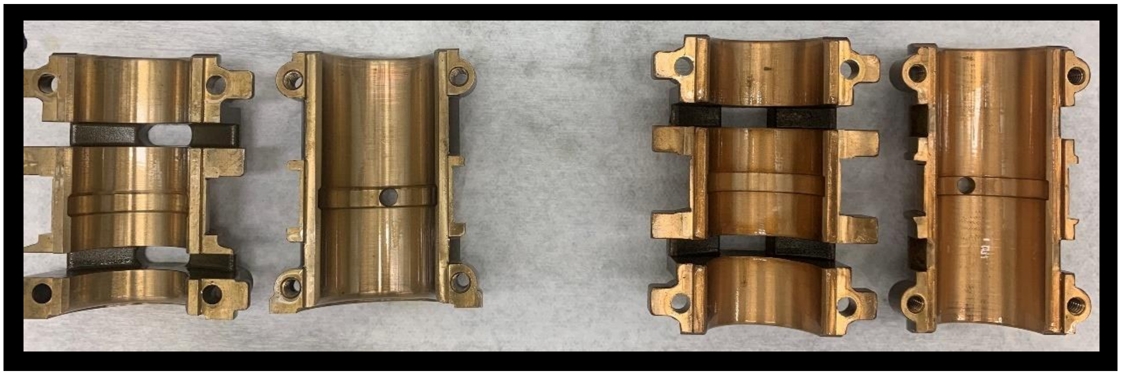


**Figure B-90:** ZS-1 journal bearings after the unheated, degraded oil, 1000 rpm test.

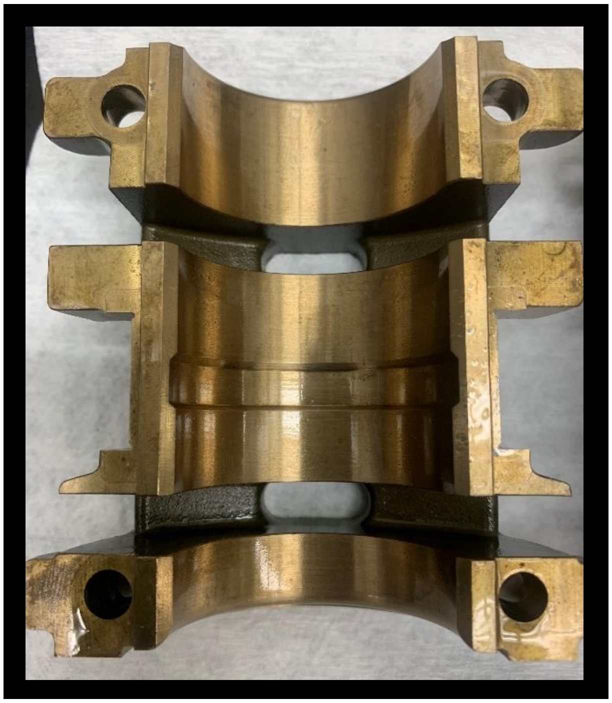


**Figure B-91:** Governor side top bearing after the unheated, degraded oil, 1000 rpm test.

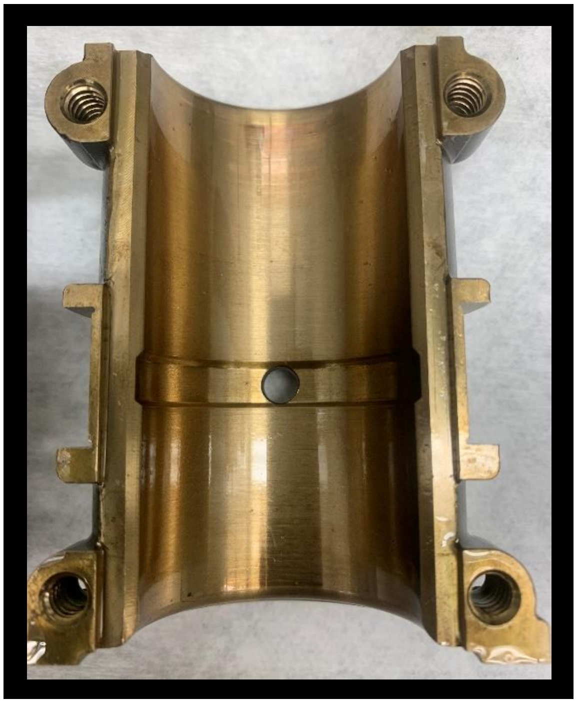


**Figure B-92:** Governor side bottom bearing after the unheated, degraded oil, 1000 rpm test.

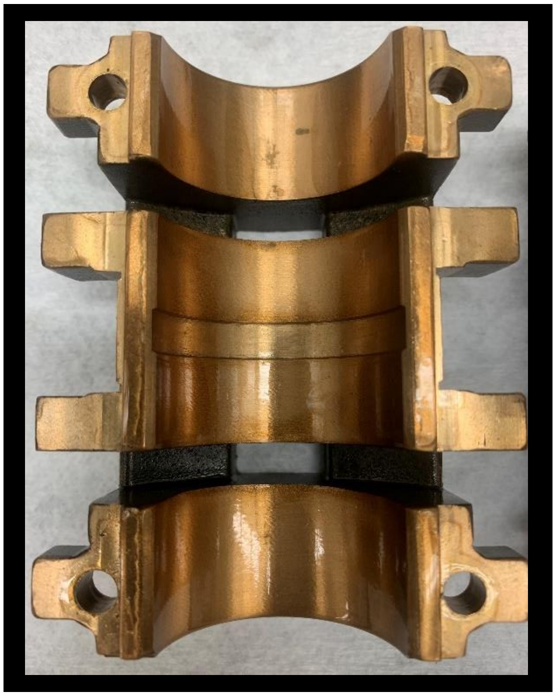


**Figure B-93:** Motor side top bearing after the unheated, degraded oil, 1000 rpm test.

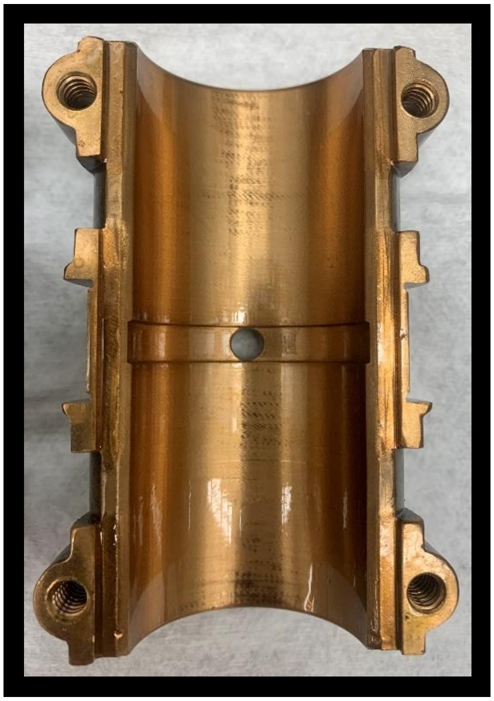


**Figure B-94:** Motor side bottom bearing after the unheated, degraded oil, 1000 rpm test.

### B.2.1.6. Degraded Oil, No Heat, 3000 rpm - 01/06/2020

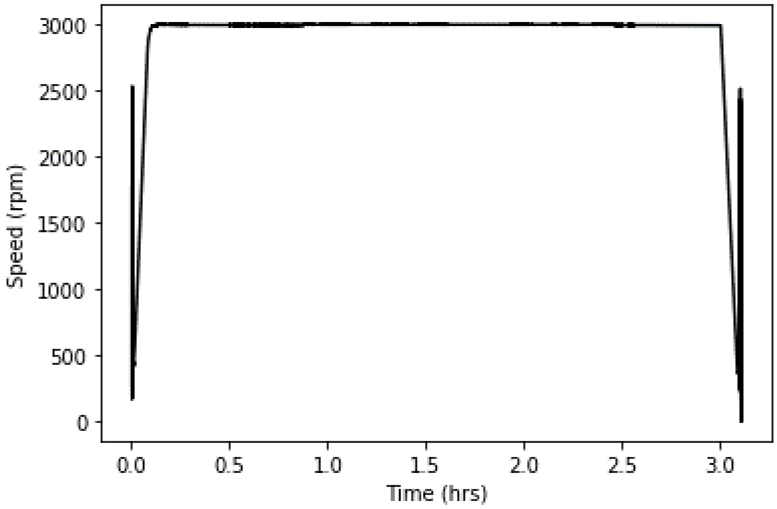


**Figure B-95:** Rotational shaft speed over time for the unheated, degraded oil, 3000 rpm test.

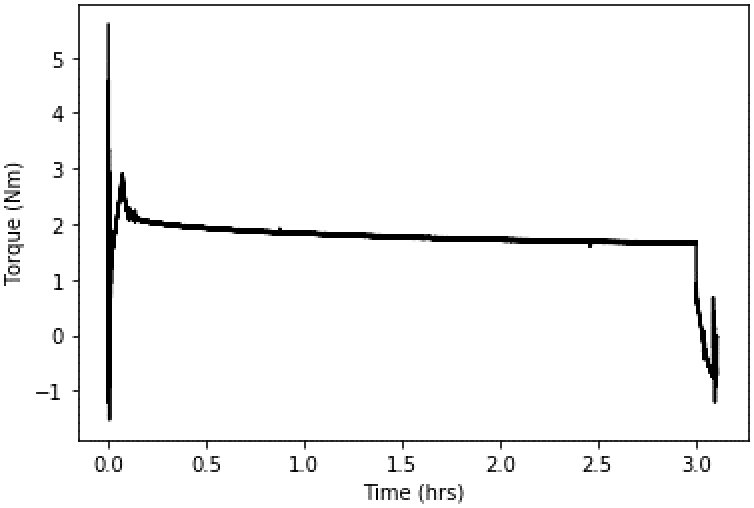


**Figure B-96:** Motor torque over time for the unheated, degraded oil, 3000 rpm test.

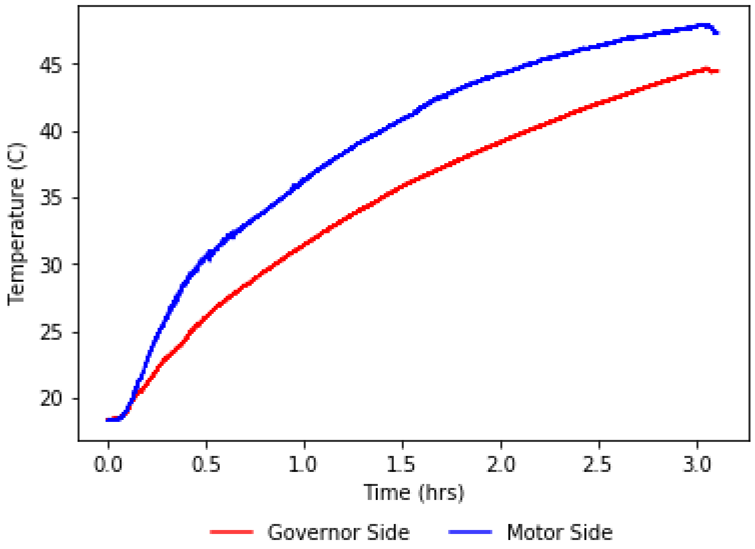


**Figure B-97:** Average oil temperature over time for the unheated, degraded oil, 3000 rpm test.

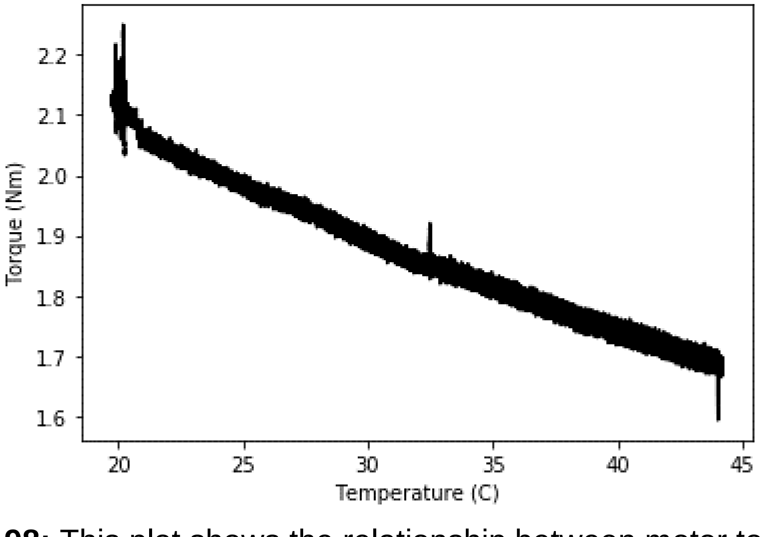


**Figure B-98:** This plot shows the relationship between motor torque and oil temperature for the unheated, degraded oil, 3000 rpm test.

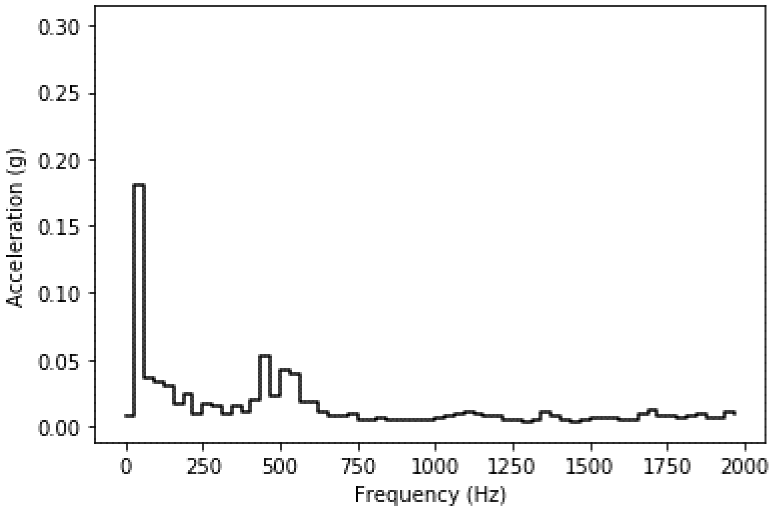


**Figure B-99:** Average acceleration magnitude per frequency bin for accelerometer 1, for the unheated, degraded oil, 3000 rpm test.

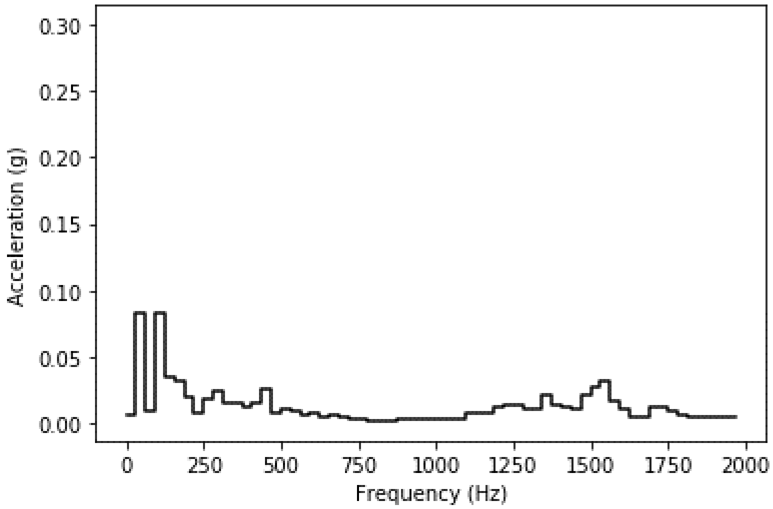


**Figure B-100:** Average acceleration magnitude per frequency bin for accelerometer 2, for the unheated, degraded oil, 3000 rpm test.

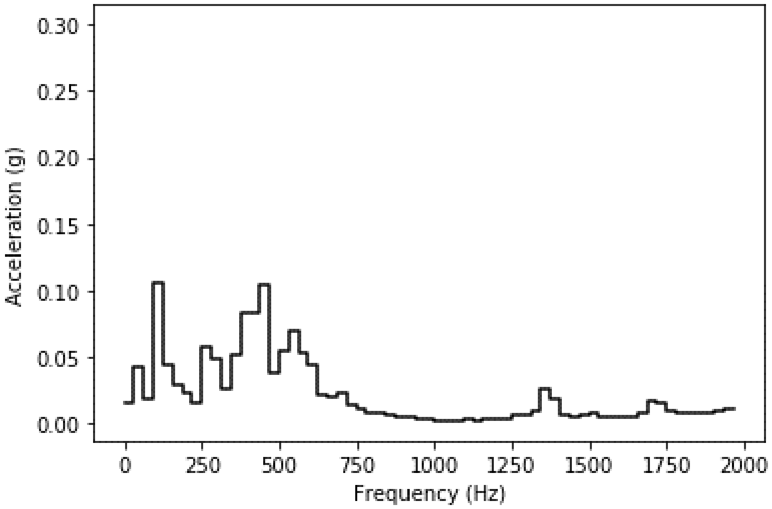


**Figure B-101:** Average acceleration magnitude per frequency bin for accelerometer 3, for the unheated, degraded oil, 3000 rpm test.

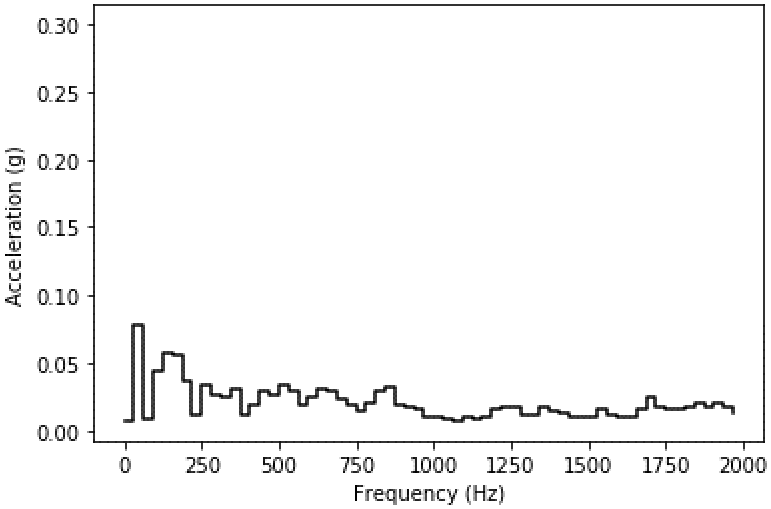


**Figure B-102:** Average acceleration magnitude per frequency bin for accelerometer 4, for the unheated, degraded oil, 3000 rpm test.

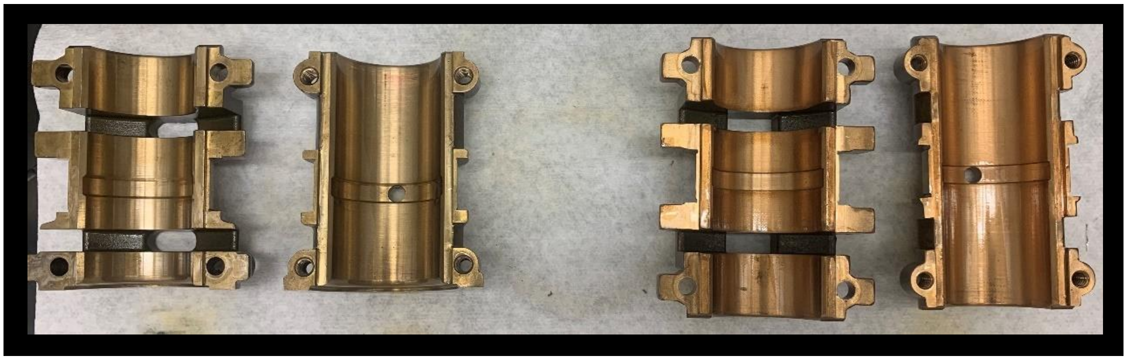


**Figure B-103:** ZS-1 journal bearings after the unheated, degraded oil, 3000 rpm test.

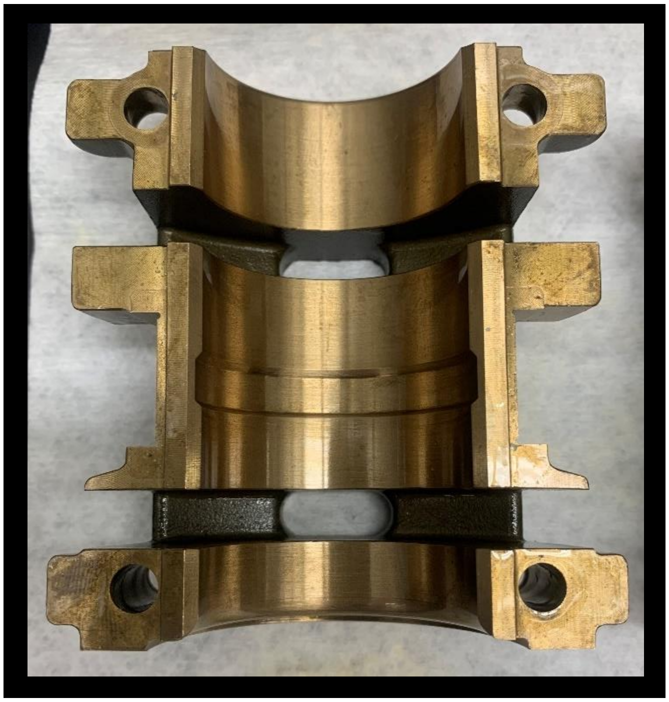


**Figure B-104:** Shown here is the governor side top bearing after the unheated, degraded oil, 3000 rpm test.

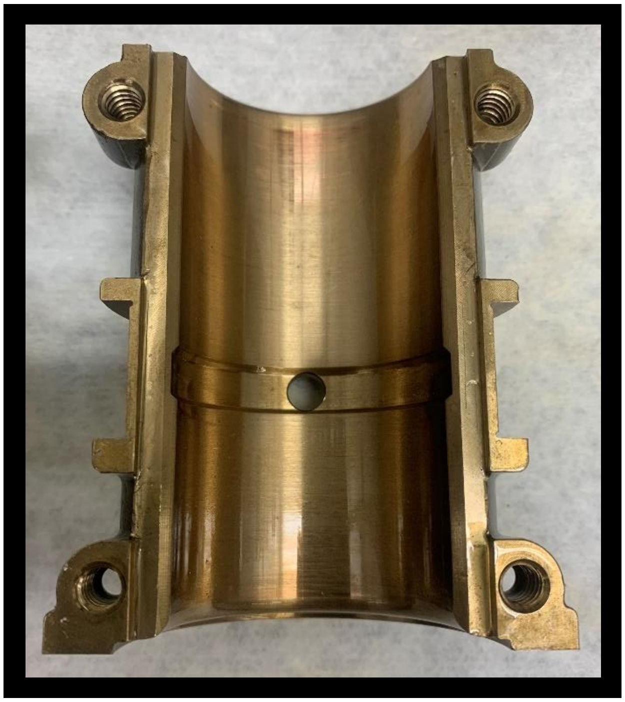


**Figure B-105:** Governor side bottom bearing after the unheated, degraded oil, 3000 rpm test.

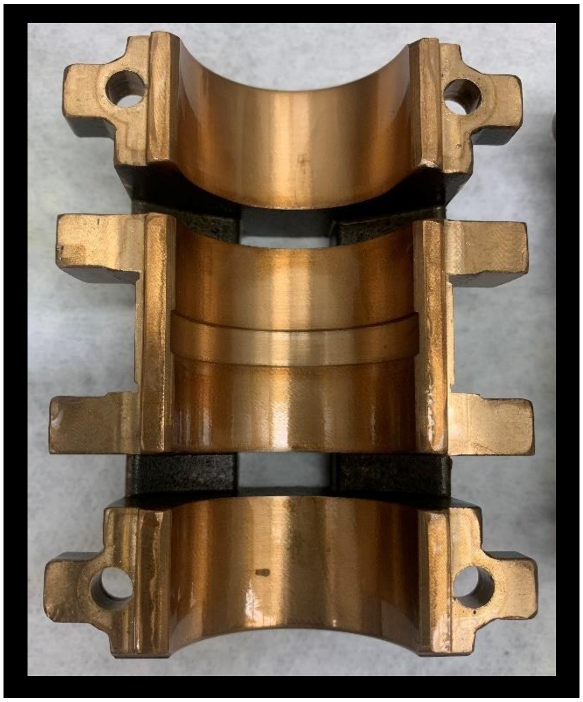


**Figure B-106:** Motor side top bearing after the unheated, degraded oil, 3000 rpm test.

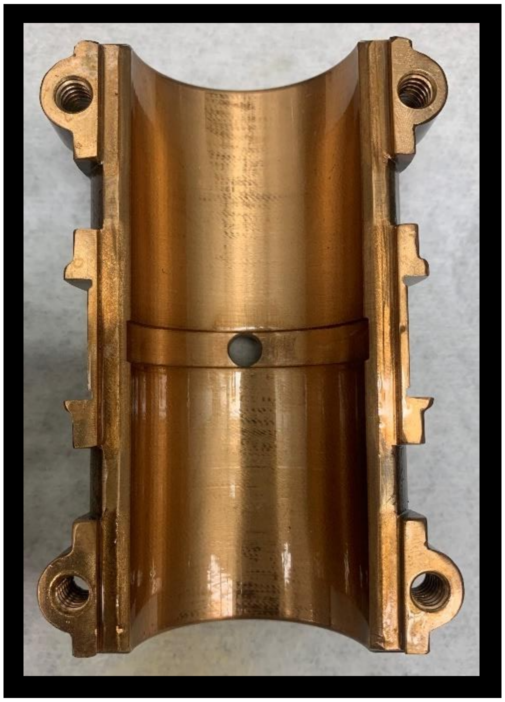


**Figure B-107:** Motor side bottom bearing after the unheated, degraded oil, 3000 rpm test.

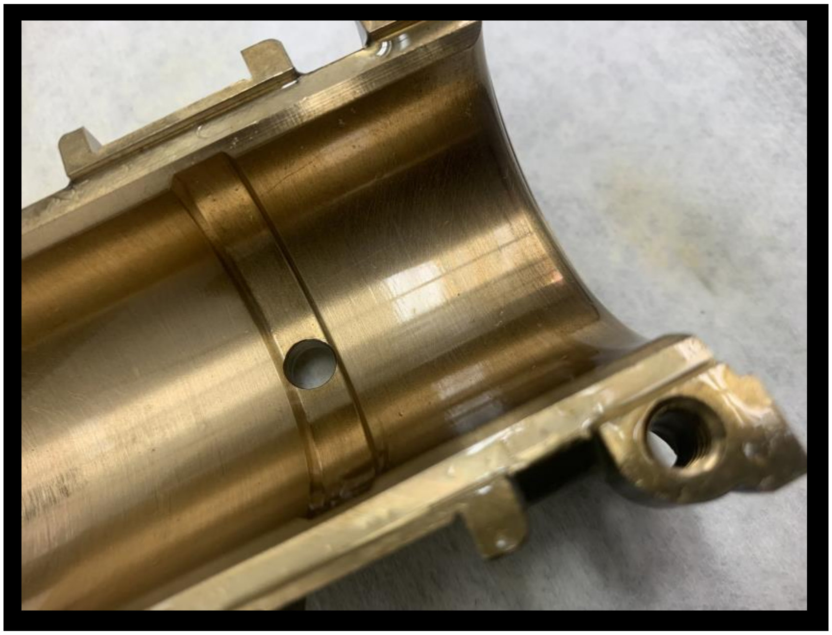


**Figure B-108:** Shown here is the governor side bottom bearing after the unheated, degraded oil, 3000 rpm test, with slight scoring.

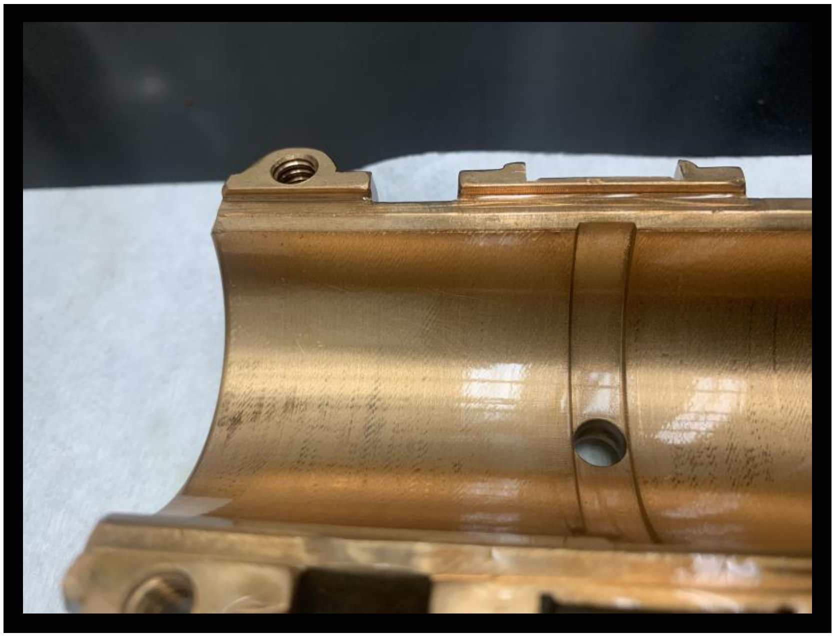


**Figure B-109:** Shown here is the motor side bottom bearing after the unheated, degraded oil, 3000 rpm test, with slight scoring.

### B.2.1.7. Degraded Oil, 121$^o$C, 1000 rpm - 01/14/2020

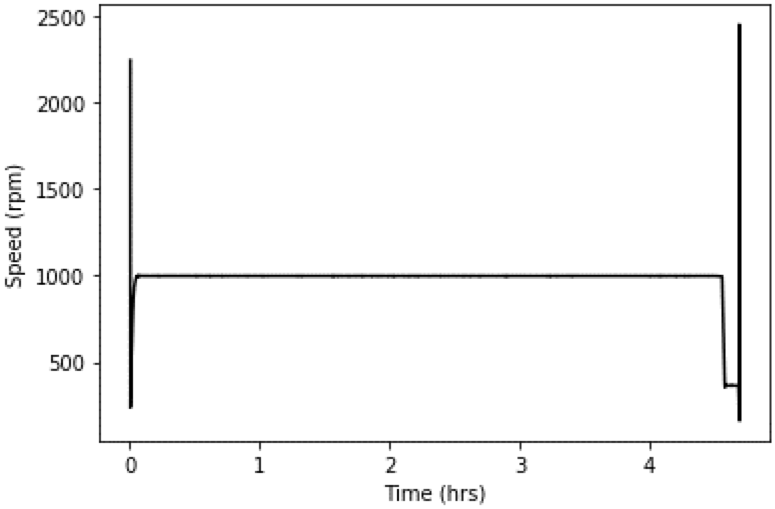


**Figure B-110:** Rotational shaft speed over time for the heated, degraded oil, 1000 rpm test.

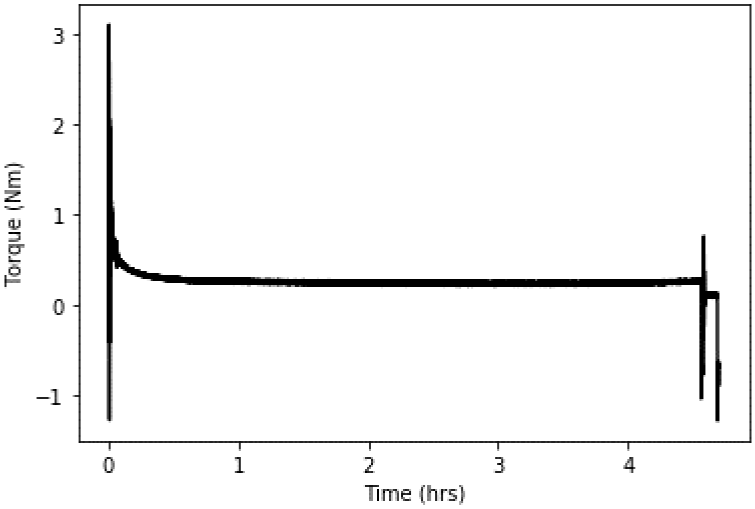


**Figure B-111:** Motor torque over time for the heated, degraded oil, 1000 rpm test.

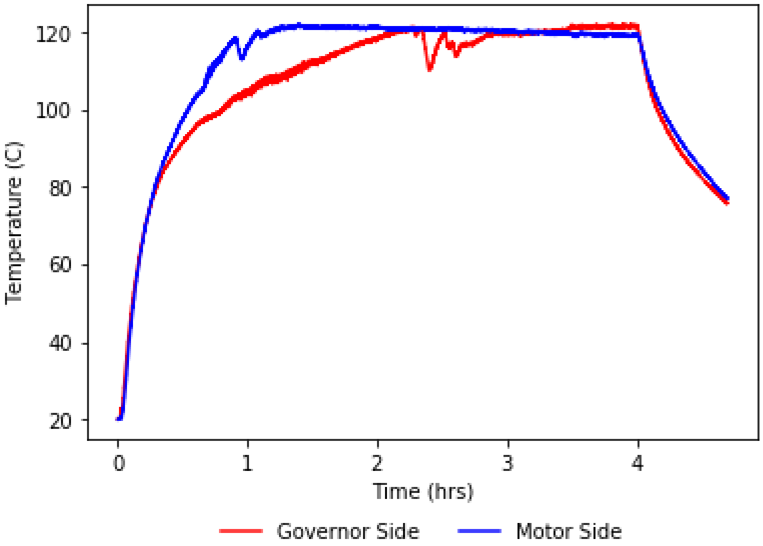


**Figure B-112:** Average oil temperature over time for the heated, degraded oil, 1000 rpm test.

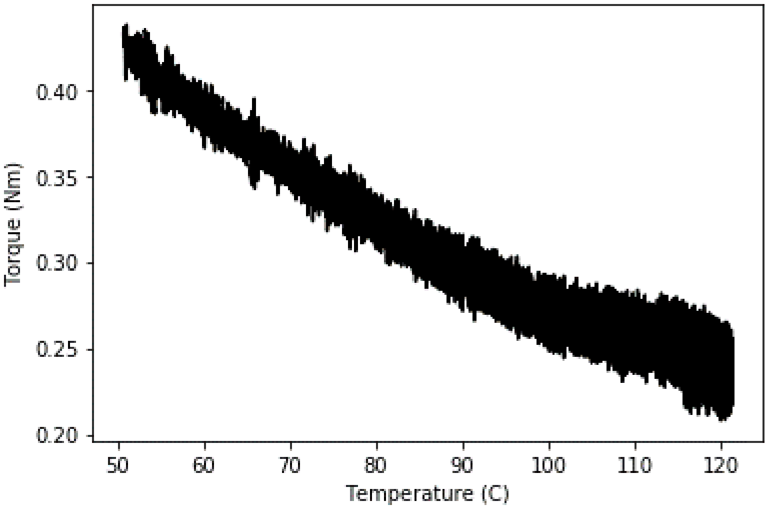


**Figure B-113:** This plot shows the relationship between motor torque and oil temperature for the heated, degraded oil, 1000 rpm test.

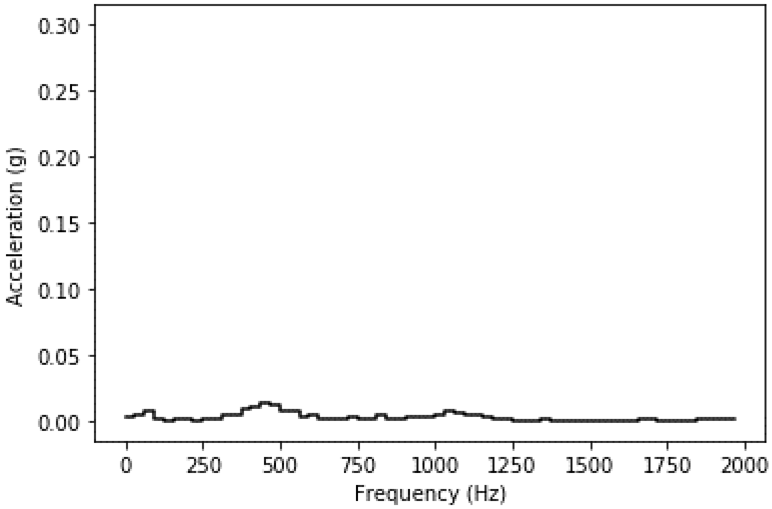


**Figure B-114:** Average acceleration magnitude per frequency bin for accelerometer 1, for the heated, degraded oil, 1000 rpm test.

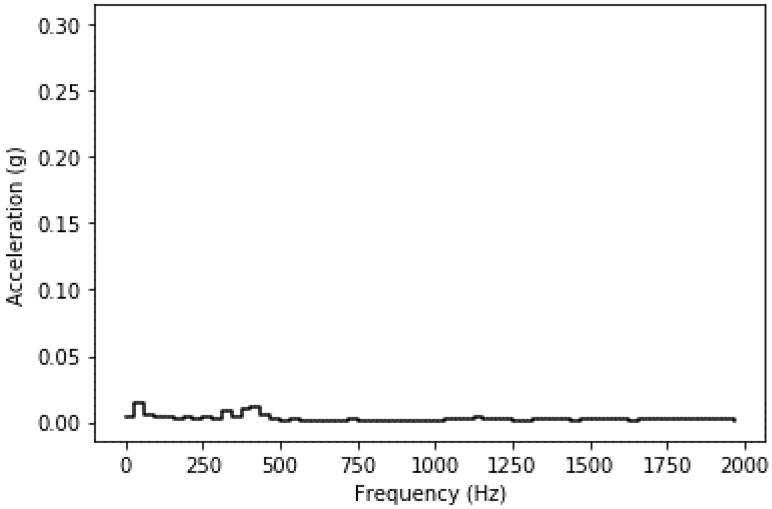


**Figure B-115:** Average acceleration magnitude per frequency bin for accelerometer 2, for the heated, degraded oil, 1000 rpm test.

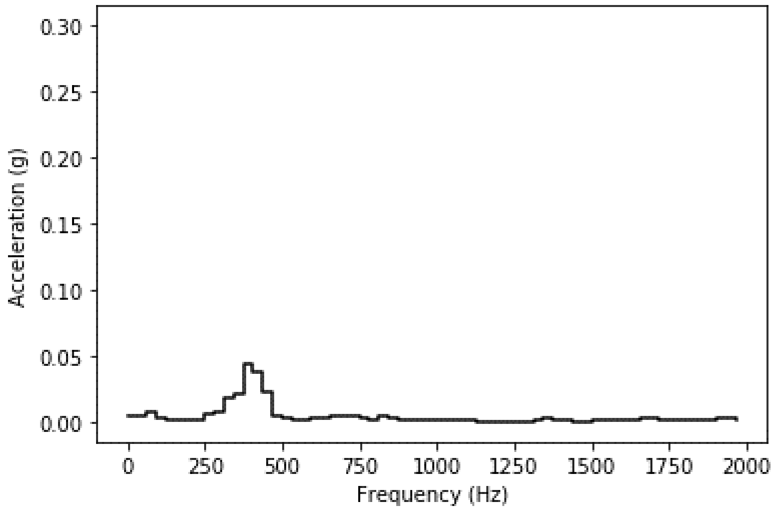


**Figure B-116:** Average acceleration magnitude per frequency bin for accelerometer 3, for the heated, degraded oil, 1000 rpm test.

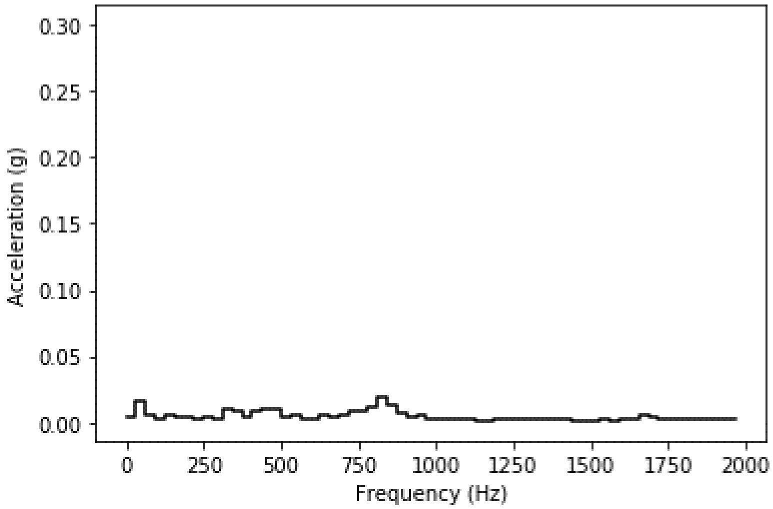


**Figure B-117:** Average acceleration magnitude per frequency bin for accelerometer 4, for the heated, degraded oil, 1000 rpm test.

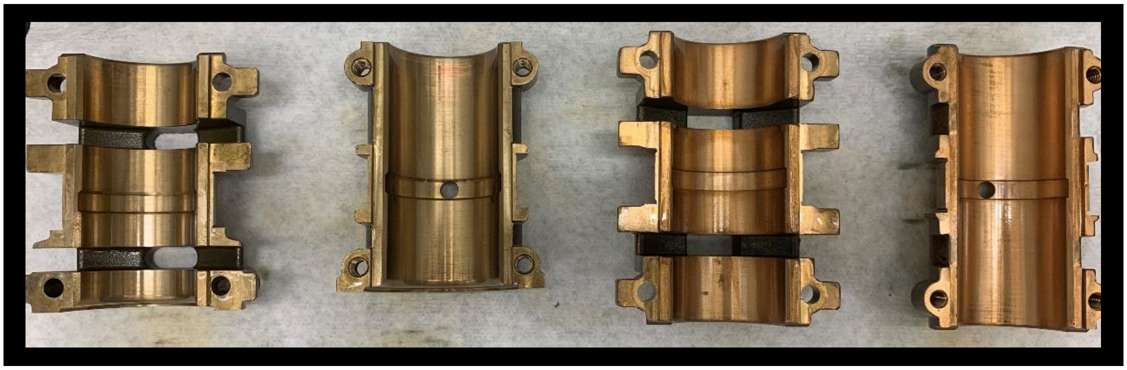


**Figure B-118:** ZS-1 journal bearings after the heated, degraded oil, 1000 rpm test.

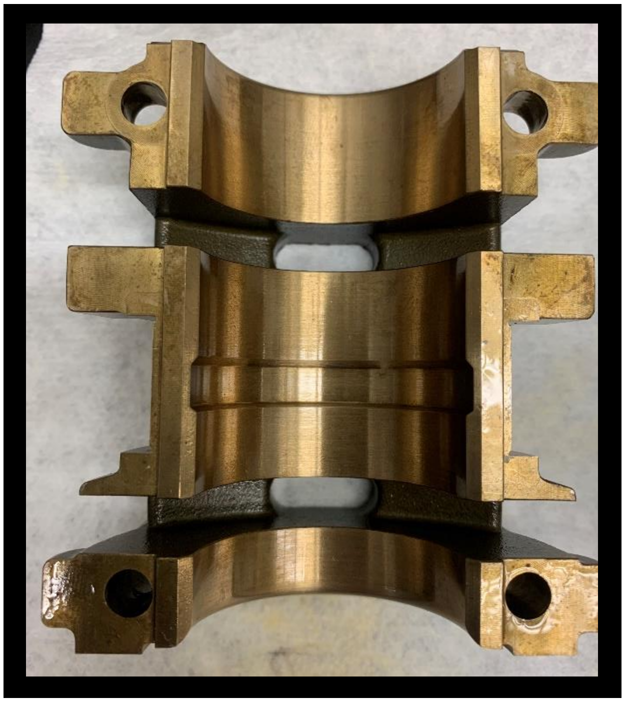


**Figure B-119:** Governor side top bearing after the heated, degraded oil, 1000 rpm test.

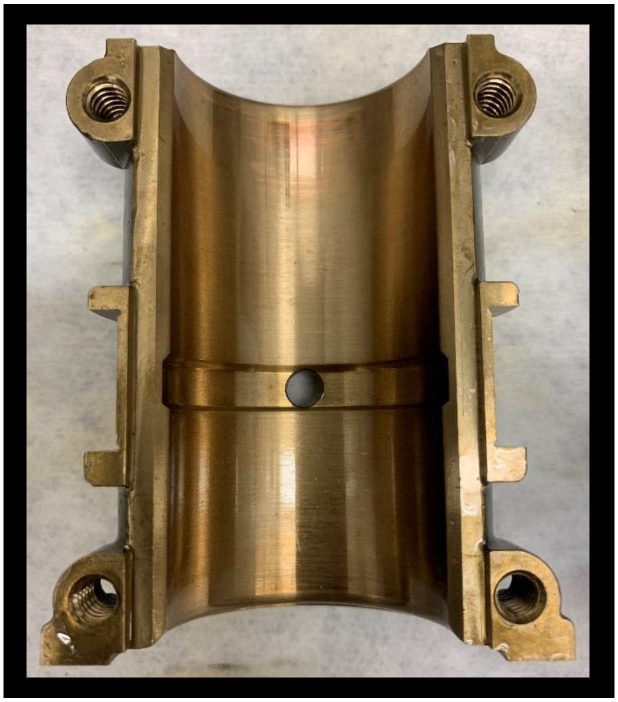


**Figure B-120:** Governor side bottom bearing after the heated, degraded oil, 1000 rpm test.

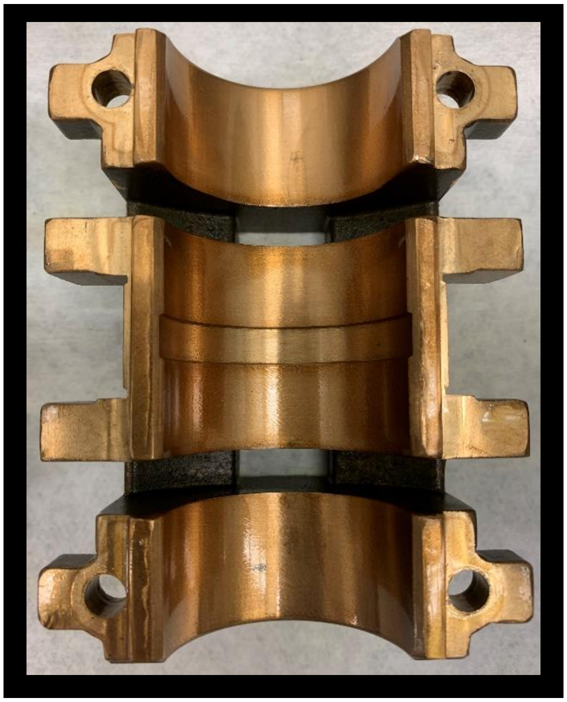


**Figure B-121:** Motor side top bearing after the heated, degraded oil, 1000 rpm test.

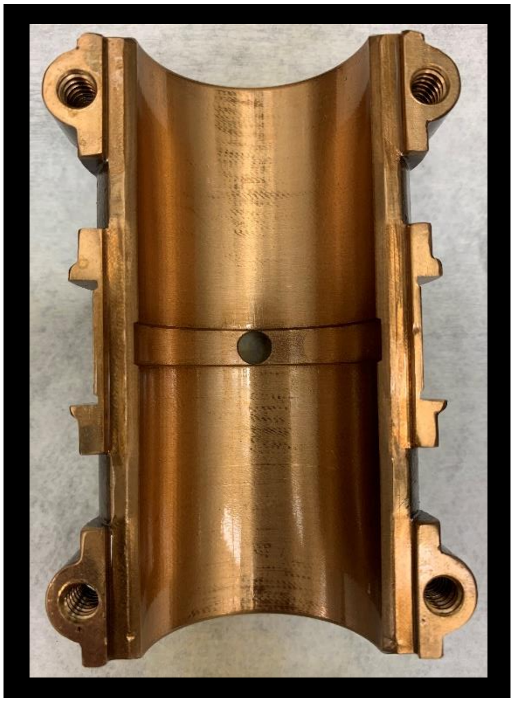


**Figure B-122:** Motor side bottom bearing after the heated, degraded oil, 1000 rpm test.

### B.2.1.8. Degraded Oil, 121ºC, 3000 rpm - 01/15/2020

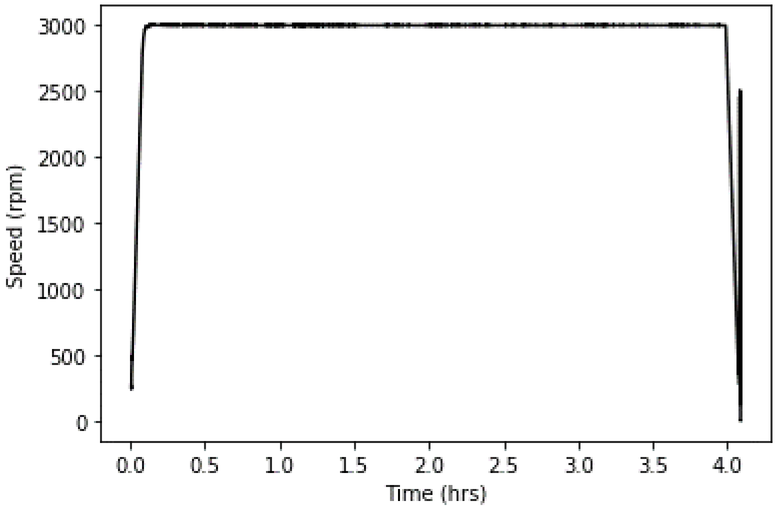


**Figure B-123:** Rotational shaft speed over time for the heated, degraded oil, 3000 rpm test.

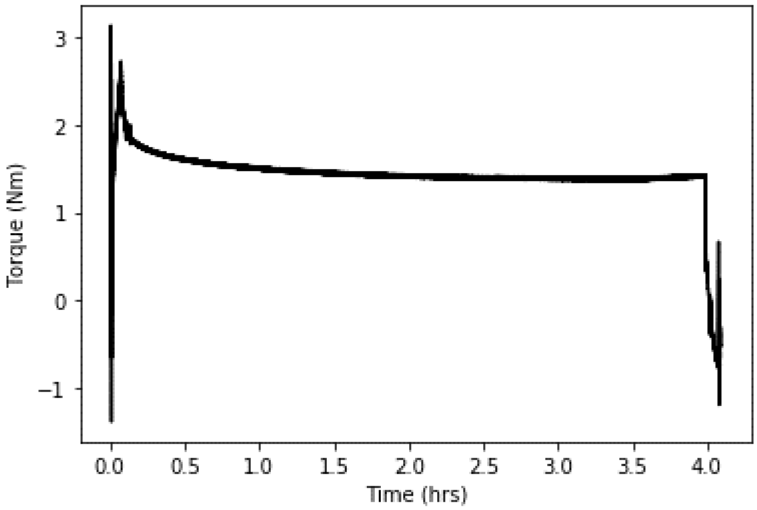


**Figure B-124:** Motor torque over time for the heated, degraded oil, 3000 rpm test.

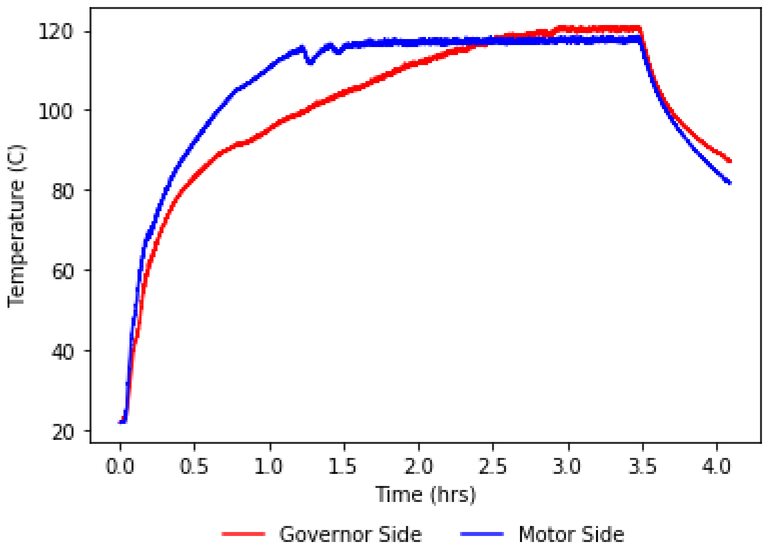


**Figure B-125:** Average oil temperature over time for the heated, degraded oil, 3000 rpm test.

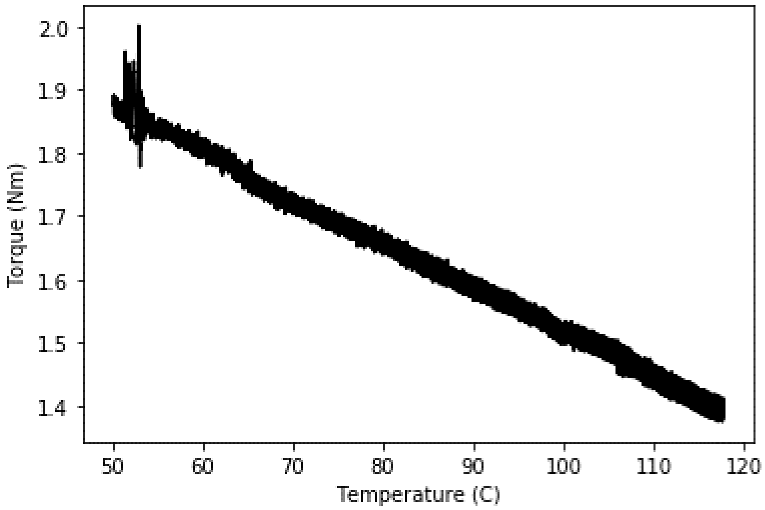


**Figure B-126:** This plot shows the relationship between motor torque and oil temperature for the heated, degraded oil, 3000 rpm test.

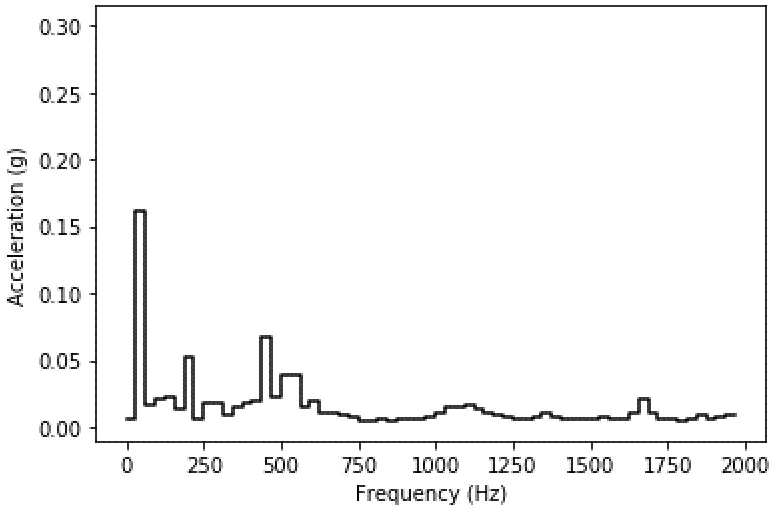


**Figure B-127:** Average acceleration magnitude per frequency bin for accelerometer 1, for the heated, degraded oil, 3000 rpm test.

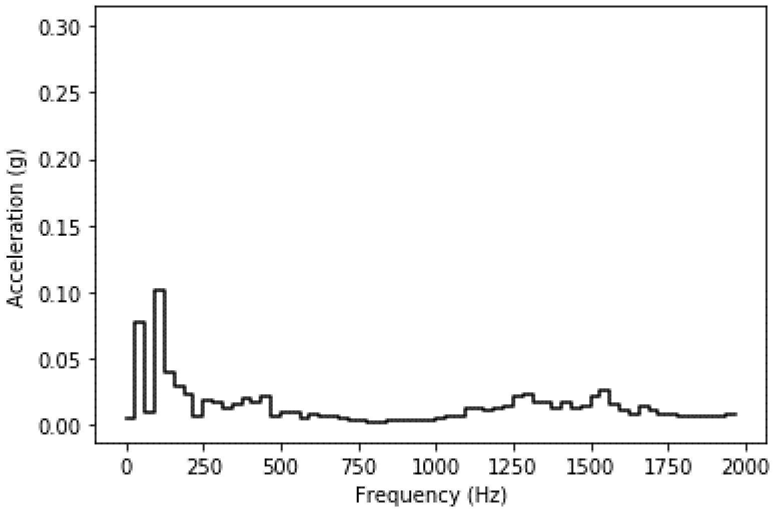


**Figure B-128:** Average acceleration magnitude per frequency bin for accelerometer 2, for the heated, degraded oil, 3000 rpm test.

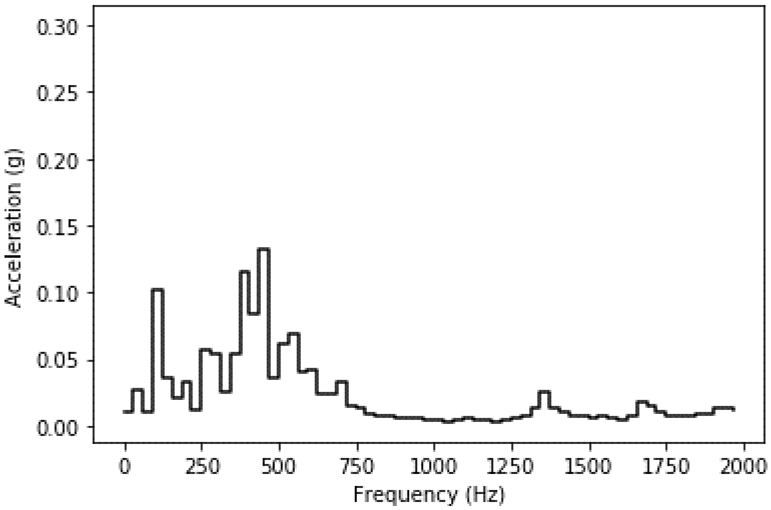


**Figure B-129:** Average acceleration magnitude per frequency bin for accelerometer 3, for the heated, degraded oil, 3000 rpm test.

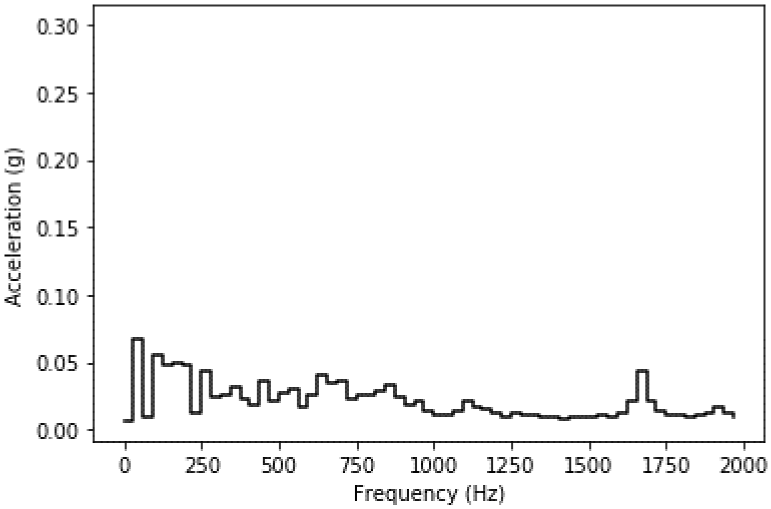


**Figure B-130:** Average acceleration magnitude per frequency bin for accelerometer 4, for the heated, degraded oil, 3000 rpm test.

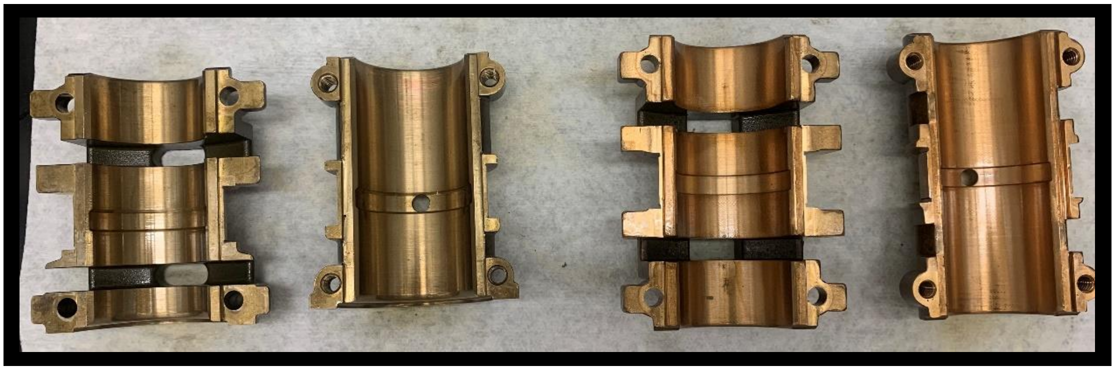


**Figure B-131:** ZS-1 journal bearings after the heated, degraded oil, 3000 rpm test.

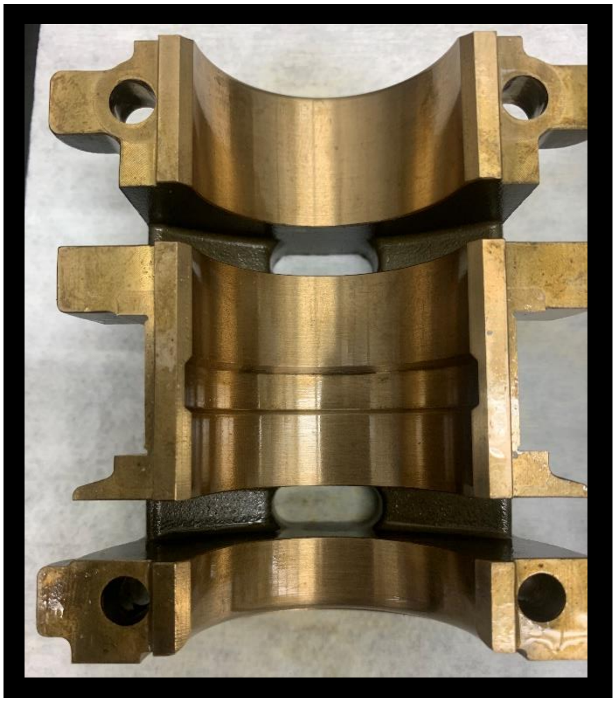


**Figure B-132:** Governor side top bearing after the heated, degraded oil, 3000 rpm test.

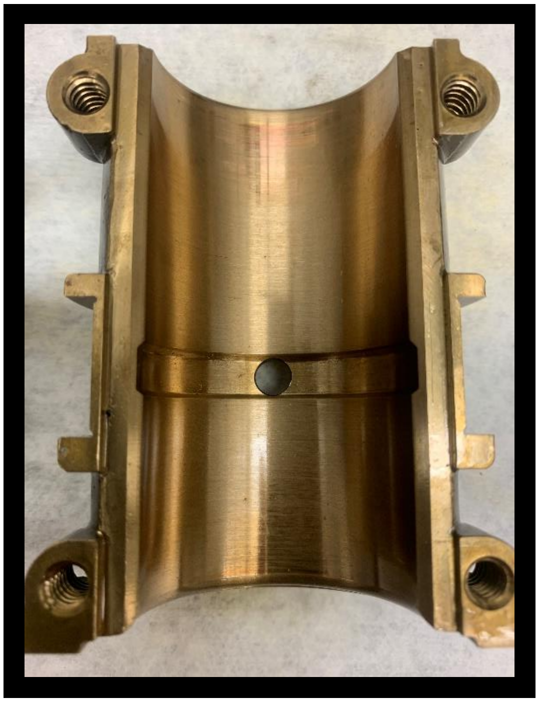


**Figure B-133:** Governor side bottom bearing after the heated, degraded oil, 3000 rpm test.

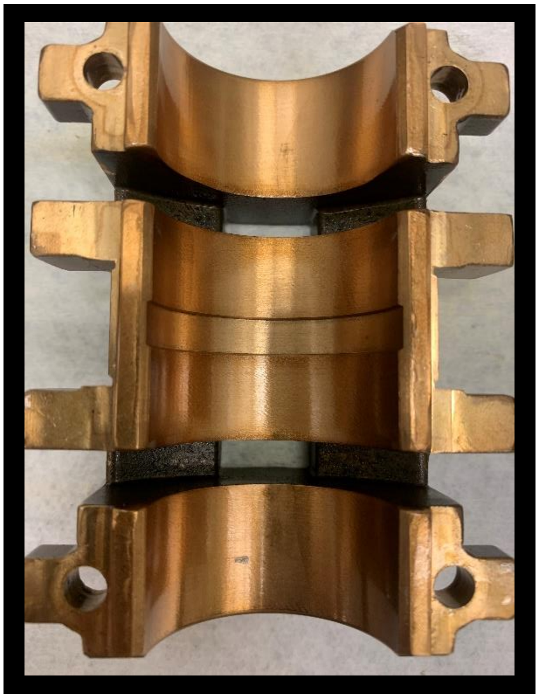


**Figure B-134:** Motor side top bearing after the heated, degraded oil, 3000 rpm test.

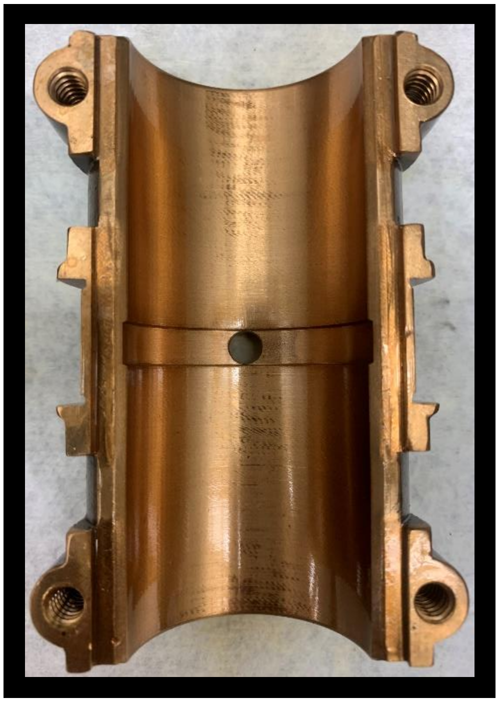


**Figure B-135:** Motor side bottom bearing after the heated, degraded oil, 3000 rpm test.

### *B.2.2. Steady State Torque*

#### B.2.2.1. Fresh Oil, No Heat - 10/29/2019

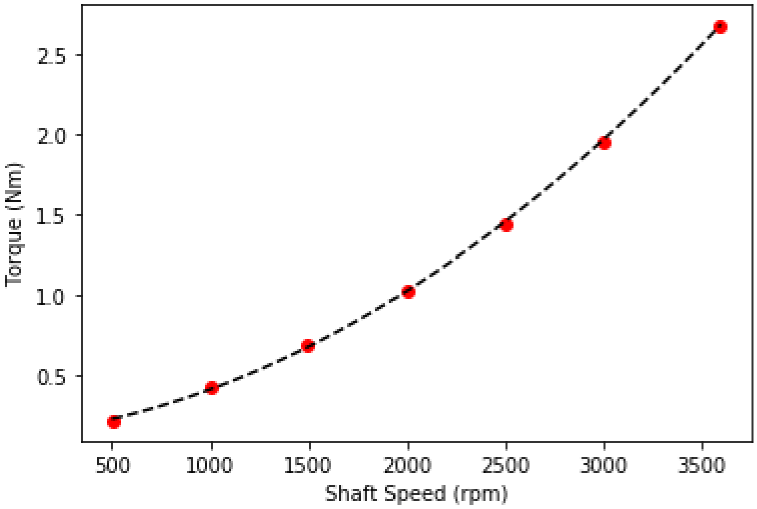


**Figure B-136:** This plot shows the relationship between speed and torque for forward spin only, for the unheated, fresh oil test. The equation for this line is 1.629e-7$x^2$ + 1.25e-4x + 0.127, with an $R^2$ of 0.999.

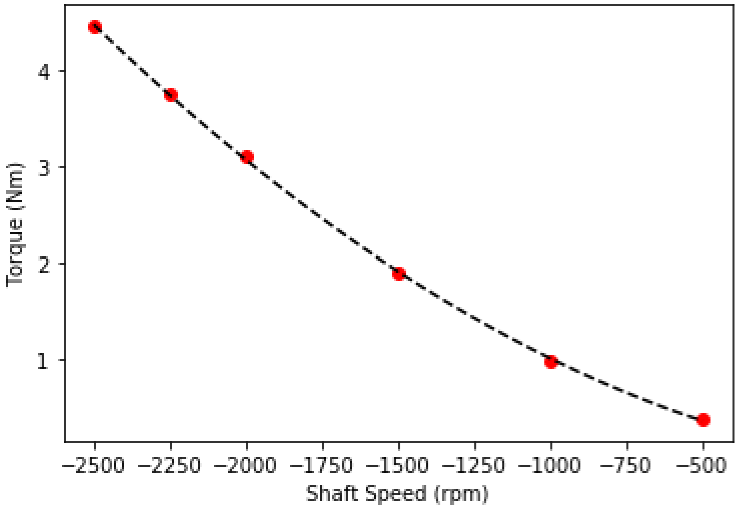


**Figure B-137:** This plot shows the relationship between speed and torque for reverse spin only, for the unheated, fresh oil test. The equation for this line is 5.058e-7x$^2$ – 5.452e-4x + 0.045, with an R$^2$ of 0.999.

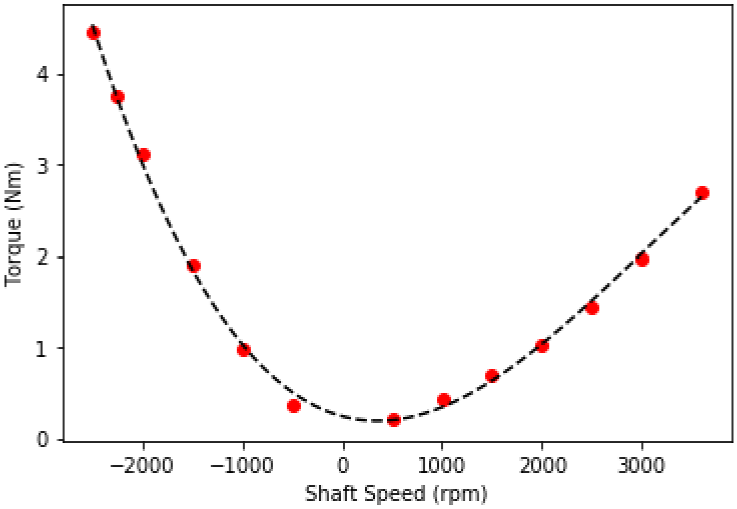


**Figure B-138:** This plot shows the full curve relationship between speed and torque, for the unheated, fresh oil test. The equation for this line is -4.933e-11x$^3$ + 4.451e-7x$^2$ – 2.963e-4x + 0.241, with an R$^2$ of 0.997.

### B.2.2.2. Fresh Oil, 121$^{o}$C - 11/26/2019

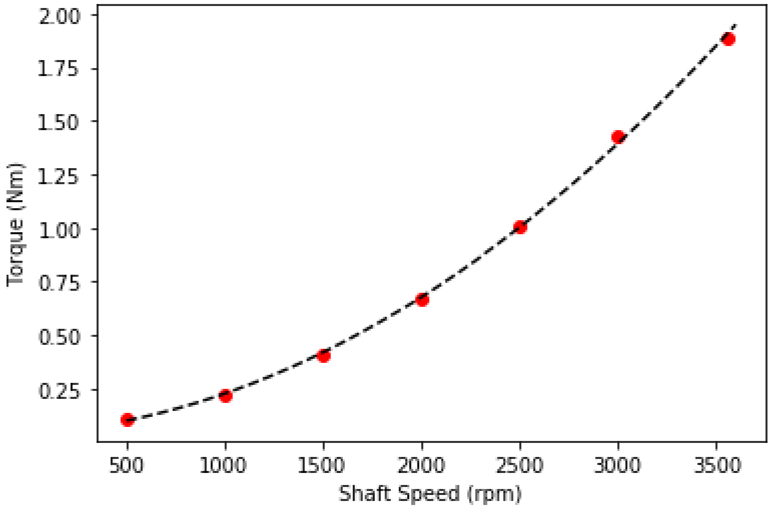


**Figure B-139:** This plot shows the relationship between speed and torque for forward spin only, for the heated, fresh oil test. The equation for this line is 1.323e-7$x^2$ + 5.442e-5x + 0.039, with an $R^2$ of 0.999.

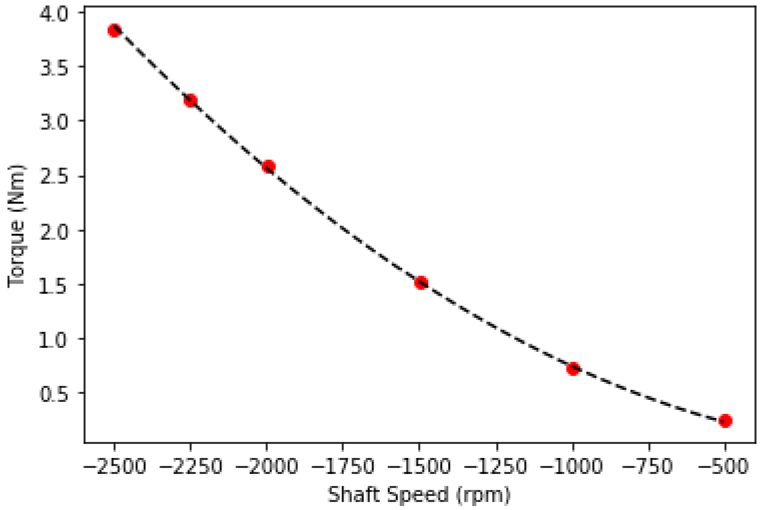


**Figure B-140:** This plot shows the relationship between speed and torque for reverse spin only, for the heated, fresh oil test. The equation for this line is 5.29e-7x$^2$ – 2.321e-4x – 0.019, with an R$^2$ of 0.999.

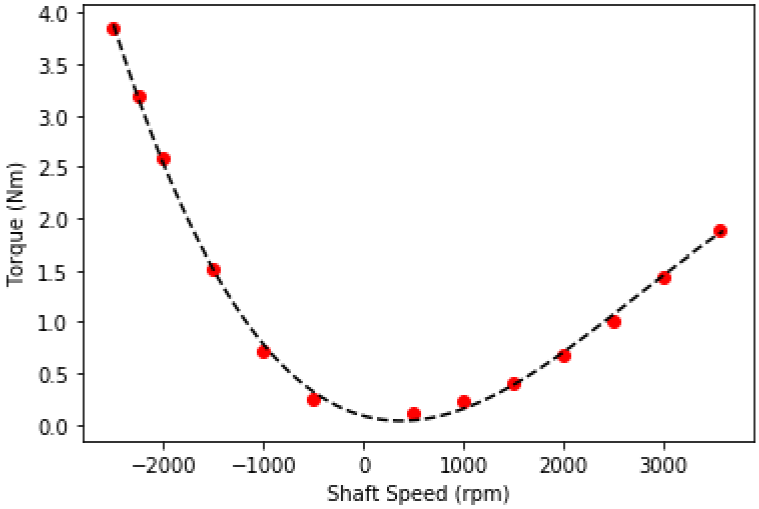


**Figure B-141:** This plot shows the full curve relationship between speed and torque, for the heated, fresh oil test. The equation for this line is -4.738e-11x$^3$ + 3.829e-7x$^2$ – 2.673e-4x + 0.083, with an R$^2$ of 0.998.

### B.2.2.3. Degraded Oil, No Heat - 01/07/2020

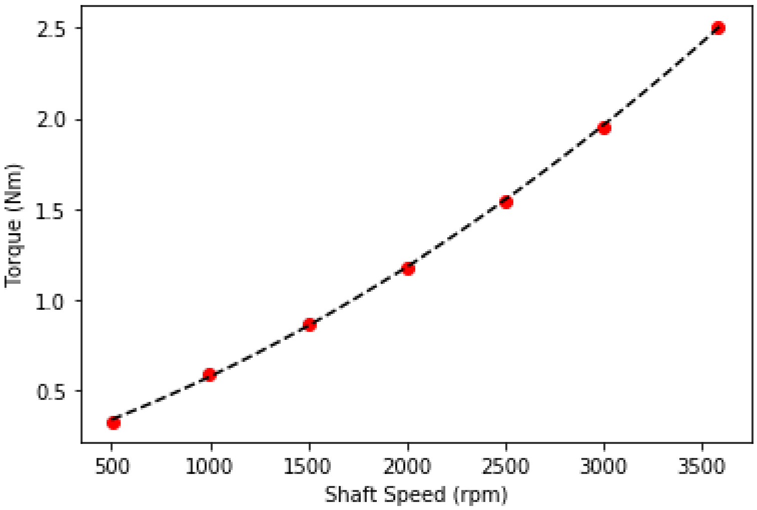


**Figure B-142:** This plot shows the relationship between speed and torque for forward spin only, for the unheated, degraded oil test. The equation for this line is 8.743e-8$x^2$ + 3.438e-4x + 0.142, with an $R^2$ of 0.999.

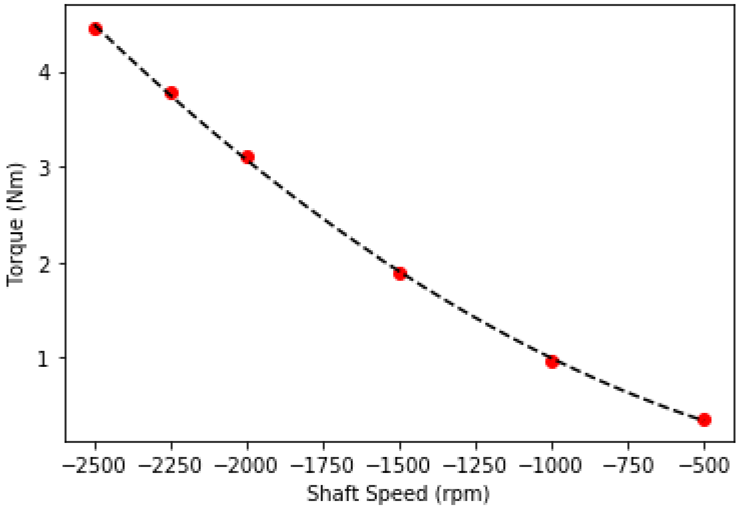


**Figure B-143:** This plot shows the relationship between speed and torque for reverse spin only, for the unheated, degraded oil test. The equation for this line is 5.085e-7x$^2$ – 5.519-4x + 0.067, with an R$^2$ of 0.999.

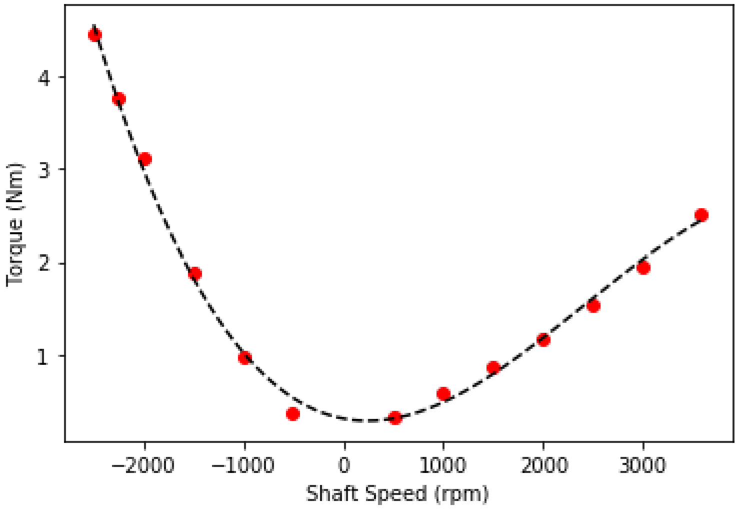


**Figure B-144:** This plot shows the full curve relationship between speed and torque, for the unheated, degraded oil test. The equation for this line is -6.157e-11x$^3$ + 4.449e-7x$^2$ – 2.065e-4x + 0.307, with an R$^2$ of 0.995.

### B.2.2.4. Degraded Oil, 121°C - 01/15/2020

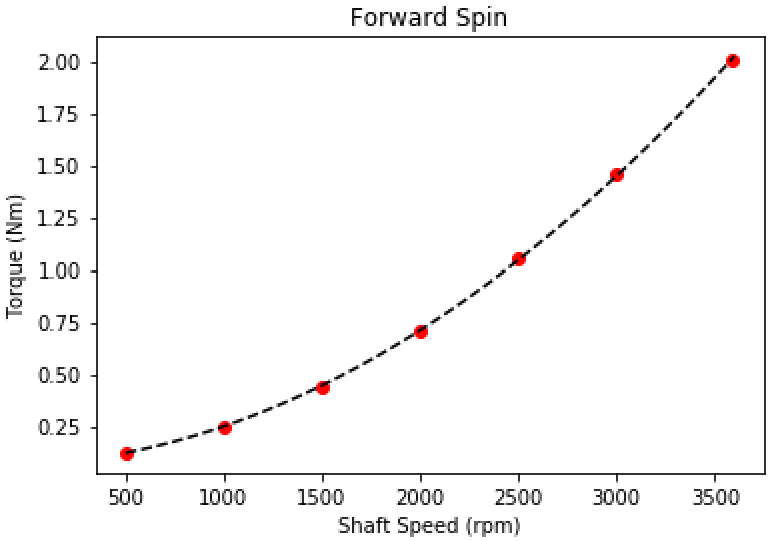


**Figure B-145:** This plot shows the relationship between speed and torque for forward spin only, for the heated, degraded oil test. The equation for this line is 1.376e-7x$^2$ + 4.877e-5x + 0.063, with an R$^2$ of 0.999.

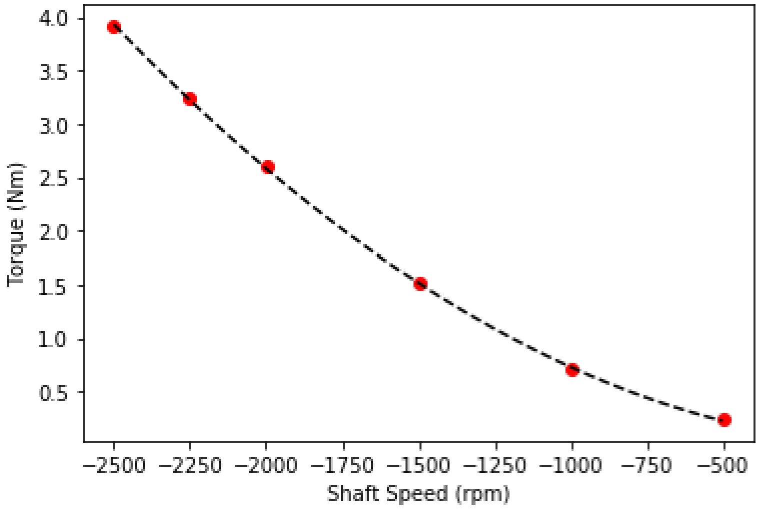


**Figure B-146:** This plot shows the relationship between speed and torque for reverse spin only, for the heated, degraded oil test. The equation for this line is 5.625e-7x$^2$ – 1.668e-4x – 8.637e-4, with an R$^2$ of 0.999.

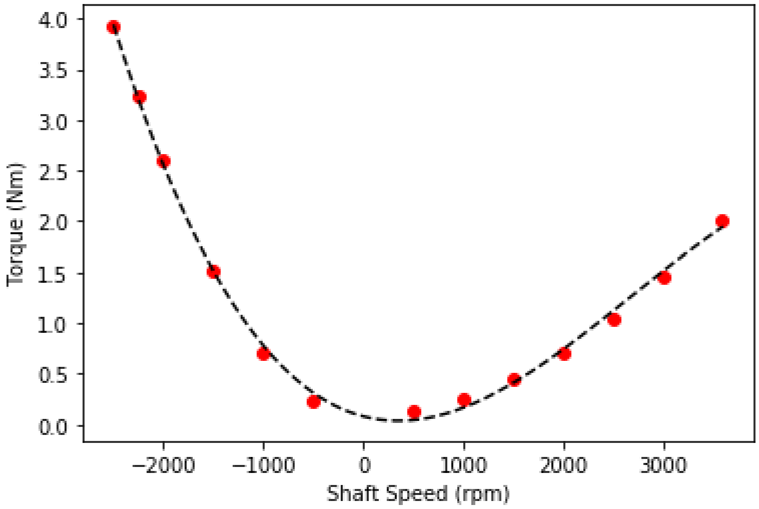


**Figure B-147:** This plot shows the full curve relationship between speed and torque, for the heated, degraded oil test. The equation for this line is -4.898e-11x$^3$ + 3.92e-7x$^2$ – 2.578e-4x + 0.08, with an R$^2$ of 0.997.

# APPENDIX C

# OPERATING PROCEDURES

## C.1. Oil Degradation Tests

1. Ensure that the beakers, hot plates, thermocouples, fume hood area, and viscometer are clean.
2. Fill two stainless steel beakers with 2500mL of DTE-732 turbine oil, each.
3. Arrange both hotplates side by side, underneath the fume hood.
4. Place the beakers on the center of the hot plates.
5. Arrange the viscometer such that the temperature and viscosity probes are both submerged in the oil, and centered with the beaker and hot plate.
6. Insert three thermocouples into each beaker at a high, middle, and low elevation in the oil.
7. Wrap both beakers loosely with two layers of insulation; seal the seams with insulation tape.
8. Turn on the power supply for the SCXI Chassis and then power on the SCXI Chassis.
9. Reboot the SCXI Chassis’ driver.
10. Open the LabVIEW file “Oil Facility”.
11. Select “yes” for “write to file”, and edit the filename to reflect the current date and test number.
12. Input the target temperature in Celsius in the “Desired Temperature” field.
13. Plug the hot plates in to different outlets on the fume hood, and turn them on to an appropriate temperature.
14. Lower the fume hood shield to the marked location (about 4in open).
15. On the computer, click “run” until the timer starts, and then select “start recording”.
16. As the oil reaches the target temperature, slowly lower the hot plate temperature to prevent a temperature overshoot.
17. Make a note of when the desired temperature was reached, and begin the 72-hour timer here.

18. Monitor the temperature of the oil closely throughout the duration of the experiment, and adjust the temperature of the hot plates accordingly; when changing the temperature of the hotplates, make very slight adjustments.
19. Once the 72 hours at temperature have elapsed, turn off and unplug the hot plates, and remove both layers of insulation.
20. Once the oil reaches room temperature, in LabVIEW, select “stop recording” and then “end execution”.
21. Turn off the power supply and SCXI Chassis, and safely store the degraded oil.

## C.2. Heated Bearing Tests

### *C.2.1. Bearing Inspection*

1. Ensure that the VFD is powered off before proceeding.
2. Check the oil levels in the bubblers and refill as needed.
3. Unscrew the top four 1/2 in horizontally oriented governor side bolts.
4. Governor Side Oil Well:
    a. Remove the two dowel pins.
    b. Unscrew the six 3/4 in bolts.
    c. Remove the oil well cover.
5. Motor Side Oil Well:
    a. Remove the two dowel pins.
    b. Unscrew the two 3/4 in bolts.
    c. Remove the oil well cover.
6. Governor Side Bearings:
    a. Unscrew the four 3/16 in hex bolts.
    b. Remove the top bearing.
    c. While lifting the shaft, rotate the bottom bearing until it is on top of the shaft.
    d. Move the slinger rings to the side.
    e. Remove the bottom bearing.
7. Motor Side Bearings:

    a. Remove the shaft guard.
    b. Unscrew the four 3/16 in hex bolts.
    c. Remove the top bearing.
    d. While lifting the shaft (grip the wheel side coupling, not the motor side), rotate the bottom bearing until it is on top of the shaft.
    e. Move the slinger rings to the side.
    f. Remove the bottom bearing.

8. Wipe the bearings clean of oil using a lint free rag.
9. Take pictures of all four bearings, and document changes (scratches, marks, discoloration, etc.)
10. Motor Side Bearings:
    a. Place the bottom bearing on the shaft, number towards the wheel.
    b. While lifting the shaft (grip the wheel side coupling, not the motor side), rotate the bottom bearing until it is beneath the shaft.
    c. Move the slinger rings into place.
    d. Place the top bearing on the shaft, logo towards the wheel.
    e. Screw in the four 3/16 in hex bolts.
    f. Manually rotate the shaft to verify proper installation; if the shaft doesn’t spin, something is wrong.
11. Governor Side Bearings:
    a. Place the bottom bearing on the shaft, large flanged side away from the wheel.
    b. While lifting the shaft, rotate the bottom bearing until it is beneath the shaft.
    c. Move the slinger rings into place.
    d. Place the top bearing on the shaft, large flanged side away from the wheel.
    e. Screw in the four 3/16 in hex bolts.
    f. Manually rotate the shaft to verify proper installation; if the shaft doesn’t spin, something is wrong.
12. Motor Side Oil Well:
    a. Put the oil well cover back into place.
    b. Tap in the two dowel pins.

    c. Screw in the two 3/4 in bolts.
    d. Bolt down the shaft guard.

13. Governor Side Oil Well:
    a. Put the oil well cover back into place.
    b. Tap in the two dowel pins.
    c. Screw in the six 3/4 in bolts.
14. Screw in the top four 1/2 in horizontally oriented governor side bolts.

### *C.2.2. Bearing Test*

1. Perform a bearing inspection.
2. Ensure the shaft guard is bolted down in place.
3. Plug in the tachometer and heater controller, and turn the heater controller on.
4. Turn on the NI PXIe-1075 power supply, and power on the NI PXIe-1075.
5. Plug in the hygrometer and remove the protective cap.
6. Power on the VFD by flipping the 61, 63, and 65 breaker switches to “on”.
7. Open the LabVIEW project “Turbine Project”:
    a. Under “My Computer”, select “VIs” and open “PC Main 1.6”.
    b. Under “NI-PXIe8821-2F178D9F”, select “VIs” and open “RT Main 1.4”.
8. In the “PC Main 1.6” window:
    a. Input the target temperature in Celsius in the “Target Temperature” field.
    b. Input the target shaft speed in rpm in the “Desired Speed” field.
    c. Edit the file name.
    d. Change “Write to File” from “No” to “Yes”.
9. In the “RT Main 1.4” window, select “Run”.
10. In the “PC Main 1.6” window, select “Run”.
11. On the VFD controller, press “Run”.

*Steps 12, 13, 14, and 15 are to be used in conjunction with a Bluetooth enabled mobile device and the Watlow “EZ-Link” app. They can be done using just the controllers themselves, but the procedure is much more complicated. Additionally, if the test does not require heat, these steps can be ignored.*

12. Connect to the controller “1 Governor Side”:
    a. Set the control mode to “Auto”.
    b. Set the active set point to the desired temperature.
    c. Disconnect from the controller.
13. Connect to the controller “1 Motor Side”:
    a. Set the control mode to “Auto”.
    b. Set the active set point to the desired temperature.
    c. Disconnect form the controller.
14. Once the test time has been reached, connect to the controller “1 Governor Side”:
    a. Set the control mode to “off”.
    b. Disconnect from the controller.
15. Connect to the controller “1 Motor Side”:
    a. Set the control mode to “off”.
    b. Disconnect from the controller.
16. Once the oil temperatures have fallen to 80$^{o}$C, in the “PC Main 1.6” window, change the desired speed to 0 rpm.
17. When the shaft speed reaches approximately 500 rpm, press “Stop/Rst” on the VFD controller.
18. When the shaft stops, in the “PC Main 1.6” window, select “Click to Stop” and then “Stop Run”.
19. Power off the VFD by flipping the 61, 63, and 65 breaker switches to “off”.
20. Power down the NI PXIe-1075, and turn off its power supply.
21. Unplug the hygrometer and put the protective cap back on the probe.
22. Turn off the heater controller, and unplug it and the tachometer.
23. Perform a bearing inspection.

# APPENDIX D

# PYTHON SCRIPTS

## D.1. Oil Data Reader

*This script is to be used to read and organize the data file produced from oil degradation tests.*

```
import numpy as np
import matplotlib.pyplot as plt



file_name = '2019_03_07_test1.dat'
dat_water = np.loadtxt(file_name)

file_name = '2019_02_18_72hr.dat'
dat_121_1 = np.loadtxt(file_name)

file_name = '2019_07_08_test1.dat'
dat_121_2 = np.loadtxt(file_name)
##############################################################################
#10% Wet Test

time_water = dat_water[: , 0]
TC1_water = dat_water[:, 7]
TC2_water = dat_water[:, 9]
TC3_water = dat_water[:, 11]
V_water = dat_water[:, 25]

Avg_water = np.mean(np.array([TC1_water, TC2_water, TC3_water]), axis = 0)

V_water_F = []
```

```
T_water_F = []
i = 0
while (Avg_water[i] < 93):
    V_water_F.append(V_water[i])
    T_water_F.append(Avg_water[i])
    i += 1

for i in reversed(range(len(Avg_water))):
    if Avg_water[i] >= 93:
        i = i
        break
V_water_D = V_water[i::]
T_water_D = Avg_water[i::]

plt.figure(1)
plt.clf()
plt.plot(time_water/3600, TC1_water, 'b-', label = 'Top')
plt.plot(time_water/3600, TC2_water, 'k-', label = 'Bottom')
plt.plot(time_water/3600, TC3_water, 'r-', label = 'Middle')
plt.xlabel('Time (hrs)')
plt.ylabel('Temperature (C)')
plt.title('Temperature Gradient')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 3, frameon = False)
plt.show()

plt.figure(2)
plt.clf()
plt.plot(time_water/3600, V_water, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Viscosity (cSt)')
plt.title('Viscosity Over Time')
plt.show()
plt.figure(3)
plt.clf()
plt.plot(T_water_F, V_water_F, 'k-', label = 'Fresh')
plt.plot(T_water_D, V_water_D, 'r-', label = 'Degraded')
plt.xlabel('Temperature (C))')
plt.ylabel('Viscosity (cSt)')
plt.title('Degraded vs Fresh Oil')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 2, frameon = False)
plt.show()
#####################################################################################
#121 C - First Test

time_121_1 = dat_121_1[: , 0]
TC1_121_1 = dat_121_1[:, 1]
TC2_121_1 = dat_121_1[:, 3]
TC3_121_1 = dat_121_1[:, 5]
TC4_121_1 = dat_121_1[:, 7]
TC5_121_1 = dat_121_1[:, 9]
TC6_121_1 = dat_121_1[:, 11]
V_121_1 = dat_121_1[:, 25]

Avg_121_1_L = np.mean(np.array([TC1_121_1, TC2_121_1, TC3_121_1]), axis = 0)
Avg_121_1_R = np.mean(np.array([TC4_121_1, TC5_121_1, TC6_121_1]), axis = 0)

V_121_1_F = []
T_121_1_F = []
i = 0
while (Avg_121_1_R[i] < 93):
   V_121_1_F.append(V_121_1[i])
   T_121_1_F.append(Avg_121_1_R[i])
   i += 1

for i in reversed(range(len(Avg_121_1_R))):
    if Avg_121_1_R[i] >= 121:
        i = i
        break
V_121_1_D = V_121_1[i::]
T_121_1_D = Avg_121_1_R[i::]

plt.figure(4)
plt.clf()
plt.plot(time_121_1/3600, Avg_121_1_L, 'k-', label = 'Left Beaker')
plt.plot(time_121_1/3600, Avg_121_1_R, 'r-', label = 'Right Beaker')
plt.xlabel('Time (hrs)')
plt.ylabel('Temperature (C)')
plt.title('Temperature Averages')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 2, frameon = False)
plt.show()

plt.figure(5)
plt.clf()
plt.plot(time_121_1/3600, V_121_1, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Viscosity (cSt)')
plt.title('Viscosity Over Time')
plt.show()

plt.figure(6)
plt.clf()
plt.plot(T_121_1_F, V_121_1_F, 'k-', label = 'Fresh')
plt.plot(T_121_1_D, V_121_1_D, 'r-', label = 'Degraded')
plt.xlabel('Temperature (C))')
plt.ylabel('Viscosity (cSt)')
plt.title('Degraded vs Fresh Oil')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 2, frameon = False)
plt.show()
```

```
#####################################################################################
#121 C - Second Test

time_121_2 = dat_121_2[: , 0]
TC1_121_2 = dat_121_2[:, 1]
TC2_121_2 = dat_121_2[:, 3]
TC3_121_2 = dat_121_2[:, 5]
TC4_121_2 = dat_121_2[:, 7]
TC5_121_2 = dat_121_2[:, 9]
TC6_121_2 = dat_121_2[:, 11]
V_121_2 = dat_121_2[:, 25]

Avg_121_2_L = np.mean(np.array([TC1_121_2, TC2_121_2, TC3_121_2]), axis = 0)
Avg_121_2_R = np.mean(np.array([TC4_121_2, TC5_121_2, TC6_121_2]), axis = 0)

V_121_2_F = []
T_121_2_F = []
i = 0
while (Avg_121_2_R[i] < 93):
    V_121_2_F.append(V_121_2[i])
    T_121_2_F.append(Avg_121_2_R[i])
    i += 1

for i in reversed(range(len(Avg_121_2_R))):
    if Avg_121_2_R[i] >= 121:
        i = i
        break
V_121_2_D = V_121_2[i::]
T_121_2_D = Avg_121_2_R[i::]

plt.figure(7)
plt.clf()
plt.plot(time_121_2/3600, Avg_121_2_L, 'k-', label = 'Left Beaker')
plt.plot(time_121_2/3600, Avg_121_2_R, 'r-', label = 'Right Beaker')
plt.xlabel('Time (hrs)')
plt.ylabel('Temperature (C)')
plt.title('Temperature Averages')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 2, frameon = False)
plt.show()

plt.figure(8)
plt.clf()
plt.plot(time_121_2/3600, V_121_2, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Viscosity (cSt)')
plt.title('Viscosity Over Time')
plt.show()

plt.figure(9)
plt.clf()
plt.plot(T_121_2_F, V_121_2_F, 'k-', label = 'Fresh')
plt.plot(T_121_2_D, V_121_2_D, 'r-', label = 'Degraded')
plt.xlabel('Temperature (C))')
plt.ylabel('Viscosity (cSt)')
plt.title('Degraded vs Fresh Oil')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 2, frameon = False)
plt.show()
#################################################################################
#Viscosity Comparison

plt.figure(10)
plt.clf()
plt.plot(T_water_F, V_water_F, 'b-', label = 'Fresh - Wet')
plt.plot(T_water_D, V_water_D, 'g-', label = 'Degraded - Wet')
plt.plot(T_121_1_F, V_121_1_F, 'k-', label = 'Fresh - Dry')
plt.plot(T_121_1_D, V_121_1_D, 'r-', label = 'Degraded - Dry')
plt.xlabel('Temperature (C))')
plt.ylabel('Viscosity (cSt)')
plt.title('Wet vs Dry Oil')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 2, frameon = False)
plt.show()

plt.figure(11)
plt.clf()
plt.plot(T_121_1_F, V_121_1_F, 'b-', label = 'Fresh - Glass')
plt.plot(T_121_1_D, V_121_1_D, 'g-', label = 'Degraded - Glass')
plt.plot(T_121_2_F, V_121_2_F, 'k-', label = 'Fresh - Steel')
plt.plot(T_121_2_D, V_121_2_D, 'r-', label = 'Degraded - Steel')
plt.xlabel('Temperature (C))')
plt.ylabel('Viscosity (cSt)')
plt.title('Glass vs Steel')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 2, frameon = False)
plt.show()
```

## D.2. Bearing Data Reader (LabVIEW 2015)

*This script is to be used to read and organize the data file produced from bearing tests perfomed prior to 5/14/19.*

```
import numpy as np
import matplotlib.pyplot as plt



file_name = '2019_04_03_test1.dat'
data = np.loadtxt(file_name)

time = data[:, 0] #Run time (s)

speed = data[:, 28] #Processed tachometer speed (rpm)

torque = data[:, 25] #Processed torque meter torque (Nm)

plt.figure(2)
plt.clf()
plt.plot(time/(3600), speed, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Speed (rpm)')
plt.title("Rotational Shaft Speed")
plt.show()

plt.figure(3)
plt.clf()
plt.plot(time/(3600), torque, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Torque (Nm)')
plt.title('Motor Torque')
plt.show()
```

### D.3. Bearing Data Reader (3 Thermocouples)

*This script is to be used to read and organize the data file produced from bearing tests consisting of a three thermocouple per oil well set up (prior to 10/21/19).*

```
import numpy as np
import matplotlib.pyplot as plt



file_name = '2019_07_26_Shakedown.dat'
data = np.loadtxt(file_name)

time = data[:, 1] #Run time (s)
bins = [] #64 frequency bins
for i in range(64):
   bins.append(i*31.25)
#######################################################################################
#Temperatures

TC1 = data[:, 2] #Thermocouple temperature values (C)
TC2 = data[:, 5]
TC3 = data[:, 8]
TC4 = data[:, 11]
TC5 = data[:, 14]
TC6 = data[:, 17]

T_gov = np.mean(np.array([TC1, TC2, TC3]), axis = 0) #Governor side temperature average
T_motor = np.mean(np.array([TC4, TC5, TC6]), axis = 0) #Motor side temperature average

plt.figure(1)
plt.clf()
plt.plot(time/(3600), T_gov, 'r-', label = 'Governor Side')
plt.plot(time/(3600), T_motor, 'b-', label = 'Motor Side')
plt.xlabel('Time (hrs)')
plt.ylabel('Temperature (C)')
plt.title('Average Oil Temperature')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 2, frameon = False)
plt.show()
##############################################################################
#Speed and Torque

speed = data[:, 35] #Processed tachometer speed (rpm)

torque = data[:, 39] #Processed torque meter torque (Nm)

plt.figure(2)
plt.clf()
plt.plot(time/(3600), speed, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Speed (rpm)')
plt.title("Rotational Shaft Speed")
plt.show()

plt.figure(3)
plt.clf()
plt.plot(time/(3600), torque, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Torque (Nm)')
plt.title('Motor Torque')
plt.show()

##############################################################################
#Accelerometer 1

AC1 = data[:, 174:238] #Accelerometer 1 64 frequency bins (g)

#plt.figure(23)
```

```
#plt.clf()
#plt.plot(time/(3600), AC1[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC1[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC1[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC1[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC1[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC1[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC1[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(24)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC1[:, 8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC1[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC1[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC1[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC1[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC1[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(25)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC1[:, 15], 'r-', label = '468.75-500 HZ')
```

```
#plt.plot(time/(3600), AC1[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC1[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC1[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC1[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC1[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(26)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC1[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC1[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC1[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC1[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC1[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC1[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(27)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC1[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC1[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC1[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC1[:, 32], 'c-', label = '1000-1031.25 HZ')
```

```
#plt.plot(time/(3600), AC1[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC1[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(28)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC1[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC1[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC1[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC1[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC1[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC1[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(29)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC1[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC1[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC1[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC1[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC1[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC1[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
```

```
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(30)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC1[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC1[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC1[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC1[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC1[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC1[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(31)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC1[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC1[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC1[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC1[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC1[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC1[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC1[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1750-2000 Hz)')
```

```
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

plt.figure(32)
plt.clf()
AC1_m = []
for i in range(64):
    AC1_m.append(np.mean(AC1[:, i]))
plt.step(bins, AC1_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 1 Acceleration Averages')
plt.show()

for i in range(64):
    if (AC1_m[i] >= .05):
        print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC1_m[i], 4), 'g')
##########################################################################################
#Accelerometer 2

AC2 = data[:, 240:304] #Accelerometer 2 64 frequency bins (g)

#plt.figure(33)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC2[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC2[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC2[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC2[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC2[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC2[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
```

```
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(34)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC2[:, 8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC2[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC2[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC2[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC2[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC2[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(35)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC2[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC2[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC2[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC2[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC2[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC2[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
```

```
#plt.show()
#
#plt.figure(36)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC2[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC2[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC2[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC2[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC2[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC2[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(37)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC2[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC2[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC2[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC2[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC2[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC2[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(38)
```

```
#plt.clf()
#plt.plot(time/(3600), AC2[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC2[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC2[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC2[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC2[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC2[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC2[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(39)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC2[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC2[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC2[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC2[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC2[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC2[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(40)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC2[:, 50], 'r-', label = '1562.5-1593.75 HZ')
```

```
#plt.plot(time/(3600), AC2[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC2[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC2[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC2[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC2[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(41)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC2[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC2[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC2[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC2[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC2[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC2[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC2[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()


plt.figure(42)
plt.clf()
AC2_m = []
for i in range(64):
    AC2_m.append(np.mean(AC2[:, i]))
plt.step(bins, AC2_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 2 Acceleration Averages')
plt.show()

for i in range(64):
    if (AC2_m[i] >= .05):
        print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC2_m[i], 4), 'g')
##################################################################################
#Accelerometer 3

AC3 = data[:, 42:106] #Accelerometer 3 64 frequency bins (g)
#plt.figure(4)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC3[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC3[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC3[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC3[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC3[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC3[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(5)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC3[:, 8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC3[:, 9], 'b-', label = '281.25-312.5 HZ')
```

```
#plt.plot(time/(3600), AC3[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC3[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC3[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC3[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(6)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC3[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC3[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC3[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC3[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC3[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC3[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(7)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC3[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC3[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC3[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC3[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC3[:, 26], 'm-', label = '812.5-843.75 HZ')
```

```
#plt.plot(time/(3600), AC3[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(8)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC3[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC3[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC3[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC3[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC3[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC3[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(9)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC3[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC3[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC3[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC3[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC3[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC3[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
```

```
#plt.title('Accelerometer 3 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(10)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC3[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC3[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC3[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC3[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC3[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC3[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(11)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC3[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC3[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC3[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC3[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC3[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC3[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
```

```
#plt.figure(11)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC3[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC3[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC3[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC3[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC3[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC3[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC3[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

plt.figure(12)
plt.clf()
AC3_m = []
for i in range(64):
    AC3_m.append(np.mean(AC3[:, i]))
plt.step(bins, AC3_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 3 Acceleration Averages')
plt.show()

for i in range(64):
    if (AC3_m[i] >= .05):
        print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC3_m[i], 4), 'g')
```

```
################################################################################
#Accelerometer 4

AC4 = data[:, 108:172] #Accelerometer 4 64 frequency bins (g)

#plt.figure(13)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC4[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC4[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC4[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC4[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC4[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC4[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(14)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC4[: ,8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC4[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC4[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC4[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC4[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC4[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
```

```
#plt.show()

#plt.figure(15)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC4[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC4[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC4[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC4[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC4[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC4[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(16)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC4[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC4[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC4[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC4[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC4[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC4[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(17)
```

```
#plt.clf()
#plt.plot(time/(3600), AC4[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC4[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC4[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC4[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC4[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC4[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC4[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(18)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC4[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC4[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC4[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC4[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC4[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC4[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(19)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC4[:, 43], 'r-', label = '1343.75-1375 HZ')
```

```
#plt.plot(time/(3600), AC4[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC4[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC4[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC4[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC4[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(20)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC4[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC4[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC4[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC4[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC4[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC4[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(21)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC4[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC4[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC4[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC4[:, 60], 'c-', label = '1875-1906.25 HZ')
```

```
#plt.plot(time/(3600), AC4[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC4[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC4[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

plt.figure(22)
plt.clf()
AC4_m = []
for i in range(64):
    AC4_m.append(np.mean(AC4[:, i]))
plt.step(bins, AC4_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 4 Acceleration Averages')
plt.show()

for i in range(64):
    if (AC4_m[i] >= .05):
        print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC4_m[i], 4), 'g')

##############################################################################
#Torque vs Temperature

temp = np.mean(np.array([T_gov, T_motor]), axis = 0)
for i in (range(len(time/3600))):
    if time[i]/3600 >= .125:
        a = i
        break
```

```
for i in (range(len(time/3600))):
    if time[i]/3600 >= 2.5:
        b = i
        break


plt.figure(43)
plt.clf()
plt.plot(temp[a:b], torque[a:b], 'k-')
plt.xlabel('Temperature (C)')
plt.ylabel('Torque (Nm)')
plt.title('Torque vs Temperature')
plt.show()
```

### D.4. Bearing Data Reader (4 Thermocouples, Misconfiguration)

*This script is to be used to read and organize the data file produced from bearing tests consisting of a four thermocouple per oil well set up, conducted between 5/14/19 and 12/09/19. This script is needed in order to mitigate an instrumentation setup error. This error consisted of all four accelerometers having been connected to the wrong channels; this issue was discovered and rectified on 12/09/19.*

```
import numpy as np

import matplotlib.pyplot as plt



file_name = '2019_11_08_test1000.dat'

data = np.loadtxt(file_name)


time = data[:, 1] #Run time (s)

bins = [] #64 frequency bins

for i in range(64):

    bins.append(i*31.25)

################################################################################

#Temperatures


TC1 = data[:, 2] #Thermocouple temperature values (C)

TC2 = data[:, 5]

TC3 = data[:, 8]

TC4 = data[:, 11]

TC5 = data[:, 14]

TC6 = data[:, 17]

TC7 = data[:, 20]

TC8 = data[:, 23]


T_gov = np.mean(np.array([TC1, TC2, TC3, TC4]), axis = 0) #Governor side temperature average

T_motor = np.mean(np.array([TC5, TC6, TC7, TC8]), axis = 0) #Motor side temperature average
```

```
plt.figure(1)
plt.clf()
plt.plot(time/(3600), T_gov, 'r-', label = 'Governor Side')
plt.plot(time/(3600), T_motor, 'b-', label = 'Motor Side')
plt.xlabel('Time (hrs)')
plt.ylabel('Temperature (C)')
plt.title('Average Oil Temperature')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 2, frameon = False)
plt.show()
##############################################################################
#Speed and Torque

speed = data[:, 41] #Processed tachometer speed (rpm)

torque = data[:, 45] #Processed torque meter torque (Nm)

plt.figure(2)
plt.clf()
plt.plot(time/(3600), speed, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Speed (rpm)')
plt.title("Rotational Shaft Speed")
plt.show()

plt.figure(3)
plt.clf()
plt.plot(time/(3600), torque, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Torque (Nm)')
plt.title('Motor Torque')
plt.show()
##############################################################################
```

```
#Accelerometer 1

AC1 = data[:, 174:238] #Accelerometer 1 64 frequency bins (g)

#plt.figure(23)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC1[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC1[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC1[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC1[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC1[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC1[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(24)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC1[:, 8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC1[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC1[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC1[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC1[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC1[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
```

```
#
#plt.figure(25)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC1[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC1[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC1[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC1[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC1[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC1[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(26)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC1[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC1[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC1[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC1[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC1[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC1[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(27)
#plt.clf()
```

```
#plt.plot(time/(3600), AC1[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC1[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC1[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC1[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC1[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC1[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC1[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(28)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC1[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC1[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC1[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC1[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC1[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC1[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(29)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC1[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC1[:, 44], 'b-', label = '1375-1406.25 HZ')
```

```
#plt.plot(time/(3600), AC1[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC1[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC1[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC1[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(30)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC1[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC1[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC1[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC1[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC1[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC1[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(31)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC1[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC1[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC1[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC1[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC1[:, 61], 'm-', label = '1906.25-1937.5 HZ')
```

```
#plt.plot(time/(3600), AC1[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC1[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

plt.figure(32)
plt.clf()
AC1_m = []
for i in range(64):
   AC1_m.append(np.mean(AC1[:, i]))
plt.step(bins, AC1_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 1 Acceleration Averages')
plt.show()

for i in range(64):
   if (AC1_m[i] >= .05):
      print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC1_m[i], 4), 'g')
################################################################################
#Accelerometer 2

AC2 = data[:, 240:304] #Accelerometer 2 64 frequency bins (g)

#plt.figure(33)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC2[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC2[:, 2], 'b-', label = '62.5-93.75 HZ')
```

```
#plt.plot(time/(3600), AC2[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC2[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC2[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC2[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(34)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC2[:, 8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC2[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC2[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC2[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC2[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC2[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(35)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC2[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC2[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC2[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC2[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC2[:, 19], 'm-', label = '593.75-625 HZ')
```

```
#plt.plot(time/(3600), AC2[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(36)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC2[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC2[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC2[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC2[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC2[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC2[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(37)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC2[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC2[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC2[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC2[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC2[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC2[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
```

```
#plt.title('Accelerometer 2 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(38)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC2[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC2[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC2[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC2[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC2[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC2[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(39)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC2[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC2[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC2[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC2[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC2[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC2[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
```

```
#
#plt.figure(40)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC2[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC2[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC2[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC2[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC2[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC2[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(41)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC2[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC2[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC2[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC2[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC2[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC2[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC2[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()


plt.figure(42)
plt.clf()
AC2_m = []
for i in range(64):
    AC2_m.append(np.mean(AC2[:, i]))
plt.step(bins, AC2_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 2 Acceleration Averages')
plt.show()

for i in range(64):
    if (AC2_m[i] >= .05):
        print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC2_m[i], 4), 'g')
################################################################################
#Accelerometer 3

AC3 = data[:, 42:106] #Accelerometer 3 64 frequency bins (g)
#plt.figure(4)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC3[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC3[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC3[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC3[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC3[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC3[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
```

```
#plt.figure(5)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC3[:, 8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC3[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC3[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC3[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC3[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC3[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(6)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC3[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC3[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC3[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC3[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC3[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC3[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(7)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 21], 'k-', label = '656.25-687.5 HZ')
```

```
#plt.plot(time/(3600), AC3[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC3[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC3[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC3[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC3[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC3[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(8)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC3[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC3[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC3[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC3[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC3[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC3[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(9)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC3[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC3[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC3[:, 38], 'g-', label = '1187.5-1218.75 HZ')
```

```
#plt.plot(time/(3600), AC3[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC3[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC3[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(10)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC3[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC3[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC3[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC3[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC3[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC3[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(11)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC3[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC3[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC3[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC3[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC3[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC3[:, 55], 'y-', label = '1718.75-1750 HZ')
```

```
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(11)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC3[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC3[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC3[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC3[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC3[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC3[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC3[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

plt.figure(12)
plt.clf()
AC3_m = []
for i in range(64):
    AC3_m.append(np.mean(AC3[:, i]))
plt.step(bins, AC3_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 3 Acceleration Averages')
plt.show()
```

```
for i in range(64):
    if (AC3_m[i] >= .05):
        print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC3_m[i], 4), 'g')

######################################################################################
#Accelerometer 4

AC4 = data[:, 108:172] #Accelerometer 4 64 frequency bins (g)

#plt.figure(13)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC4[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC4[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC4[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC4[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC4[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC4[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(14)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC4[: ,8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC4[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC4[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC4[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC4[:, 12], 'm-', label = '375-406.25 HZ')
```

```
#plt.plot(time/(3600), AC4[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(15)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC4[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC4[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC4[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC4[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC4[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC4[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(16)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC4[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC4[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC4[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC4[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC4[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC4[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
```

```
#plt.title('Accelerometer 4 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(17)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC4[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC4[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC4[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC4[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC4[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC4[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(18)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC4[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC4[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC4[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC4[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC4[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC4[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
```

```
#plt.figure(19)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC4[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC4[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC4[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC4[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC4[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC4[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(20)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC4[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC4[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC4[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC4[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC4[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC4[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(21)
#plt.clf()
```

```
#plt.plot(time/(3600), AC4[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC4[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC4[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC4[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC4[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC4[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC4[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC4[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

plt.figure(22)
plt.clf()
AC4_m = []
for i in range(64):
    AC4_m.append(np.mean(AC4[:, i]))
plt.step(bins, AC4_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 4 Acceleration Averages')
plt.show()

for i in range(64):
    if (AC4_m[i] >= .05):
        print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC4_m[i], 4), 'g')

################################################################################
#Torque vs Temperature
```

```
temp = np.mean(np.array([T_gov, T_motor]), axis = 0)
for i in (range(len(time/3600))):
    if time[i]/3600 >= .125:
        a = i
        break
for i in (range(len(time/3600))):
    if time[i]/3600 >= 2.5:
        b = i
        break

plt.figure(43)
plt.clf()
plt.plot(temp[a:b], torque[a:b], 'k-')
plt.xlabel('Temperature (C)')
plt.ylabel('Torque (Nm)')
plt.title('Torque vs Temperature')
plt.show()
```

### D.5. Bearing Data Reader (4 Thermocouples)

*This script is to be used to read and organize the data file produced from bearing tests consisting of a four thermocouple per oil well set up.*

```
import numpy as np
import matplotlib.pyplot as plt



file_name = '2019_10_24_test1.dat'
data = np.loadtxt(file_name)

time = data[:, 1] #Run time (s)
bins = [] #64 frequency bins
for i in range(64):
    bins.append(i*31.25)
##############################################################################
#Temperatures

TC1 = data[:, 2] #Thermocouple temperature values (C)
TC2 = data[:, 5]
TC3 = data[:, 8]
TC4 = data[:, 11]
TC5 = data[:, 14]
TC6 = data[:, 17]
TC7 = data[:, 20]
TC8 = data[:, 23]

T_gov = np.mean(np.array([TC1, TC2, TC3, TC4]), axis = 0) #Governor side temperature average
T_motor = np.mean(np.array([TC5, TC6, TC7, TC8]), axis = 0) #Motor side temperature average

plt.figure(1)
plt.clf()
plt.plot(time/(3600), T_gov, 'r-', label = 'Governor Side')
plt.plot(time/(3600), T_motor, 'b-', label = 'Motor Side')
plt.xlabel('Time (hrs)')
plt.ylabel('Temperature (C)')
plt.title('Average Oil Temperature')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 2, frameon = False)
plt.show()
######################################################################################
#Speed and Torque

speed = data[:, 41] #Processed tachometer speed (rpm)

torque = data[:, 45] #Processed torque meter torque (Nm)

plt.figure(2)
plt.clf()
plt.plot(time/(3600), speed, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Speed (rpm)')
plt.title("Rotational Shaft Speed")
plt.show()

plt.figure(3)
plt.clf()
plt.plot(time/(3600), torque, 'k-')
plt.xlabel('Time (hrs)')
plt.ylabel('Torque (Nm)')
plt.title('Motor Torque')
plt.show()
######################################################################################
#Accelerometer 1

AC1 = data[:, 48:112] #Accelerometer 1 64 frequency bins (g)
```

```
#plt.figure(4)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC1[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC1[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC1[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC1[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC1[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC1[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(5)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC1[:, 8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC1[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC1[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC1[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC1[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC1[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(6)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 14], 'k-', label = '437.5-468.75 HZ')
```

```
#plt.plot(time/(3600), AC1[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC1[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC1[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC1[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC1[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC1[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(7)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC1[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC1[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC1[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC1[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC1[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC1[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(8)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC1[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC1[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC1[:, 31], 'g-', label = '968.75-1000 HZ')
```

```
#plt.plot(time/(3600), AC1[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC1[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC1[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(9)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC1[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC1[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC1[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC1[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC1[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC1[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(10)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC1[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC1[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC1[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC1[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC1[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC1[:, 48], 'y-', label = '1500-1531.25 HZ')
```

```
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(11)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC1[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC1[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC1[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC1[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC1[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC1[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 1 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(11)
#plt.clf()
#plt.plot(time/(3600), AC1[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC1[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC1[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC1[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC1[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC1[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC1[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC1[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
```

```
#plt.title('Accelerometer 1 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

plt.figure(12)
plt.clf()
AC1_m = []
for i in range(64):
    AC1_m.append(np.mean(AC1[:, i]))
plt.step(bins, AC1_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 1 Acceleration Averages')
plt.show()

for i in range(64):
    if (AC1_m[i] >= .05):
        print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC1_m[i], 4), 'g')
#####################################################################################
#Accelerometer 2

AC2 = data[:, 114:178] #Accelerometer 2 64 frequency bins (g)

#plt.figure(13)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC2[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC2[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC2[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC2[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC2[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC2[:, 6], 'y-', label = '187.5-218.75 HZ')
```

```
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(14)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC2[: ,8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC2[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC2[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC2[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC2[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC2[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(15)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC2[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC2[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC2[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC2[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC2[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC2[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(437.5-656.25 Hz)')
```

```
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(16)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC2[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC2[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC2[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC2[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC2[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC2[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(17)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC2[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC2[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC2[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC2[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC2[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC2[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
```

```
#plt.figure(18)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC2[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC2[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC2[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC2[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC2[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC2[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(19)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC2[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC2[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC2[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC2[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC2[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC2[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(20)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 49], 'k-', label = '1531.25-1562.5 HZ')
```

```
#plt.plot(time/(3600), AC2[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC2[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC2[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC2[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC2[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC2[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

#plt.figure(21)
#plt.clf()
#plt.plot(time/(3600), AC2[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC2[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC2[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC2[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC2[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC2[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC2[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC2[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 2 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

plt.figure(22)
plt.clf()
AC2_m = []
for i in range(64):
    AC2_m.append(np.mean(AC2[:, i]))
```

```
plt.step(bins, AC2_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 2 Acceleration Averages')
plt.show()

for i in range(64):
    if (AC2_m[i] >= .05):
        print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC2_m[i], 4), 'g')
######################################################################################
#Accelerometer 3

AC3 = data[:, 180:244] #Accelerometer 3 64 frequency bins (g)

#plt.figure(23)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC3[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC3[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC3[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC3[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC3[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC3[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(24)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 7], 'k-', label = '218.75-250 HZ')
```

```
#plt.plot(time/(3600), AC3[:, 8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC3[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC3[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC3[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC3[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC3[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(218.75-437.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(25)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC3[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC3[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC3[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC3[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC3[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC3[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(26)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC3[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC3[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC3[:, 24], 'g-', label = '750-781.25 HZ')
```

```
#plt.plot(time/(3600), AC3[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC3[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC3[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(27)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC3[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC3[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC3[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC3[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC3[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC3[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(28)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC3[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC3[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC3[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC3[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC3[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC3[:, 41], 'y-', label = '1281.25-1312.5 HZ')
```

```
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(29)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 42], 'k-', label = '1312.5-1343.75 HZ')
#plt.plot(time/(3600), AC3[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC3[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC3[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC3[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC3[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC3[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(30)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC3[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC3[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC3[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC3[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC3[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC3[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1531.25 1750 Hz)')
```

```
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(31)
#plt.clf()
#plt.plot(time/(3600), AC3[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC3[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC3[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC3[:, 59], 'g-', label = '1843.75-1875 HZ')
#plt.plot(time/(3600), AC3[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC3[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC3[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC3[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 3 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

plt.figure(32)
plt.clf()
AC3_m = []
for i in range(64):
   AC3_m.append(np.mean(AC3[:, i]))
plt.step(bins, AC3_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 3 Acceleration Averages')
plt.show()

for i in range(64):
   if (AC3_m[i] >= .05):
    print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC3_m[i], 4), 'g')
#############################################################################
#Accelerometer 4

AC4 = data[:, 246:310] #Accelerometer 4 64 frequency bins (g)

#plt.figure(33)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 0], 'k-', label = '0-31.25 HZ')
#plt.plot(time/(3600), AC4[:, 1], 'r-', label = '31.25-62.5 HZ')
#plt.plot(time/(3600), AC4[:, 2], 'b-', label = '62.5-93.75 HZ')
#plt.plot(time/(3600), AC4[:, 3], 'g-', label = '93.75-125 HZ')
#plt.plot(time/(3600), AC4[:, 4], 'c-', label = '125-156.25 HZ')
#plt.plot(time/(3600), AC4[:, 5], 'm-', label = '156.25-187.5 HZ')
#plt.plot(time/(3600), AC4[:, 6], 'y-', label = '187.5-218.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(0-218.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(34)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 7], 'k-', label = '218.75-250 HZ')
#plt.plot(time/(3600), AC4[:, 8], 'r-', label = '250-281.25 HZ')
#plt.plot(time/(3600), AC4[:, 9], 'b-', label = '281.25-312.5 HZ')
#plt.plot(time/(3600), AC4[:, 10], 'g-', label = '312.5-343.75 HZ')
#plt.plot(time/(3600), AC4[:, 11], 'c-', label = '343.75-375 HZ')
#plt.plot(time/(3600), AC4[:, 12], 'm-', label = '375-406.25 HZ')
#plt.plot(time/(3600), AC4[:, 13], 'y-', label = '406.25-437.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(218.75-437.5 Hz)')
```

```
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(35)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 14], 'k-', label = '437.5-468.75 HZ')
#plt.plot(time/(3600), AC4[:, 15], 'r-', label = '468.75-500 HZ')
#plt.plot(time/(3600), AC4[:, 16], 'b-', label = '500-531.25 HZ')
#plt.plot(time/(3600), AC4[:, 17], 'g-', label = '531.25-562.5 HZ')
#plt.plot(time/(3600), AC4[:, 18], 'c-', label = '562.5-593.75 HZ')
#plt.plot(time/(3600), AC4[:, 19], 'm-', label = '593.75-625 HZ')
#plt.plot(time/(3600), AC4[:, 20], 'y-', label = '625-656.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(437.5-656.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(36)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 21], 'k-', label = '656.25-687.5 HZ')
#plt.plot(time/(3600), AC4[:, 22], 'r-', label = '687.5-718.75 HZ')
#plt.plot(time/(3600), AC4[:, 23], 'b-', label = '718.75-750 HZ')
#plt.plot(time/(3600), AC4[:, 24], 'g-', label = '750-781.25 HZ')
#plt.plot(time/(3600), AC4[:, 25], 'c-', label = '781.25-812.5 HZ')
#plt.plot(time/(3600), AC4[:, 26], 'm-', label = '812.5-843.75 HZ')
#plt.plot(time/(3600), AC4[:, 27], 'y-', label = '843.75-875 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(656.25-875 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
```

```
#plt.figure(37)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 28], 'k-', label = '875-906.25 HZ')
#plt.plot(time/(3600), AC4[:, 29], 'r-', label = '906.25-937.5 HZ')
#plt.plot(time/(3600), AC4[:, 30], 'b-', label = '937.5-968.75 HZ')
#plt.plot(time/(3600), AC4[:, 31], 'g-', label = '968.75-1000 HZ')
#plt.plot(time/(3600), AC4[:, 32], 'c-', label = '1000-1031.25 HZ')
#plt.plot(time/(3600), AC4[:, 33], 'm-', label = '1031.25-1062.5 HZ')
#plt.plot(time/(3600), AC4[:, 34], 'y-', label = '1062.5-1093.75 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(875-1093.75 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(38)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 35], 'k-', label = '1093.75-1125 HZ')
#plt.plot(time/(3600), AC4[:, 36], 'r-', label = '1125-1156.25 HZ')
#plt.plot(time/(3600), AC4[:, 37], 'b-', label = '1156.25-1187.5 HZ')
#plt.plot(time/(3600), AC4[:, 38], 'g-', label = '1187.5-1218.75 HZ')
#plt.plot(time/(3600), AC4[:, 39], 'c-', label = '1218.75-1250 HZ')
#plt.plot(time/(3600), AC4[:, 40], 'm-', label = '1250-1281.25 HZ')
#plt.plot(time/(3600), AC4[:, 41], 'y-', label = '1281.25-1312.5 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1093.75-1312.5 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(39)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 42], 'k-', label = '1312.5-1343.75 HZ')
```

```
#plt.plot(time/(3600), AC4[:, 43], 'r-', label = '1343.75-1375 HZ')
#plt.plot(time/(3600), AC4[:, 44], 'b-', label = '1375-1406.25 HZ')
#plt.plot(time/(3600), AC4[:, 45], 'g-', label = '1406.25-1437.5 HZ')
#plt.plot(time/(3600), AC4[:, 46], 'c-', label = '1437.5-1468.75 HZ')
#plt.plot(time/(3600), AC4[:, 47], 'm-', label = '1468.75-1500 HZ')
#plt.plot(time/(3600), AC4[:, 48], 'y-', label = '1500-1531.25 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1312.5-1531.25 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(40)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 49], 'k-', label = '1531.25-1562.5 HZ')
#plt.plot(time/(3600), AC4[:, 50], 'r-', label = '1562.5-1593.75 HZ')
#plt.plot(time/(3600), AC4[:, 51], 'b-', label = '1593.75-1625 HZ')
#plt.plot(time/(3600), AC4[:, 52], 'g-', label = '1625-1656.25 HZ')
#plt.plot(time/(3600), AC4[:, 53], 'c-', label = '1656.25-1687.5 HZ')
#plt.plot(time/(3600), AC4[:, 54], 'm-', label = '1687.5-1718.75 HZ')
#plt.plot(time/(3600), AC4[:, 55], 'y-', label = '1718.75-1750 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1531.25 1750 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()
#
#plt.figure(41)
#plt.clf()
#plt.plot(time/(3600), AC4[:, 56], 'k-', label = '1750-1781.25 HZ')
#plt.plot(time/(3600), AC4[:, 57], 'r-', label = '1781.25-1812.5 HZ')
#plt.plot(time/(3600), AC4[:, 58], 'b-', label = '1812.5-1843.75 HZ')
#plt.plot(time/(3600), AC4[:, 59], 'g-', label = '1843.75-1875 HZ')
```

```
#plt.plot(time/(3600), AC4[:, 60], 'c-', label = '1875-1906.25 HZ')
#plt.plot(time/(3600), AC4[:, 61], 'm-', label = '1906.25-1937.5 HZ')
#plt.plot(time/(3600), AC4[:, 62], 'y-', label = '1937.5-1968.75 HZ')
#plt.plot(time/(3600), AC4[:, 63], 'k-', label = '1968.75-2000 HZ')
#plt.xlabel('Time (hrs)')
#plt.ylabel('Acceleration (g)')
#plt.title('Accelerometer 4 Frequency Breakdown\n(1750-2000 Hz)')
#plt.legend(loc = 'upper center', bbox_to_anchor = (1.2, 1), frameon = False)
#plt.show()

plt.figure(42)
plt.clf()
AC4_m = []
for i in range(64):
    AC4_m.append(np.mean(AC4[:, i]))
plt.step(bins, AC4_m, 'k-', where = 'post')
plt.plot(0, .3)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Acceleration (g)')
plt.title('Accelerometer 4 Acceleration Averages')
plt.show()

for i in range(64):
    if (AC4_m[i] >= .05):
        print('Frequency Bin:', bins[i], '-', bins[i+1], 'Hz  ', '\t', 'Acceleration:', round(AC4_m[i], 4), 'g')
##################################################################################
#Torque vs Temperature

temp = np.mean(np.array([T_gov, T_motor]), axis = 0)
for i in (range(len(time/3600))):
    if time[i]/3600 >= .125:
        a = i
        break
```

```
for i in (range(len(time/3600))):
    if time[i]/3600 >= 2.5:
        b = i
        break

plt.figure(43)
plt.clf()
plt.plot(temp[a:b], torque[a:b], 'k-')
plt.xlabel('Temperature (C)')
plt.ylabel('Torque (Nm)')
plt.title('Torque vs Temperature')
plt.show()
```

### D.6. Torque vs Speed Regression Generator (LabVIEW 2015)

*This script is to be used to read and organize the data files produced from steady state torque tests perfomed prior to 5/14/19.*

```
import numpy as np

import matplotlib.pyplot as plt



file_name = '2019_02_22_test1_1000F.dat'

data_1000 = np.loadtxt(file_name)


file_name = '2019_02_22_test2_1500F.dat'

data_1500 = np.loadtxt(file_name)


file_name = '2019_02_22_test3_2000F.dat'

data_2000 = np.loadtxt(file_name)


file_name = '2019_02_22_test4_2500F.dat'

data_2500 = np.loadtxt(file_name)


file_name = '2019_02_22_test5_3000F.dat'

data_3000 = np.loadtxt(file_name)


file_name = '2019_02_22_test10_3490F.dat'

data_3490 = np.loadtxt(file_name)


file_name = '2019_02_22_test6_1000R.dat'

data_neg_1000 = np.loadtxt(file_name)


file_name = '2019_02_22_test7_1500R.dat'

data_neg_1500 = np.loadtxt(file_name)
```

```
file_name = '2019_02_22_test8_2000R.dat'
data_neg_2000 = np.loadtxt(file_name)

file_name = '2019_02_22_test9_2250R.dat'
data_neg_2250 = np.loadtxt(file_name)


################################################################################
#Data Formatting

speed_1000 = 1000
torque_1000 = data_1000[:, 25]
torque_1000 = np.mean(torque_1000[200:300])

speed_1500 = 1500
torque_1500 = data_1500[:, 25]
torque_1500 = np.mean(torque_1500[200:300])

speed_2000 = 2000
torque_2000 = data_2000[:, 25]
torque_2000 = np.mean(torque_2000[200:300])

speed_2500 = 2500
torque_2500 = data_2500[:, 25]
torque_2500 = np.mean(torque_2500[200:300])

speed_3000 = 3000
torque_3000 = data_3000[:, 25]
torque_3000 = np.mean(torque_3000[200:300])

speed_3490 = 3490
torque_3490 = data_3490[:, 25]
torque_3490 = np.mean(torque_3490[200:300])
```

```
speed_neg_1000 = -1000

torque_neg_1000 = data_neg_1000[:, 25]

torque_neg_1000 = np.mean(-torque_neg_1000[200:300])


speed_neg_1500 = -1500

torque_neg_1500 = data_neg_1500[:, 25]

torque_neg_1500 = np.mean(-torque_neg_1500[200:300])


speed_neg_2000 = -2000

torque_neg_2000 = data_neg_2000[:, 25]

torque_neg_2000 = np.mean(-torque_neg_2000[200:300])


speed_neg_2250 = -2250

torque_neg_2250 = data_neg_2250[:, 25]

torque_neg_2250 = np.mean(-torque_neg_2250[200:300])


speed = [speed_1000, speed_1500, speed_2000, speed_2500, speed_3000, speed_3490, speed_neg_1000, speed_neg_1500, speed_neg_2000, speed_neg_2250]

torque = [torque_1000, torque_1500, torque_2000, torque_2500, torque_3000, torque_3490, torque_neg_1000, torque_neg_1500, torque_neg_2000, torque_neg_2250]

##############################################################################

#Regression


F = np.linspace(1000, 3490, 500)

R = np.linspace(-2250, -1000, 500)

x = np.linspace(-2250, 3490, 1000)


reg_F = np.polyfit(speed[0:6], torque[0:6], 2)

f_F = np.poly1d(reg_F)

reg_R = np.polyfit(speed[6::], torque[6::], 2)

f_R = np.poly1d(reg_R)

reg = np.polyfit(speed, torque, 3)

f = np.poly1d(reg)
```

```
mean = np.mean(torque[0:6])
SS_reg = np.sum((f_F(speed[0:6]) - mean)**2)
SS_tot = np.sum((torque[0:6] - mean)**2)
R2_F = SS_reg/SS_tot

mean = np.mean(torque[6::])
SS_reg = np.sum((f_R(speed[6::]) - mean)**2)
SS_tot = np.sum((torque[6::] - mean)**2)
R2_R = SS_reg/SS_tot

mean = np.mean(torque)
SS_reg = np.sum((f(speed) - mean)**2)
SS_tot = np.sum((torque - mean)**2)
R2 = SS_reg/SS_tot
#####################################################################################
#Plot

plt.figure(1)
plt.clf()
plt.plot(speed[0:6], torque[0:6], 'ro')
plt.plot(F, f_F(F), 'k--')
plt.xlabel('Shaft Speed (rpm)')
plt.ylabel('Torque (Nm)')
plt.title('Forward Spin')
plt.show()
print(f_F, '\nR-squared =', R2_F)

plt.figure(2)
plt.clf()
plt.plot(speed[6::], torque[6::], 'ro')
plt.plot(R, f_R(R), 'k--')
plt.xlabel('Shaft Speed (rpm)')
plt.ylabel('Torque (Nm)')
plt.title('Reverse Spin')
plt.show()
print(f_R, '\nR-squared =', R2_R)

plt.figure(3)
plt.clf()
plt.plot(speed, torque, 'ro')
plt.plot(x, f(x), 'k--')
plt.xlabel('Shaft Speed (rpm)')
plt.ylabel('Torque (Nm)')
plt.title('Torque vs Speed')
plt.show()
print(f, '\nR-squared =', R2)

plt.figure(4)
plt.clf()
plt.plot(x, f(x), 'k--', label = 'Combined')
plt.plot(F, f_F(F), 'r--', label = 'Forward')
plt.plot(R, f_R(R), 'b--', label = 'Reverse')
plt.xlabel('Shaft Speed (rpm)')
plt.ylabel('Torque (Nm)')
plt.title('Regression Comparison')
plt.legend(loc = 'upper center', bbox_to_anchor = (.5, -.15), ncol = 3, frameon = False)
plt.show()
```

### D.7. Torque vs Speed Regression Generator

*This script is to be used to read and organize the data files produced from steady state torque tests.*

```
import numpy as np
import matplotlib.pyplot as plt



file_name = '2019_10_29_500.dat'
data_500 = np.loadtxt(file_name)

file_name = '2019_10_29_1000.dat'
data_1000 = np.loadtxt(file_name)

file_name = '2019_10_29_1500.dat'
data_1500 = np.loadtxt(file_name)

file_name = '2019_10_29_2000.dat'
data_2000 = np.loadtxt(file_name)

file_name = '2019_10_29_2500.dat'
data_2500 = np.loadtxt(file_name)

file_name = '2019_10_29_3000.dat'
data_3000 = np.loadtxt(file_name)
```

```
file_name = '2019_10_29_3600.dat'

data_3600 = np.loadtxt(file_name)


file_name = '2019_10_29_-500.dat'

data_neg_500 = np.loadtxt(file_name)


file_name = '2019_10_29_-1000.dat'

data_neg_1000 = np.loadtxt(file_name)


file_name = '2019_10_29_-1500.dat'

data_neg_1500 = np.loadtxt(file_name)


file_name = '2019_10_29_-2000.dat'

data_neg_2000 = np.loadtxt(file_name)


file_name = '2019_10_29_-2250.dat'

data_neg_2250 = np.loadtxt(file_name)


file_name = '2019_10_29_-2500.dat'

data_neg_2500 = np.loadtxt(file_name)

#####################################################################################

#Data Formatting


speed_neg_500 = data_neg_500[:, 41]

speed_neg_500 = np.mean(-speed_neg_500)

torque_neg_500 = data_neg_500[:, 45]
```

```
torque_neg_500 = np.mean(-torque_neg_500)

speed_neg_1000 = data_neg_1000[:, 41]
speed_neg_1000 = np.mean(-speed_neg_1000)
torque_neg_1000 = data_neg_1000[:, 45]
torque_neg_1000 = np.mean(-torque_neg_1000)

speed_neg_1500 = data_neg_1500[:, 41]
speed_neg_1500 = np.mean(-speed_neg_1500)
torque_neg_1500 = data_neg_1500[:, 45]
torque_neg_1500 = np.mean(-torque_neg_1500)

speed_neg_2000 = data_neg_2000[:, 41]
speed_neg_2000 = np.mean(-speed_neg_2000)
torque_neg_2000 = data_neg_2000[:, 45]
torque_neg_2000 = np.mean(-torque_neg_2000)

speed_neg_2250 = data_neg_2250[:, 41]
speed_neg_2250 = np.mean(-speed_neg_2250)
torque_neg_2250 = data_neg_2250[:, 45]
torque_neg_2250 = np.mean(-torque_neg_2250)

speed_neg_2500 = data_neg_2500[:, 41]
speed_neg_2500 = np.mean(-speed_neg_2500)
torque_neg_2500 = data_neg_2500[:, 45]
torque_neg_2500 = np.mean(-torque_neg_2500)
```

```
speed_500 = data_500[:, 41]

speed_500 = np.mean(speed_500)

torque_500 = data_500[:, 45]

torque_500 = np.mean(torque_500)


speed_1000 = data_1000[:, 41]

speed_1000 = np.mean(speed_1000)

torque_1000 = data_1000[:, 45]

torque_1000 = np.mean(torque_1000)


speed_1500 = data_1500[:, 41]

speed_1500 = np.mean(speed_1500)

torque_1500 = data_1500[:, 45]

torque_1500 = np.mean(torque_1500)


speed_2000 = data_2000[:, 41]

speed_2000 = np.mean(speed_2000)

torque_2000 = data_2000[:, 45]

torque_2000 = np.mean(torque_2000)


speed_2500 = data_2500[:, 41]

speed_2500 = np.mean(speed_2500)

torque_2500 = data_2500[:, 45]

torque_2500 = np.mean(torque_2500)
```

```
speed_3000 = data_3000[:, 41]
speed_3000 = np.mean(speed_3000)
torque_3000 = data_3000[:, 45]
torque_3000 = np.mean(torque_3000)

speed_3600 = data_3600[:, 41]
speed_3600 = np.mean(speed_3600)
torque_3600 = data_3600[:, 45]
torque_3600 = np.mean(torque_3600)

speed = [speed_neg_500, speed_neg_1000, speed_neg_1500, speed_neg_2000, speed_neg_2250, speed_neg_2500, speed_500, speed_1000, speed_1500, speed_2000, speed_2500, speed_3000, speed_3600]
torque = [torque_neg_500, torque_neg_1000, torque_neg_1500, torque_neg_2000, torque_neg_2250, torque_neg_2500, torque_500, torque_1000, torque_1500, torque_2000, torque_2500, torque_3000, torque_3600]
#####################################################################################
#Regression

F = np.linspace(500, 3600, 500)
R = np.linspace(-2500, -500, 500)
x = np.linspace(-2500, 3600, 1000)

reg_F = np.polyfit(speed[6::], torque[6::], 2)
f_F = np.poly1d(reg_F)
reg_R = np.polyfit(speed[0:6], torque[0:6], 2)
f_R = np.poly1d(reg_R)
```

```
reg = np.polyfit(speed, torque, 3)

f = np.poly1d(reg)



mean = np.mean(torque[6::])

SS_reg = np.sum((f_F(speed[6::]) - mean)**2)

SS_tot = np.sum((torque[6::] - mean)**2)

R2_F = SS_reg/SS_tot



mean = np.mean(torque[0:6])

SS_reg = np.sum((f_R(speed[0:6]) - mean)**2)

SS_tot = np.sum((torque[0:6] - mean)**2)

R2_R = SS_reg/SS_tot



mean = np.mean(torque)

SS_reg = np.sum((f(speed) - mean)**2)

SS_tot = np.sum((torque - mean)**2)

R2 = SS_reg/SS_tot

#####################################################################################

#Plot



plt.figure(1)

plt.clf()

plt.plot(speed[6::], torque[6::], 'ro')

plt.plot(F, f_F(F), 'k--')

plt.xlabel('Shaft Speed (rpm)')

plt.ylabel('Torque (Nm)')
plt.title('Forward Spin')
plt.show()
print(f_F, '\nR-squared =', R2_F)

plt.figure(2)
plt.clf()
plt.plot(speed[0:6], torque[0:6], 'ro')
plt.plot(R, f_R(R), 'k--')
plt.xlabel('Shaft Speed (rpm)')
plt.ylabel('Torque (Nm)')
plt.title('Reverse Spin')
plt.show()
print(f_R, '\nR-squared =', R2_R)

plt.figure(3)
plt.clf()
plt.plot(speed, torque, 'ro')
plt.plot(x, f(x), 'k--')
plt.xlabel('Shaft Speed (rpm)')
plt.ylabel('Torque (Nm)')
plt.title('Torque vs Speed')
plt.show()
print(f, '\nR-squared =', R2)
```