

11-1-2007

Statistical Analysis of Fastener Vibration Life Tests

Christopher Cheatham
University of South Florida

Follow this and additional works at: <https://scholarcommons.usf.edu/etd>

 Part of the [American Studies Commons](#)

Scholar Commons Citation

Cheatham, Christopher, "Statistical Analysis of Fastener Vibration Life Tests" (2007). *Graduate Theses and Dissertations*.
<https://scholarcommons.usf.edu/etd/666>

This Thesis is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.

Statistical Analysis of Fastener Vibration Life Tests

by

Christopher Cheatham

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Mechanical Engineering
Department of Mechanical Engineering
College of Engineering
University of South Florida

Major Professor: Daniel P. Hess, Ph.D.
Craig Lusk, Ph.D.
Nathan Crane, Ph.D.

Date of Approval:
November 1, 2007

Keywords: preload loss, loosening, population predictions, threaded insert, bolt

© Copyright 2007, Christopher Cheatham

Dedication

I would like to dedicate this work to my late grandfather, Russ Woods. He is an immense and inspiring influence in my life, having always dedicated his time and effort for others, demonstrating his unselfishness and endless generosity. I believe that he lived by the motto “you do what you have to”, and in that spirit he helped and supported his family, friends, and even strangers.

Like all great teachers, my grandfather’s greatest lessons were imparted to me through his actions. Regardless of what I was involved in, he was always there to cheer me on and offer his support. Through his work ethic he taught me the value of hard work, efficiency, and thoroughness. He taught me to never quit until the job is done, and done right. In life he taught me how to be generous, helpful, caring, punctual, brave, and obedient. Most importantly, he has taught me to enjoy life no matter how long or short it may be.

Thank you Old Man, this one is for you.

Acknowledgment

I would like to express my appreciation for Dr. Daniel Hess for all of the guidance, time, and support he has given me throughout the completion of this work. I would like to thank Dr. Craig Lusk and Dr. Nathan Crane for giving their time and energy to be on my committee.

I am grateful to my mother Shelley Dougherty who has gone above and beyond of what a mother needs to do. She is one of the main reasons I am where I am today. I am equally as grateful to my father Jeff Cheatham, who has supported me through my life's journey. He has helped make me the person I am today.

Table of Contents

List of Tables	iii
List of Figures	iv
Abstract	xii
Chapter 1 Introduction	1
1.1 Introduction.....	1
1.2 Background.....	2
1.3 Overview.....	5
Chapter 2 Test Data	7
2.1 Introduction.....	7
2.2 Test apparatus	7
2.3 Test specimens	8
2.4 Test setup	9
2.5 Experiment design	10
2.6 Test data	12
Chapter 3 Descriptive Analysis	31
3.1 Introduction.....	31
3.2 Minimum preload extraction.....	31
3.3 Sample statistics.....	40
Chapter 4 Test for Normality	50
4.1 Introduction.....	50
4.2 Probability plots	51
4.3 Probability plot correlation coefficient test for normality	53
Chapter 5 Population Predictions.....	58
5.1 Introduction.....	58
5.2 Confidence interval.....	58
5.3 Prediction intervals and tolerance intervals	64
Chapter 6 Interpretation of Results	66
Chapter 7 Conclusions	80
References.....	82

Bibliography	84
Appendices.....	85
Appendix A: M-files for minimum preload composite plots.....	86
Appendix B: M-files for sample statistic plots	106
Appendix C: Normal plots	128
Appendix D: M-files for PPCC tests for normality	141
Appendix E: M-files for confidence intervals	168

List of Tables

Table 1.2	Experiment run numbers for three locking levels.....	11
Table 1.2	Randomized test sequence.....	11
Table 4.1	Correlation coefficients R_p	55
Table 4.2	Empirical percentage points for correlation coefficient test based on Blom's plotting position.	56
Table 4.3	Correlation coefficients R_p for "Locking Heli-Coil with Braycote" with sample size 11.....	57

List of Figures

Figure 2.1	Schematic of test machine.	8
Figure 2.2	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 1.	12
Figure 2.3	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 2.	13
Figure 2.4	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 3.	13
Figure 2.5	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 4.	14
Figure 2.6	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 5.	14
Figure 2.7	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 6.	15
Figure 2.8	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 7.	15
Figure 2.9	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 8.	16
Figure 2.10	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 9.	16
Figure 2.11	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 10.	17
Figure 2.12	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 11.	17
Figure 2.13	Preload versus cycles for “Standard Heli-Coil with Braycote” run number 12.	18

Figure 2.14	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 13.....	18
Figure 2.15	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 14.....	19
Figure 2.16	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 15.....	19
Figure 2.17	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 16.....	20
Figure 2.18	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 17.....	20
Figure 2.19	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 18.....	21
Figure 2.20	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 19.....	21
Figure 2.21	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 20.....	22
Figure 2.22	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 21.....	22
Figure 2.23	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 22.....	23
Figure 2.24	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 23.....	23
Figure 2.25	Preload versus cycles for “Locking Heli-Coil with Braycote” run number 24.....	24
Figure 2.26	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 25.....	24
Figure 2.27	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 26.....	25
Figure 2.28	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 27.....	25

Figure 2.29	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 28.....	26
Figure 2.30	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 29.....	26
Figure 2.31	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 30.....	27
Figure 2.32	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 31.....	27
Figure 2.33	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 32.....	28
Figure 2.34	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 33.....	28
Figure 2.35	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 34.....	29
Figure 2.36	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 35.....	29
Figure 2.37	Preload versus cycles for “Standard Heli-Coil with Loctite” run number 36.....	30
Figure 3.1	Plot of all data for “Standard Heli-Coil with Braycote” runs.....	32
Figure 3.2	Plot of all data for “Locking Heli-Coil with Braycote” runs.....	33
Figure 3.3	Plot of all data for “Standard Heli-Coil with Loctite” runs.....	33
Figure 3.4	Composite plot for “Standard Heli-Coil with Braycote” runs.....	34
Figure 3.5	Composite plot for “Locking Heli-Coil with Braycote” runs.....	34
Figure 3.6	Composite plot for “Standard Heli-Coil with Loctite” runs.....	35
Figure 3.7	Comparison plot of the actual test data and extracted data for run number 7.....	37
Figure 3.8	Comparison plot of the actual test data and extracted data for run number 7 zoomed in.....	37

Figure 3.9	Comparison plot of the actual test data and extracted data for run number 13.	38
Figure 3.10	Comparison plot of the actual test data and extracted data for run 13 zoomed in.	38
Figure 3.11	Comparison plot of the actual test data and extracted data for run number 30.	39
Figure 3.12	Comparison plot of the actual test data and extracted data for run number 30 zoomed in.	39
Figure 3.13	Sample mean for “Standard Heli-Coil with Braycote” runs.	41
Figure 3.14	Sample mean for “Locking Heli-Coil with Braycote” runs.	41
Figure 3.15	Sample mean for “Standard Heli-Coil with Loctite” runs.	42
Figure 3.16	Sample median for “Standard Heli-Coil with Braycote” runs.	42
Figure 3.17	Sample median for “Locking Heli-Coil with Braycote” runs.	43
Figure 3.18	Sample median for “Standard Heli-Coil with Loctite” runs.	43
Figure 3.19	Sample upper and lower quartiles for “Standard Heli-Coil with Braycote” runs.	44
Figure 3.20	Sample upper and lower quartiles for “Locking Heli-Coil with Braycote” runs.	44
Figure 3.21	Sample upper and lower quartiles for “Standard Heli-Coil with Loctite” runs.	45
Figure 3.22	Upper and lower extents for “Standard Heli-Coil with Braycote” runs.	45
Figure 3.23	Upper and lower extents for “Locking Heli-Coil with Braycote” runs.	46
Figure 3.24	Upper and lower extents for “Standard Heli-Coil with Loctite” runs.	46
Figure 3.25	Mean, Median, upper and lower quartile, and extent curves for “Standard Heli-Coil with Braycote” runs.	47

Figure 3.26	Mean, Median, upper and lower quartile, and extent curves for “Locking Heli-Coil with Braycote” runs.	48
Figure 3.27	Mean, Median, upper and lower quartile, and extent curves for “Standard Heli-Coil with Loctite” runs.	48
Figure 4.1	Normal plot for “Standard Heli-Coil with Braycote” minimum preloads at 250 cycles.	51
Figure 4.2	Normal plot for “Locking Heli-Coil with Braycote” minimum preloads at 250 cycles.	52
Figure 4.3	Normal plot for “Standard Heli-Coil with Loctite” minimum preloads at 250 cycles.	52
Figure 5.1	95% Confidence Intervals for the population mean of “Standard Heli-Coil with Braycote” runs.	61
Figure 5.2	95% Confidence Intervals for the population mean of “Locking Heli-Coil with Braycote” runs.	61
Figure 5.3	95% Confidence Intervals for the population mean of “Standard Heli-Coil with Loctite” runs.	62
Figure 5.4	99% Confidence Intervals for the population mean of “Standard Heli-Coil with Braycote” runs.	62
Figure 5.5	99% Confidence Intervals for the population mean of “Locking Heli-Coil with Braycote” runs.	63
Figure 5.6	99% Confidence Intervals for the population mean of “Standard Heli-Coil with Loctite” runs.	63
Figure 6.1	Composite Plot (a), Descriptive Statistics Plot (b), 95% Confidence Interval Plot (c), and 99% Confidence Interval Plot (d) for the “Standard Heli-Coil with Braycote” runs.	67
Figure 6.2	Composite Plot (a), Descriptive Statistics Plot (b), 95% Confidence Interval Plot (c), and 99% Confidence Interval Plot (d) for the “Locking Heli-Coil with Braycote” runs.	68
Figure 6.3	Composite Plot (a), Descriptive Statistics Plot (b), 95% Confidence Interval Plot (c), and 99% Confidence Interval Plot (d) for the “Standard Heli-Coil with Loctite” runs.	69
Figure 6.4	Loosening regions for “Standard Heli-Coil with Braycote” plots.	70

Figure 6.5	Loosening regions for “Locking Heli-Coil with Braycote” plots.....	72
Figure 6.6	Grip Coil on “Locking Heli-Coil with Braycote” provided by [13].....	73
Figure 6.7	Free body diagram of fastener with prevailing torque.....	73
Figure 6.8	Locking minimum preload minus Std minimum preload versus loss of Locking preload.....	75
Figure 6.9	Loosening regions for “Standard Heli-Coil with Loctite” plots.	76
Figure 6.10	Loctite minimum preload minus Std minimum preload versus loss of Loctite preload.....	78
Figure 6.11	95 percent Confidence Intervals for all three locking levels.	79
Figure 6.12	99 percent Confidence Intervals for all three locking levels.	79
Figure C.1	Normal plot for “Standard Heli-Coil with Braycote” preloads at 10 cycles.....	128
Figure C.2	Normal plot for “Standard Heli-Coil with Braycote” preloads at 250 cycles.....	128
Figure C.3	Normal plot for “Standard Heli-Coil with Braycote” preloads at 500 cycles.....	129
Figure C.4	Normal plot for “Standard Heli-Coil with Braycote” preloads at 750 cycles.....	129
Figure C.5	Normal plot for “Standard Heli-Coil with Braycote” preloads at 1000 cycles.....	130
Figure C.6	Normal plot for “Standard Heli-Coil with Braycote” preloads at 1250 cycles.....	130
Figure C.7	Normal plot for “Locking Heli-Coil with Braycote” preloads at 10 cycles.....	131
Figure C.8	Normal plot for “Locking Heli-Coil with Braycote” preloads at 250 cycles.....	131
Figure C.9	Normal plot for “Locking Heli-Coil with Braycote” preloads at 500 cycles.....	132

Figure C.10	Normal plot for “Locking Heli-Coil with Braycote” preloads at 750 cycles.....	132
Figure C.11	Normal plot for “Locking Heli-Coil with Braycote” preloads at 1000 cycles.....	133
Figure C.12	Normal plot for “Locking Heli-Coil with Braycote” preloads at 1250 cycles.....	133
Figure C.13	Normal plot for “Locking Heli-Coil with Braycote” preloads at 1500 cycles.....	134
Figure C.14	Normal plot for “Locking Heli-Coil with Braycote” preloads at 1750 cycles.....	134
Figure C.15	Normal plot for “Locking Heli-Coil with Braycote” preloads at 2000cycles.....	135
Figure C.16	Normal plot for “Locking Heli-Coil with Braycote” preloads at 2250 cycles.....	135
Figure C.17	Normal plot for “Standard Heli-Coil with Loctite” preloads at 10 cycles.....	136
Figure C.18	Normal plot for “Standard Heli-Coil with Loctite” preloads at 250 cycles.....	136
Figure C.19	Normal plot for “Standard Heli-Coil with Loctite” preloads at 500 cycles.....	137
Figure C.20	Normal plot for “Standard Heli-Coil with Loctite” preloads at 750 cycles.....	137
Figure C.21	Normal plot for “Standard Heli-Coil with Loctite” preloads at 1000 cycles.....	138
Figure C.22	Normal plot for “Standard Heli-Coil with Loctite” preloads at 1250 cycles.....	138
Figure C.23	Normal plot for “Standard Heli-Coil with Loctite” preloads at 1500 cycles.....	139
Figure C.24	Normal plot for “Standard Heli-Coil with Loctite” preloads at 1750 cycles.....	139

Figure C.25	Normal plot for “Standard Heli-Coil with Loctite” preloads at 2000 cycles.....	140
Figure C.26	Normal plot for “Standard Heli-Coil with Loctite” preloads at 2250 cycles.....	140

Statistical Analysis of Fastener Vibration Life Tests

Christopher Cheatham

ABSTRACT

This thesis presents methods to statistically quantify data from fastener vibration life tests. Data from fastener vibration life tests with secondary locking features of threaded inserts is used. Threaded inserts in three different configurations are examined: no locking feature, prevailing torque locking feature, and adhesive locking feature. Useful composite plots were developed by extracting minimum preloads versus cycles from test data. Minimum preloads were extracted due to the overlapping of varying test data and because the minimum preload is of most interest in such tests.

In addition to composite plots, descriptive statistics of the samples were determined including mean, median, quartiles, and extents. These descriptive statistics were plotted to illustrate variability within a sample as well as variability between samples. These plots also reveal that characteristics of loosening for a sample, such as preload loss and rates of preload loss, are preserved when summarizing such tests. Usually fastener vibration life tests are presented and compared with one test sample, which is why statistically quantifying them is needed and important.

Methods to predict the sample population have been created as well. To predict populations, tests to determine the distribution of the sample, such as probability plots and probability plot correlation coefficient, have been conducted. Once samples were determined to be normal, confidence intervals were created for test samples, which

provides a range of where the population mean should lie. It has been shown that characteristics of loosening are preserved in the confidence intervals. Populations of fastener vibration life tests have never before been presented or created.

The evaluation of loosening has been conducted for fastener vibration life tests in the past with plots of one test sample; however, in this work statistically quantified results of multiple tests were used. This is important because evaluating loosening with more than one test sample can determine variation between tests. It has been found that secondary locking features do help reduce the loss of preload. The prevailing torque secondary locking feature is found to be more effective as preload is lost. The best secondary locking feature has been found to be the adhesive.

Chapter 1

Introduction

1.1 Introduction

On August 20, 2007, China Airlines flight CI120 landed at Naha airport in Japan after a flight from Taiwan. Shortly after the 737-800 pulled into its parking spot the ground crew informed the captain of the plane that there was a fuel leak. The 165 occupants of the plane immediately evacuated. Captured on video was the horrific explosion which occurred at the instant that the pilot jumped from the cockpit. Thankfully, there were no deaths and no injuries caused by the explosion. Later officials explained that a loose bolt had come off of the slat on one of the main wings and pierced through the fuel tank, causing the fuel to leak out.

This is a prime example of how loose bolts cause costly problems and sometimes even catastrophic failures. Though loosening is still a problem, threaded fasteners are used in almost all modern applications. The advantages of threaded fasteners outweigh the disadvantages. One of the most useful advantages of a threaded fastener is that it has the ability to be disassembled.

It has been understood for several decades [1] that threaded fasteners become loose from dynamic loads of vibration and shock. Enduring and resisting these dynamic loads is key for fasteners to retain their clamping force. Usually the friction at the head and threads is the primary force that retains the clamping force (preload). In addition to the friction at the head and threads, some fasteners use secondary locking features to help

maintain preload. Some of the secondary locking features include: lock wire, adhesives, pins, and lock washers. However, even with these secondary locking features loosening can occur.

Typically vibration life tests of fasteners produce loosening curves or preload versus cycle data, from which plots are created. In the literature, there is limited data for these fastener life tests, and there are no current methods that quantify the results of such tests. Since these tests can and have been run, there is a need for a formal test that will quantify their results. To contribute to this area, this thesis develops methods that statistically quantify and predict the preload loss over cycles for a fastener.

1.2 Background

For more than half of a century [1] researchers have studied the loosening of threaded fasteners resulting from vibration. One researcher in the 1960's named Gerhard Junker, showed that transverse vibration (shear loading) was the most severe condition for self-loosening of thread fasteners. With the use of hysteresis curves, he [2] determined that loosening results from gross slip (friction loss) at the head and thread interfaces.

More recently, Pai and Hess [3, 4] extended the work of Junker and showed that there are four possible conditions that allow loosening to exist: localized head slip and localized thread slip; localized head slip and complete thread slip; complete head slip and localized thread slip; complete head slip and complete thread slip. Junker showed that there was complete slip at the head and the threads, but Pai and Hess were able to show that there is also localized slip as well as complete slip in the head and the threads when

loosening is occurring. Pai and Hess were able to show this through finite element models and used hysteresis plots as well as preload versus cycles plots. Also with the use of the hysteresis plots, Pai and Hess were able to show in the preloads versus cycles plots where localized slip and complete slip occurs for a particular test run. These preloads versus cycles plots helped to present how the loss of preload coincides with loosening of a threaded fastener.

In other work, Finkelston [5] measured preloads versus cycles data from transverse vibration machines to study and compare preload loss for different threaded fastener conditions. He presents plots that have one test run with another test run of a different configuration. Finkelston compared the preload loss of threaded fasteners with different initial preloads, fasteners with different thread pitch and the effects of prevailing torque locknuts. Aided by the preload-versus-cycle plots he provides reasons of why the preload loss occurs differently in the different configurations.

In another study, Sanclemente and Hess [6] use preload versus time data from a transverse vibration test to determine preload loss for threaded fasteners that have multiple high level and low level factors. With the preload loss information, they created a statistical model that determined which of the level factors had the most influence in loosening.

Nord-Lock [7] presents test results of transverse vibration tests in the form of preloads versus time to show that their product retains most of the initial preload compared to other available products. Like Finkelston, these plots present comparison runs between the company's product against other available products. Similarly, Faroni [8] provides reasons for self-loosening of threaded fasteners and presents a plot that

contains the range of preload loss over the duration of a vibration test for different quality fasteners (i.e., best-Aircraft fasteners, poorest-Commercial fasteners).

These methods of studying preload loss in the form of preloads versus cycles are beneficial in assisting in the understanding the self-loosening of threaded fasteners. The use of preloads versus cycles plots are useful when comparing the loosening between two different threaded fastener configurations. Most of these works use preloads versus cycles plots, but only use the preload loss for individual test samples to determine which configuration loosens less. Finkelston [5] provided explanations of loosening using the preload versus cycles plots, but he described the results from one test run. Pai and Hess [3] also describes how a fastener loosens with the preloads versus cycles plots in a plot that shows one test run; however Pai and Hess used more than one test run to develop their conclusions. Methods that will quantify test data of fasteners that undergo transverse vibration tests are needed in order to quantify the loosening when using preload-versus-cycle plots.

In a technical paper produced by the Society of Manufacturing Engineers [9], it was shown how statistics can be used to report, design, or even predict the behavior of bolted joints. By using the test sample data, one can statistically describe the population. In testing bolted joints, the technical paper suggests that the population could consist of: all of the bolts in a given joint and all of the bolts in all such joints in a given assembly. This technical paper provides ways to statistically evaluate the approximate torque for a given preload, nut factors, relaxation, and friction; however it never provides methods of statically quantifying a vibration life test of threaded fasteners.

Similar to vibration life tests of fasteners, stress life tests or fatigue tests present fatigue data in plots of stress versus cycles to failure. These plots, also known as S-N curves are created by fitting the observed data, which was collected from tests that have sample sizes greater than one [10]. Collins [10] explains that because of the scatter of fatigue life data, a statistical description of fatigue failure data should be used to make appropriate S-N curves. Using statistics, curves of constant probability of failure can be created and are seen in S-N-P curves. S-N-P curves show many curves that present different probabilities of failure at a percent of reliability. This shows that statistically quantifying the variation of vibration life tests of fasteners can be done. The variation between fastener life tests should be quantified statistically to help explain the loosening occurring in the fastener vibration life test.

All of this work [2-9] uses preload versus cycles data from vibration life tests. Most of these works use data from individual test runs to develop conclusions. Because individual test runs will vary from one run to the next, a method to quantify and predict the loss of preload that incorporates that variation is needed. Such quantified results can be used to help develop more meaningful conclusions to describe a population with a sample size greater than one. This thesis contributes to this void in the literature. This thesis provides methods that statistically quantify the preload versus cycles data for a reasonable sample size and predicts the loosening of threaded fasteners for a population.

1.3 Overview

This thesis presents methods that statistically quantify the loosening curve life test data. Chapter 2 presents how fastener vibration life test data was obtain in this thesis so

that methods to quantify the data from such test can be created. To quantify the test results, Chapter 3 explains the techniques used to plot all of the minimum preloads versus cycles for a sample. This chapter also presents how plots of the sample statistics were created. Before statistical quantifications can be made, the distribution from which the data belongs must be identified. Chapter 4 discusses the tests of normality that were used to determine if the sample data was normally distributed. After the data was identified to be normally distributed, predictions of the population are created in Chapter 5. The predictions of the population are in the form of confidence intervals, which provided an interval in which the population's mean will lie at a percent confidence. To demonstrate how these quantified results can describe the underlying mechanisms of fastener loosening, Chapter 6 uses the results developed in this thesis to describe the various mechanisms of loosening. Finally, in Chapter 7 the conclusions are stated.

Chapter 2

Test Data

2.1 Introduction

In this thesis, test data is used to predict populations that describe loosening for three different locking features. This chapter presents the data used in this work as well as information on how the data was acquired. Even though this test data was not measured as part of this thesis, the test apparatus, test specimens, and how the tests were run are described.

2.2 Test apparatus

The test apparatus typically used to study loosening is shown in Figure 2.1. This machine applied a cyclic shear load by means of an adjustable eccentric that was driven by a five HP motor. The adjustable eccentric was connected to the top plate by an arm, and the top plate was fastened to a fixed base through a threaded insert using a test screw. The test screw clamped the top plate to the fixed base through a washer, a cone, and load cell fixture as shown in Figure 2.1. The load cell fixture is set in a preload a load cell and the cone was set in the top plate. Roller bearings were located between the top plate and the fixed base; which allowed the top plate to move when a shear load was applied through the arm from the eccentric. Measurements of shear force on the top plate as well as preload of the screw were measured by load cells. A Linear Variable Differential

Transformer, or LVDT for short, was used to measure the transverse displacement of the top plate.

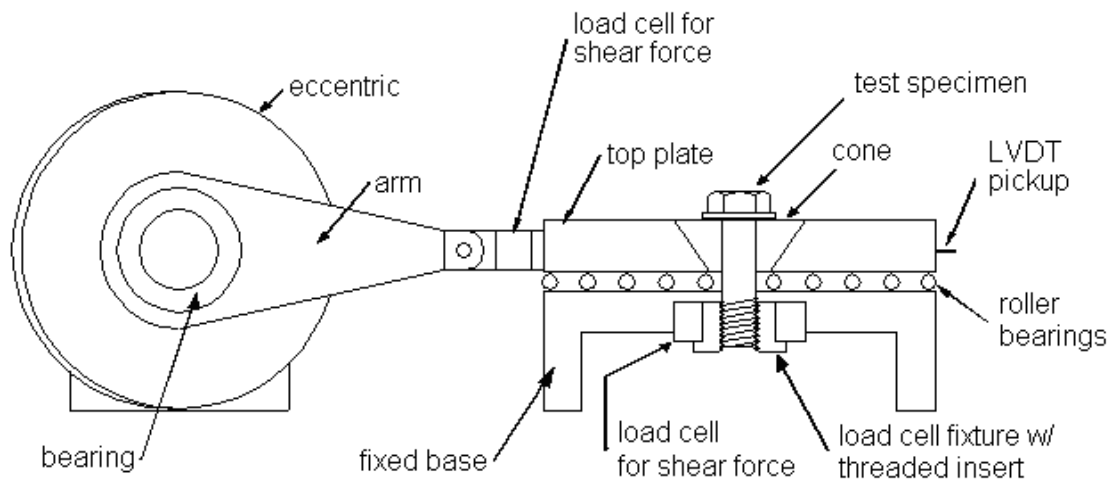


Figure 2.1 Schematic of test machine.

2.3 Test specimens

The test data used in this work was obtained from the apparatus described above with NAS 1004 ¼ -28 UNJF-3A hex head screws [11] with:

1. Standard free-running Heli-Coil inserts with Braycote 601 EF high vacuum grease
2. Locking Heli-Coil inserts with Braycote 601 EF high vacuum grease, and
3. Standard free-running Heli-Coil inserts with Loctite 242 threadlocker.

These three different locking levels were tested twelve times each, providing data in sets of twelve. The specifications for the screw, Heli-Coil inserts, and washers used in the tests are as follows:

1. Thirty-six NAS 1004 ¼ -28 UNJF-3A, 2.356 inch long, hex head screws, made of A286 Stainless steel [11]
2. Thirty-six NAS 1149-C0463R washers for ¼ inch screw made of corrosion resistant steel with a passivated finish [12]
3. Twenty-four MS124696, 0.375 inch long, standard, free-running Heli-Coil inserts, made of stainless steel [13]
4. Twelve MS21209-F4-15, 0.375 inch long, locking Heli-Coil inserts, made of 304 stainless steel [14]

The thirty-six cones and load cell fixtures were made of 15-5 stainless steel and heat treated to RC35. The cones had a thru-hole for the test screw to run through, and the load cell fixtures had tapped holes ready for the Heli-Coils to be installed.

2.4 Test setup

The tests were set up to provide significant loosening for the “Standard Heli-Coil with Braycote” configuration over a finite number of cycles without causing screws to break for any of the locking levels. Before the tests were run, the thirty-six cones and thirty-six load cell fixtures were thoroughly pre-cleaned in an ultrasonic cleaner with MEK (methyl ethyl ketone). After the pre-cleaning the Heli-Coil inserts were installed

into the load cell fixture. All parts, including the test screw, washer, cone, and load cell with the Heli-Coil installed, were cleaned again in an ultrasonic cleaner with MEK.

When the parts are installed in the test apparatus, Braycote grease was applied under the screw head and washer for all the tests. Tests for “Standard Heli-Coil with Braycote” runs and “Locking Heli-Coil with Braycote” have Braycote applied to their threads, and then are tightened to 2,400 lbs of preload. The tests for “Standard Heli-Coil with Loctite” runs have Loctite applied to their threads, and then are tightened to 2,400 lbs of preload.

After test specimens were fully assembled, the test machine was run at 15 Hz with a 0.12 inch (3 mm) eccentric. The data obtained from each test was collected at 51.2 samples/second for a total of 8,192 data points for each measured variable (displacement, preload, and shear force).

2.5 Experiment design

The tests were designed for replication and randomization for a single-factor experiment. The single-factor in the tests was the secondary locking feature. There were three levels of this factor: “Standard Heli-Coil with Braycote”, “Locking Heli-Coil with Braycote”, and “Standard Heli-Coil with Loctite”. Replication was achieved by using twelve replicas for each of the locking levels for a total of thirty-six runs. Run numbers for each level are presented in Table 2.1. Randomization of the tests ensured that the observations were independently distributed random variables and it also averaged out the effects of extraneous factors [15]. The test sequence was randomized using the Excel “rand” function and sort tool. Table 2.2 lists the test sequence for the runs.

Table 1.1 Experiment run numbers for three locking levels.

Locking Level	Experiment Run Number											
Std Heli-Coil w/ Braycote	1	2	3	4	5	6	7	8	9	10	11	12
Locking Heli-Coil w/ Braycote	13	14	15	16	17	18	19	20	21	22	23	24
Std Heli-Coil w/ Loctite	25	26	27	28	29	30	31	32	33	34	35	36

Table 1.2 Randomized test sequence.

Test sequence	Run Number	Locking level
1	7	Std Heli-coil w/ Brycote
2	13	Locking Heli-coil w/ Brycote
3	1	Std Heli-coil w/ Brycote
4	10	Std Heli-coil w/ Brycote
5	30	Std Heli-coil w/ Loctite
6	27	Std Heli-coil w/ Loctite
7	6	Std Heli-coil w/ Brycote
8	3	Std Heli-coil w/ Brycote
9	16	Locking Heli-coil w/ Brycote
10	22	Locking Heli-coil w/ Brycote
11	5	Std Heli-coil w/ Brycote
12	28	Std Heli-coil w/ Loctite
13	26	Std Heli-coil w/ Loctite
14	21	Locking Heli-coil w/ Brycote
15	12	Std Heli-coil w/ Brycote
16	14	Locking Heli-coil w/ Brycote
17	31	Std Heli-coil w/ Loctite
18	24	Locking Heli-coil w/ Brycote
19	29	Std Heli-coil w/ Loctite
20	36	Std Heli-coil w/ Loctite
21	2	Std Heli-coil w/ Brycote
22	34	Std Heli-coil w/ Loctite
23	25	Std Heli-coil w/ Loctite
24	19	Locking Heli-coil w/ Brycote
25	4	Std Heli-coil w/ Brycote
26	8	Std Heli-coil w/ Brycote
27	23	Locking Heli-coil w/ Brycote
28	9	Std Heli-coil w/ Brycote
29	32	Std Heli-coil w/ Loctite
30	17	Locking Heli-coil w/ Brycote
31	33	Std Heli-coil w/ Loctite
32	15	Locking Heli-coil w/ Brycote
33	11	Std Heli-coil w/ Brycote
34	35	Std Heli-coil w/ Loctite
35	18	Locking Heli-coil w/ Brycote
36	20	Std Heli-coil w/ Loctite

2.6 Test data

The test data obtained from these test are in the form of plots that show preload versus cycles. There are 8192 data points for each test that was run. The plots allow individual test runs to be viewed, and allows preload loss over a finite number of cycles to be determined for each test. Comparisons of preload loss can be made from one plot to the next. From the test data that was acquired, comparisons can be made for each of the thirty-six tests. Figures 2.2 -2.37 show the data that was obtained.

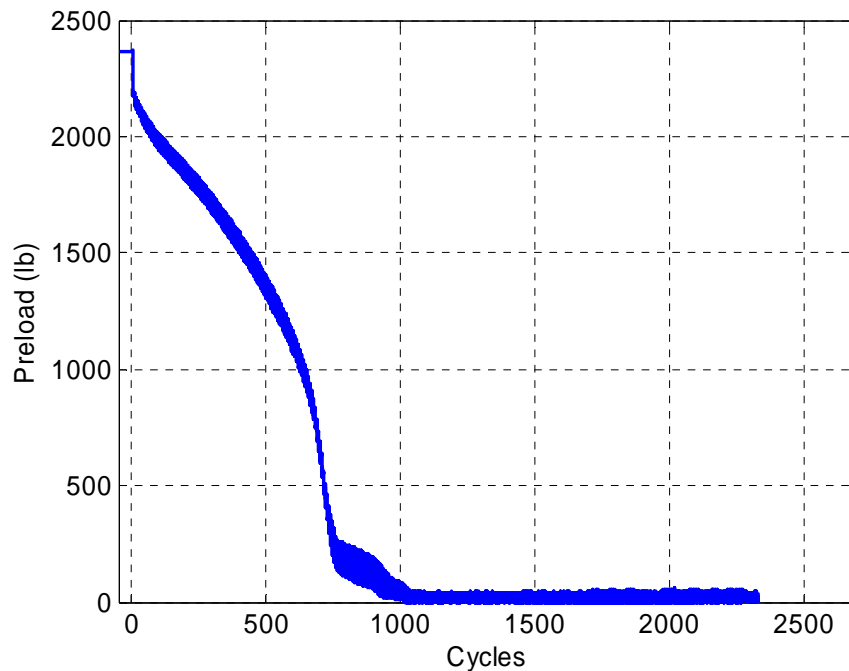


Figure 2.2 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 1.

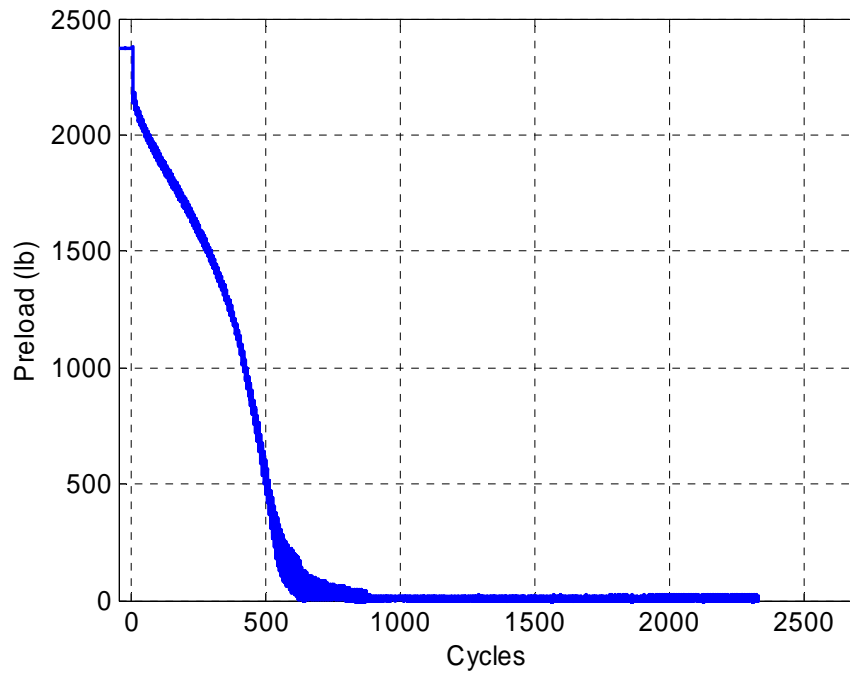


Figure 2.3 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 2.

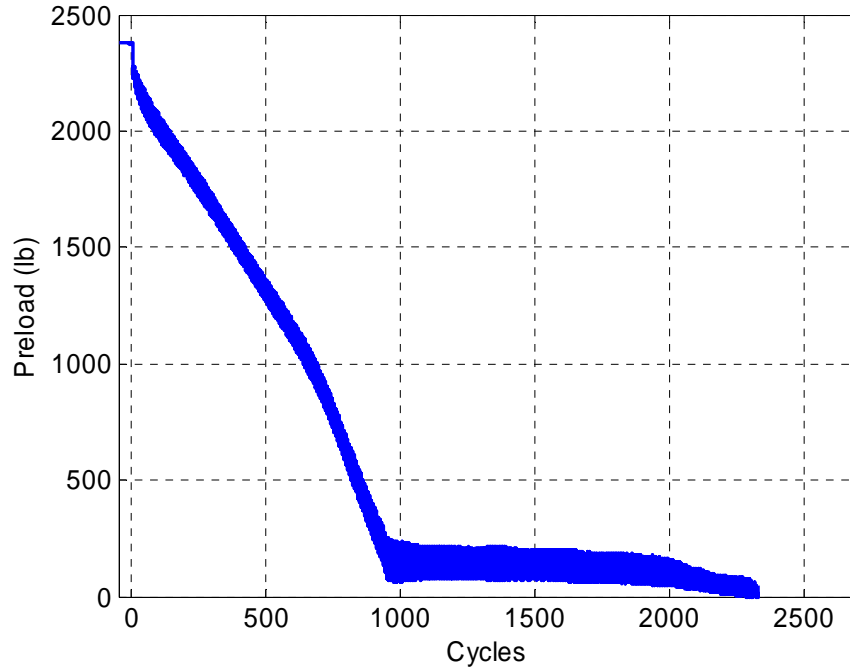


Figure 2.4 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 3.

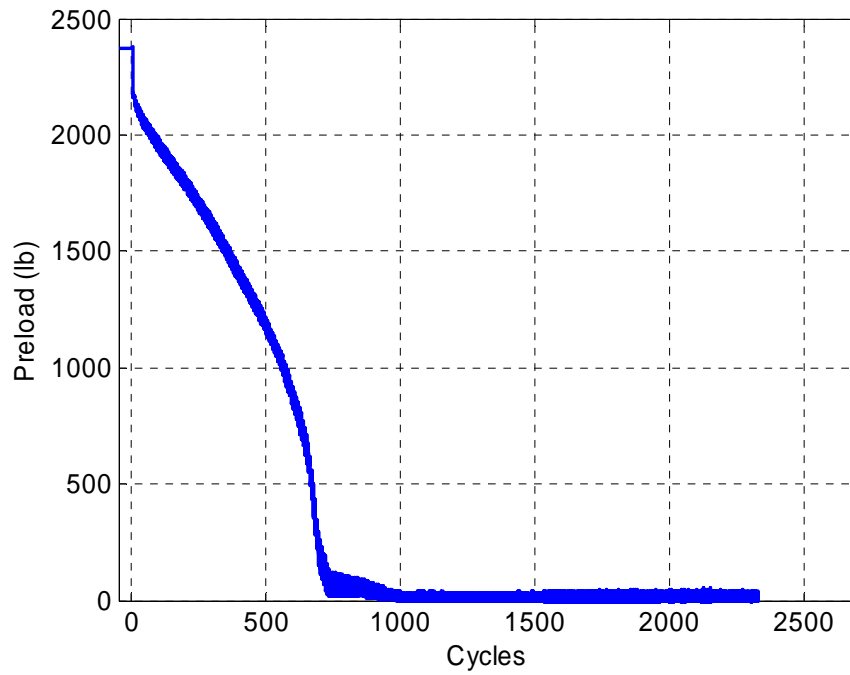


Figure 2.5 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 4.

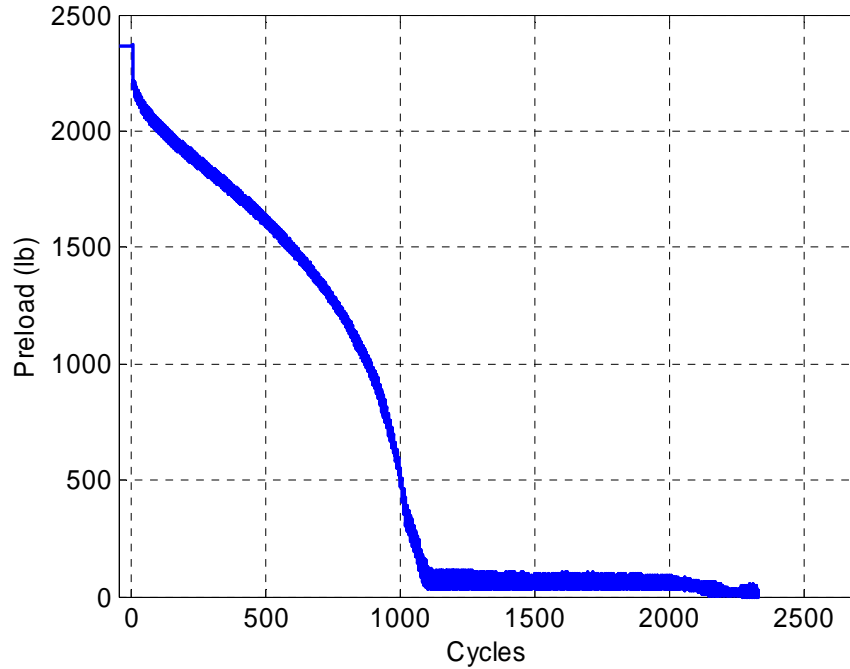


Figure 2.6 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 5.

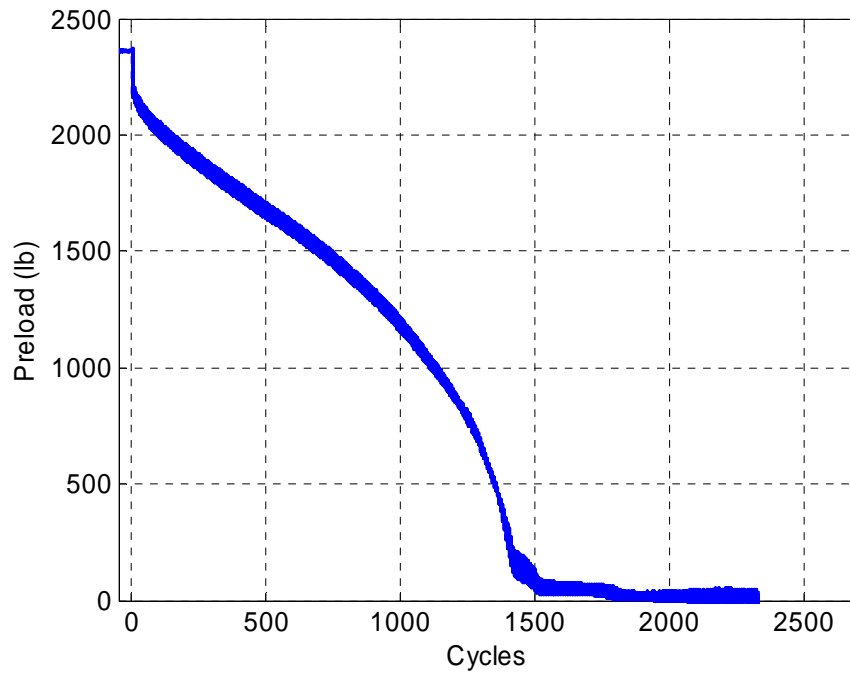


Figure 2.7 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 6.

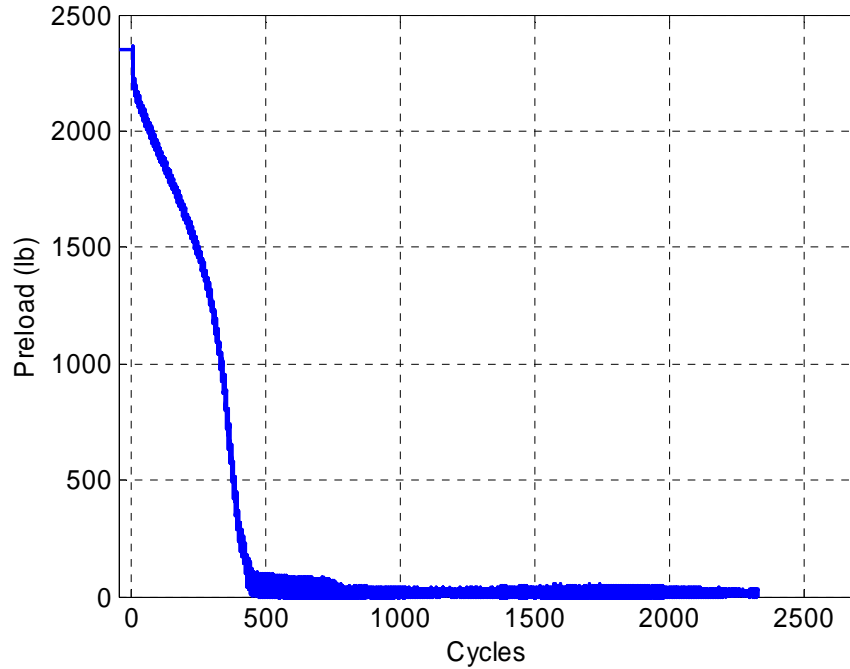


Figure 2.8 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 7.

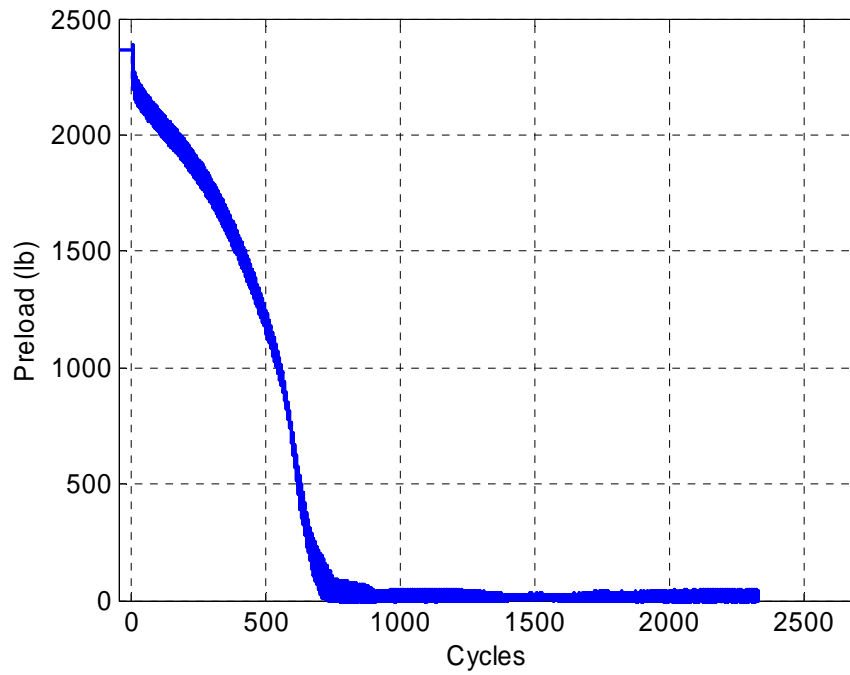


Figure 2.9 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 8.

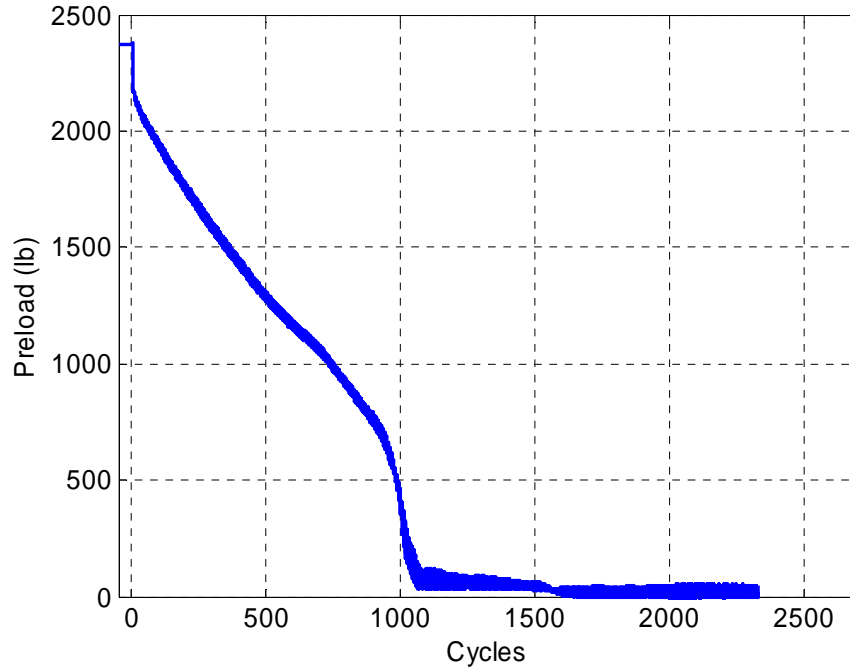


Figure 2.10 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 9.

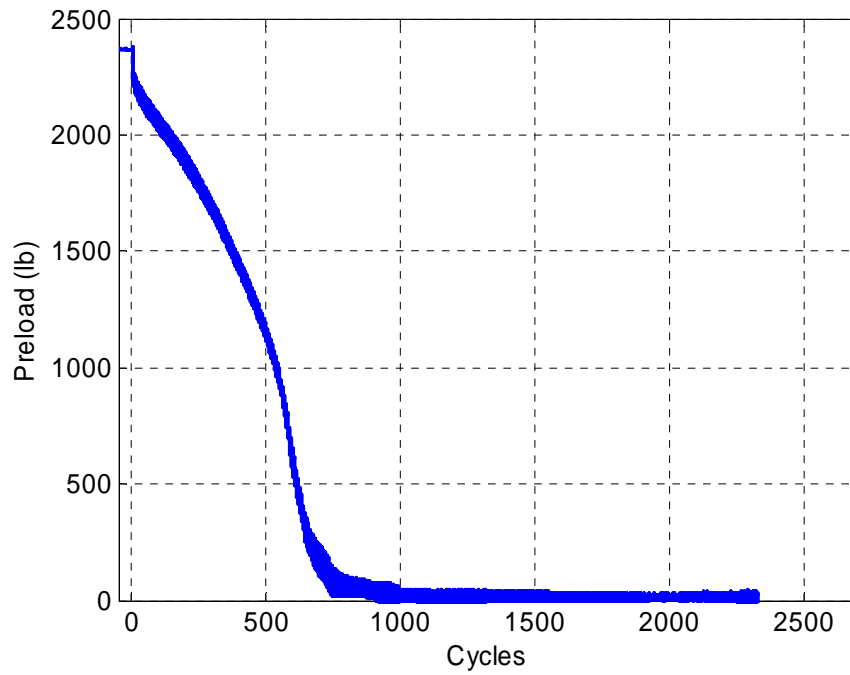


Figure 2.11 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 10.

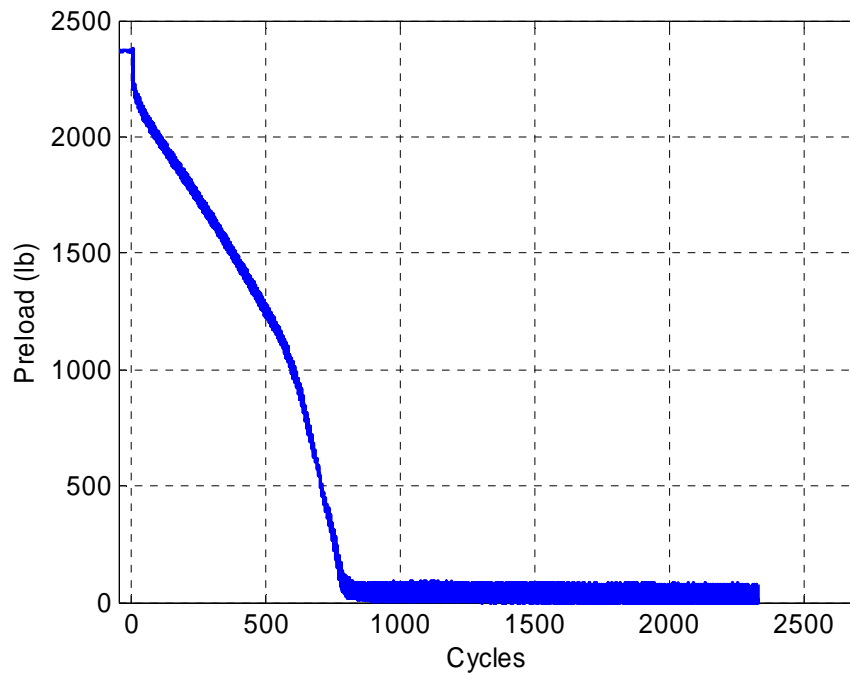


Figure 2.12 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 11.

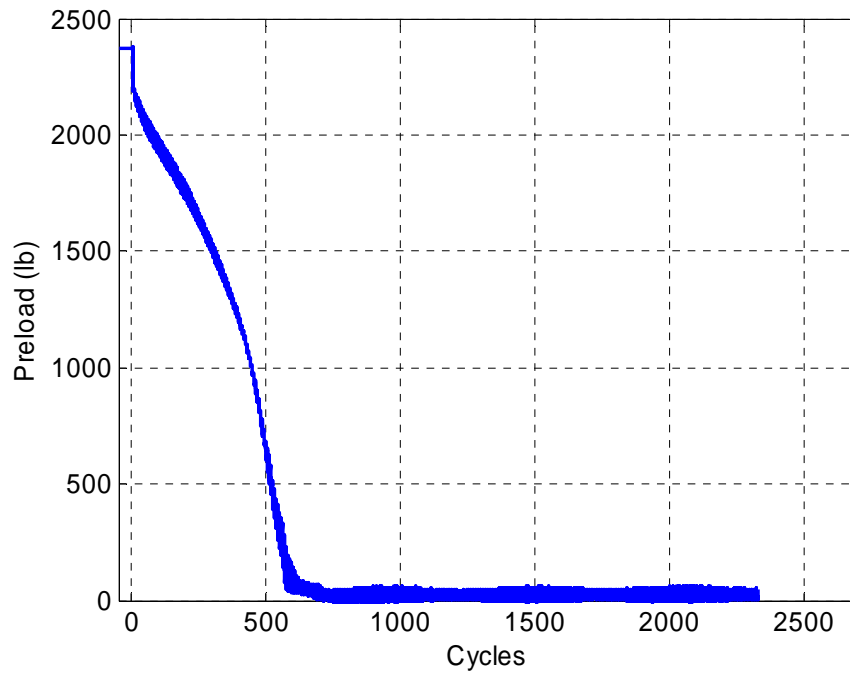


Figure 2.13 Preload versus cycles for “Standard Heli-Coil with Braycote” run number 12.

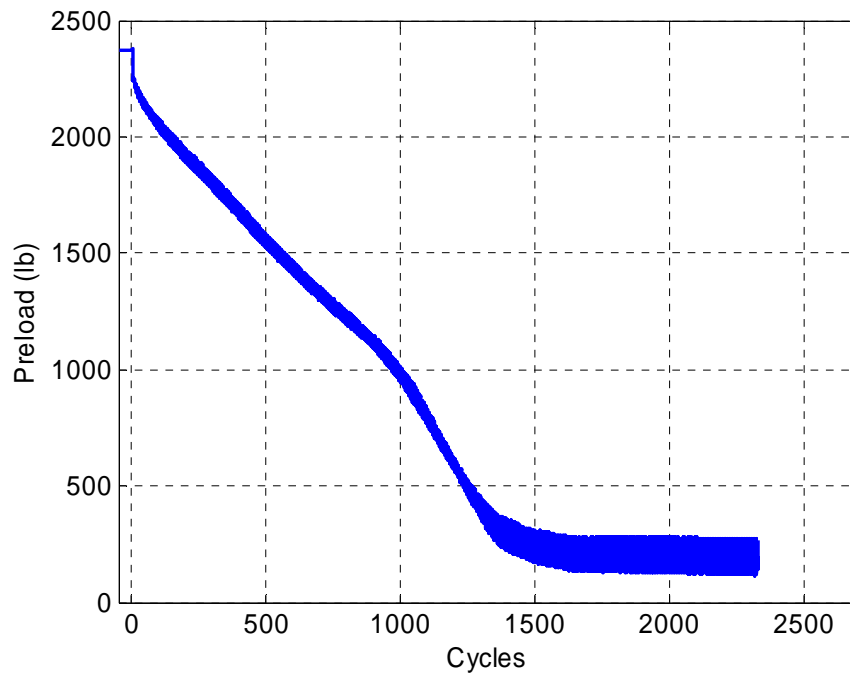


Figure 2.14 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 13.

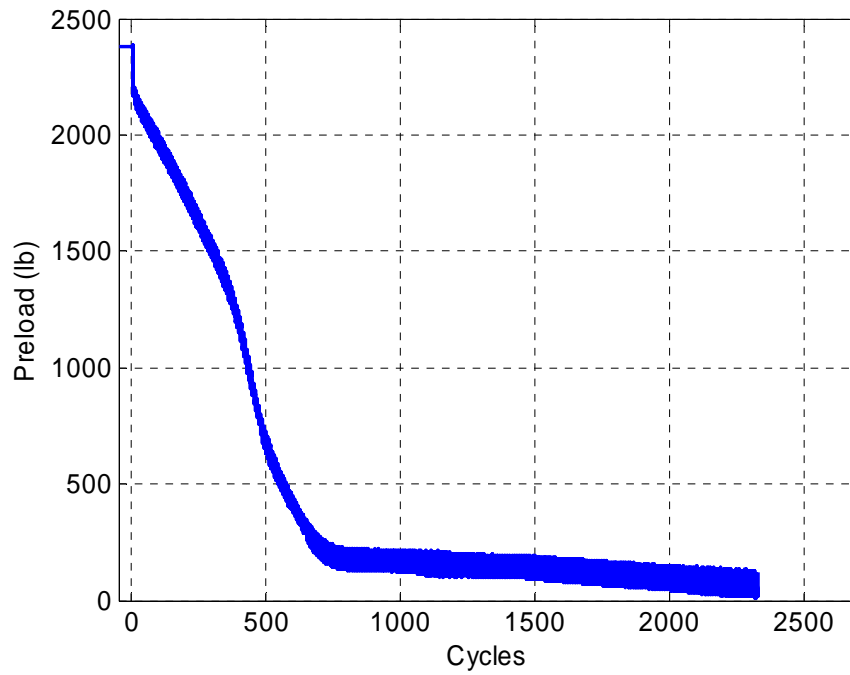


Figure 2.15 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 14.

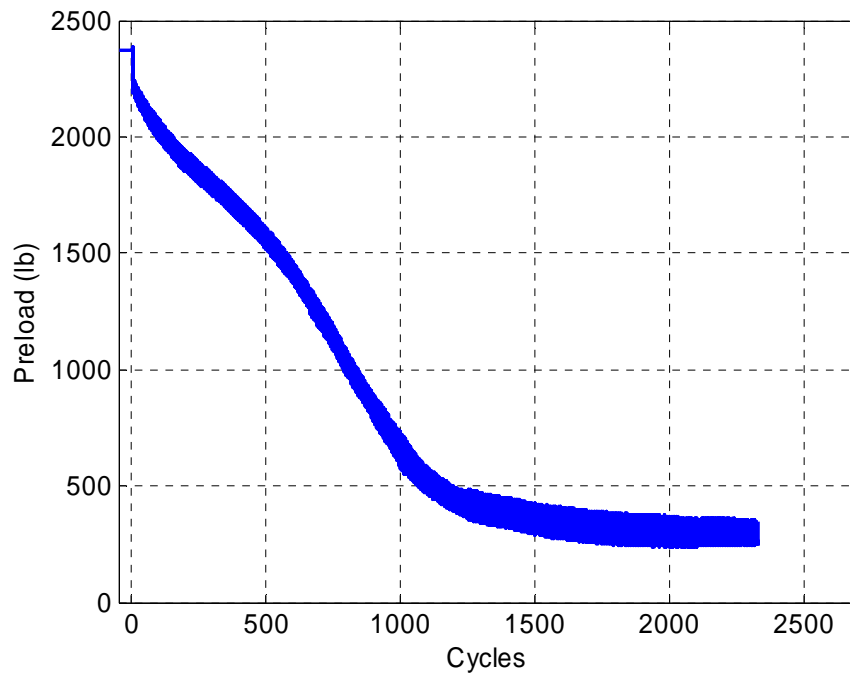


Figure 2.16 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 15.

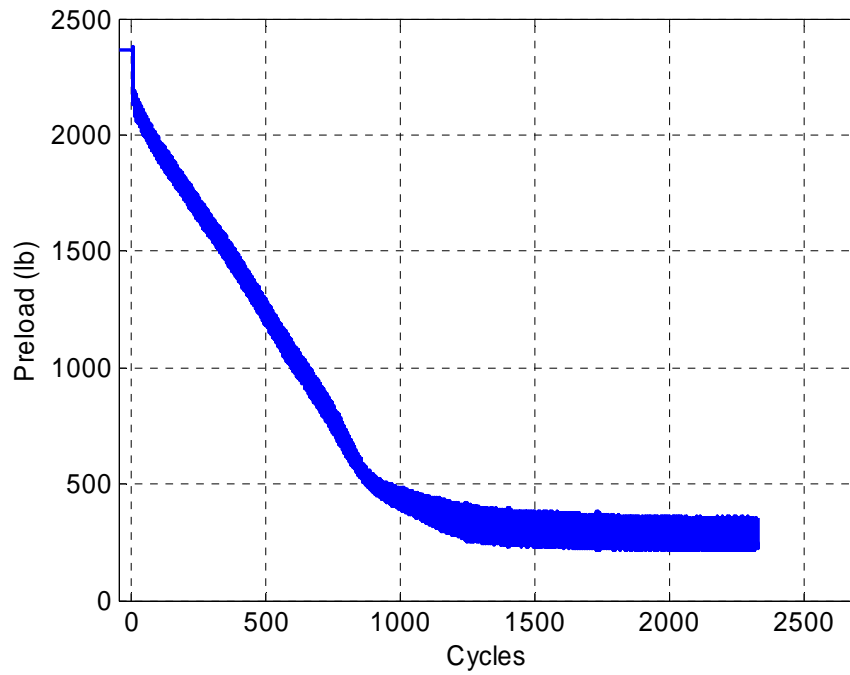


Figure 2.17 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 16.

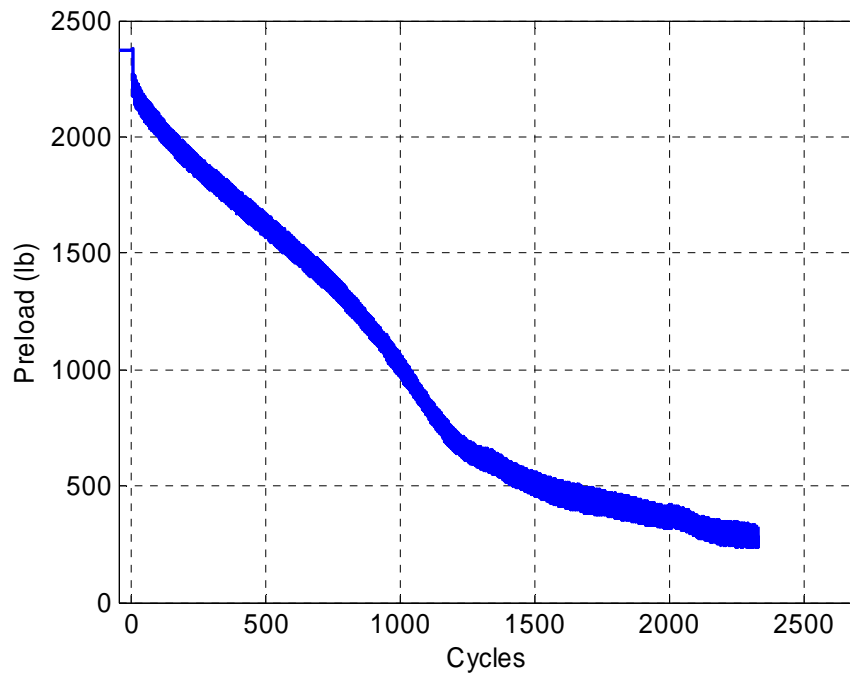


Figure 2.18 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 17.

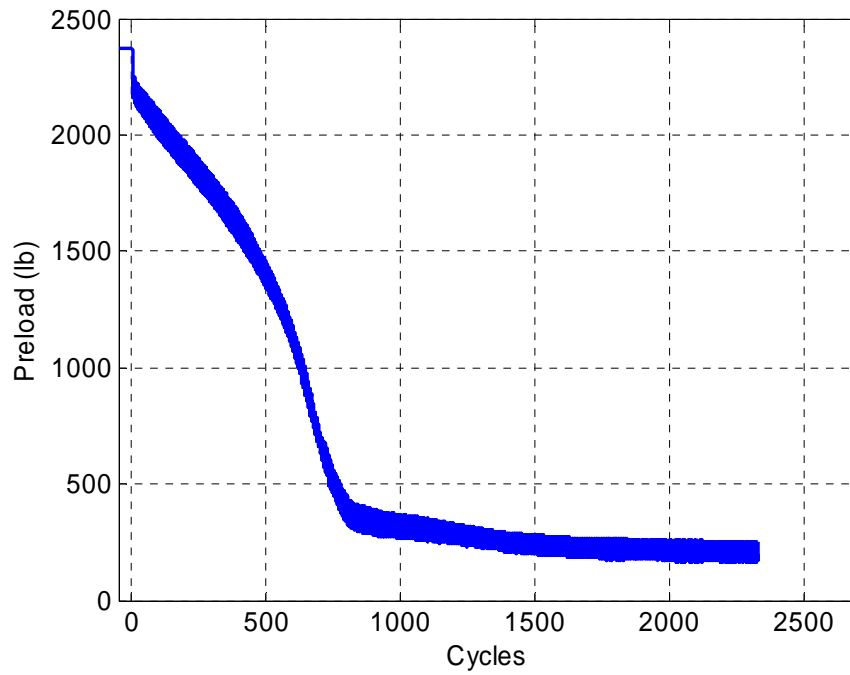


Figure 2.19 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 18.

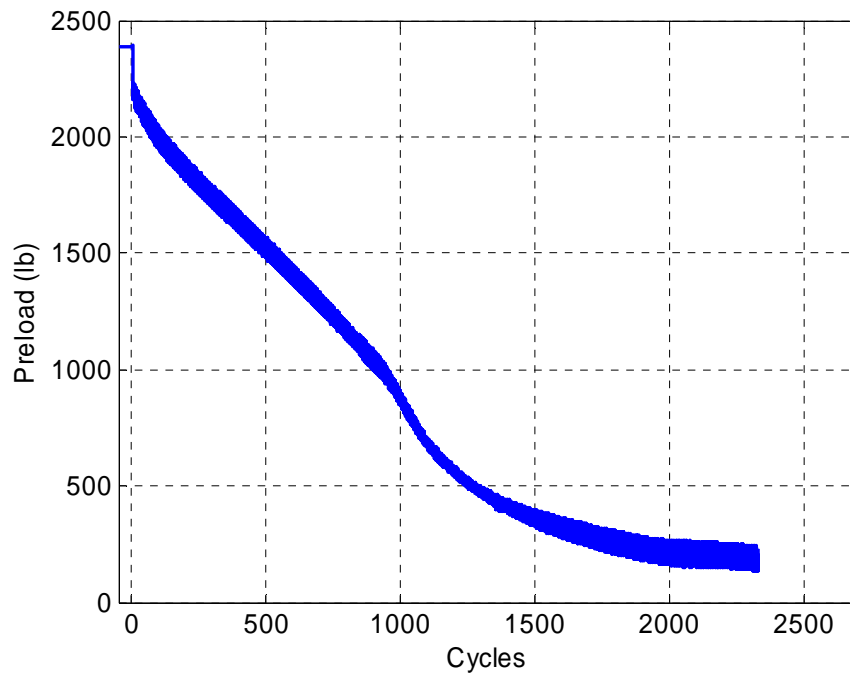


Figure 2.20 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 19.

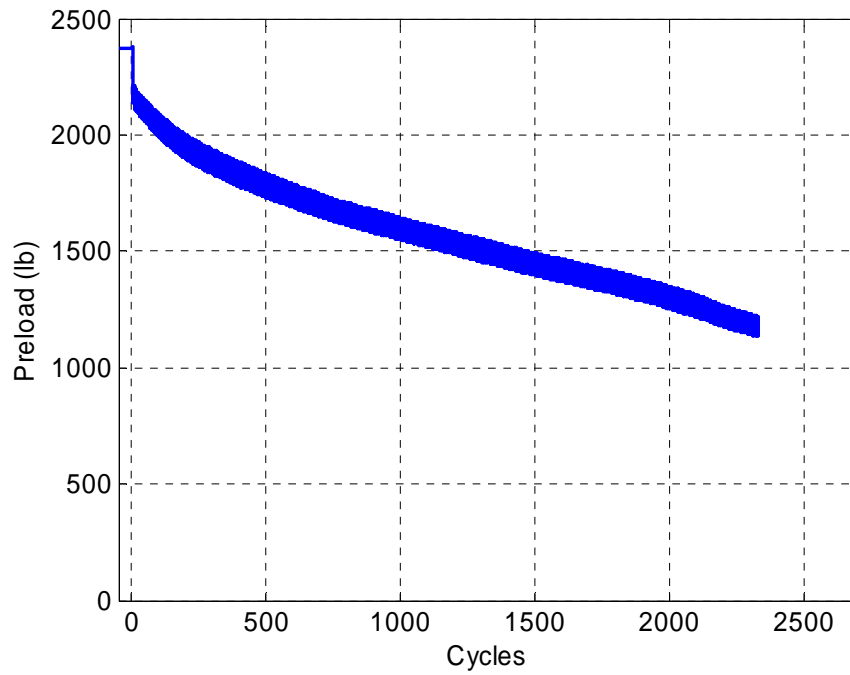


Figure 2.21 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 20.

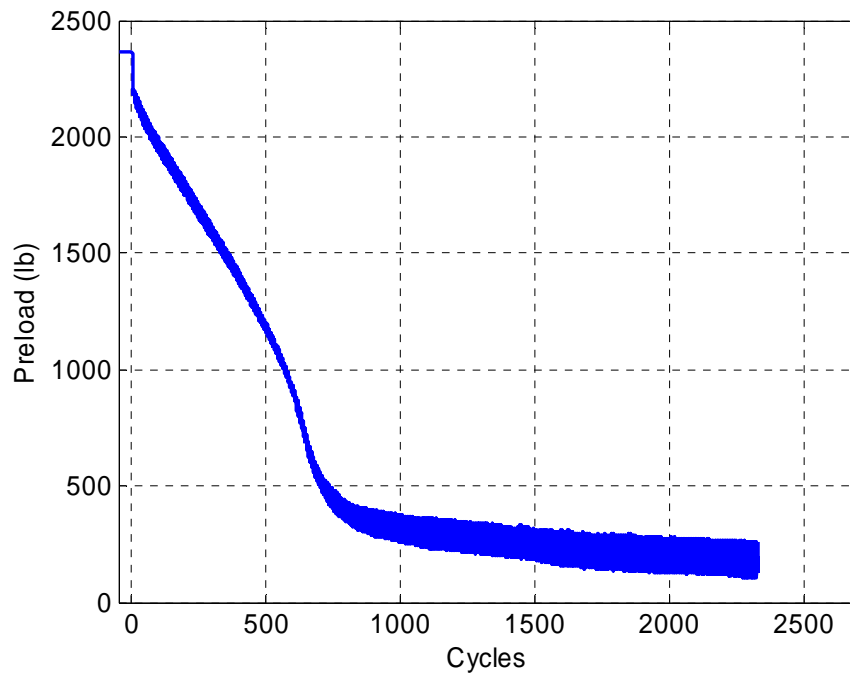


Figure 2.22 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 21.

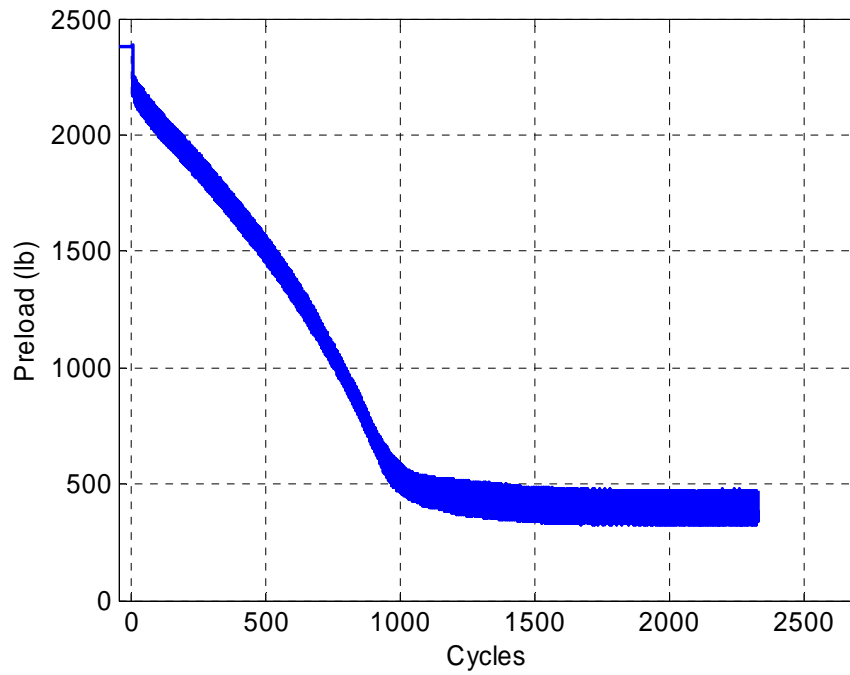


Figure 2.23 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 22.

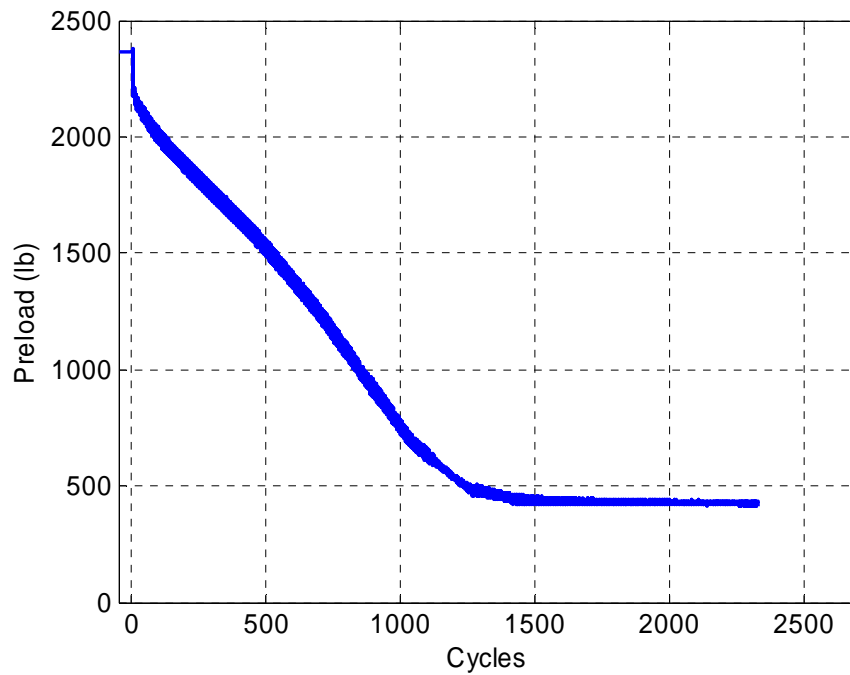


Figure 2.24 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 23.

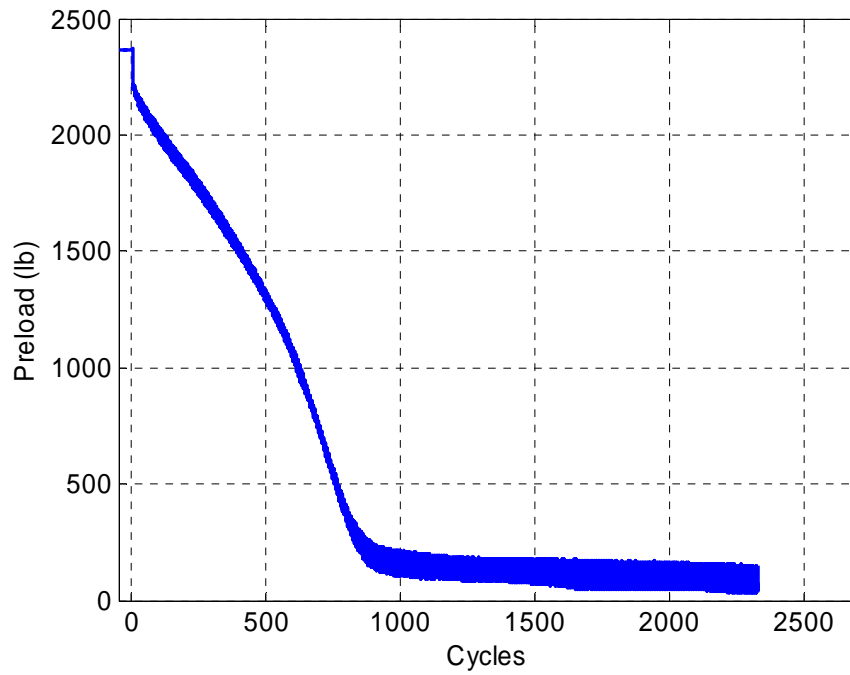


Figure 2.25 Preload versus cycles for “Locking Heli-Coil with Braycote” run number 24.

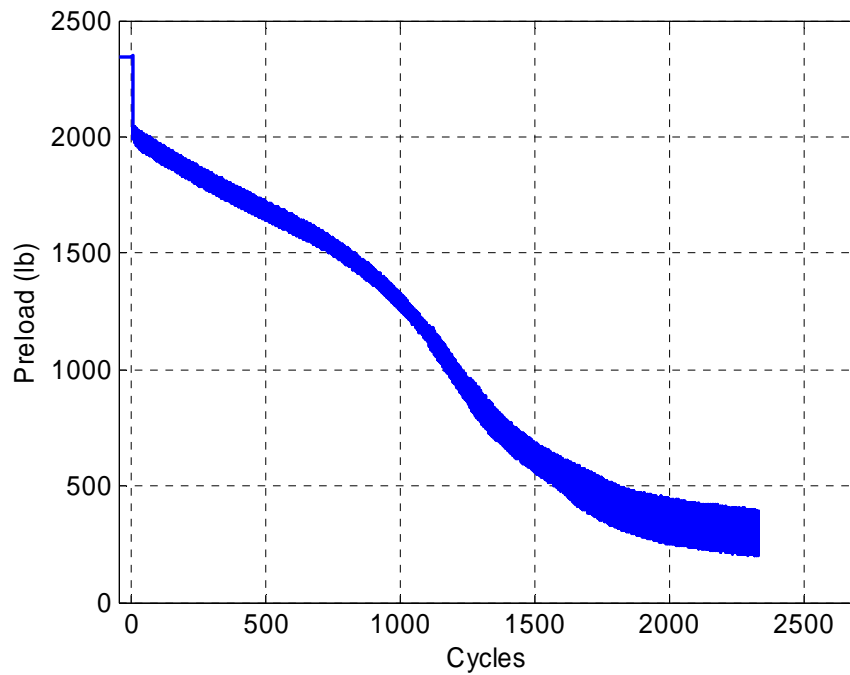


Figure 2.26 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 25.

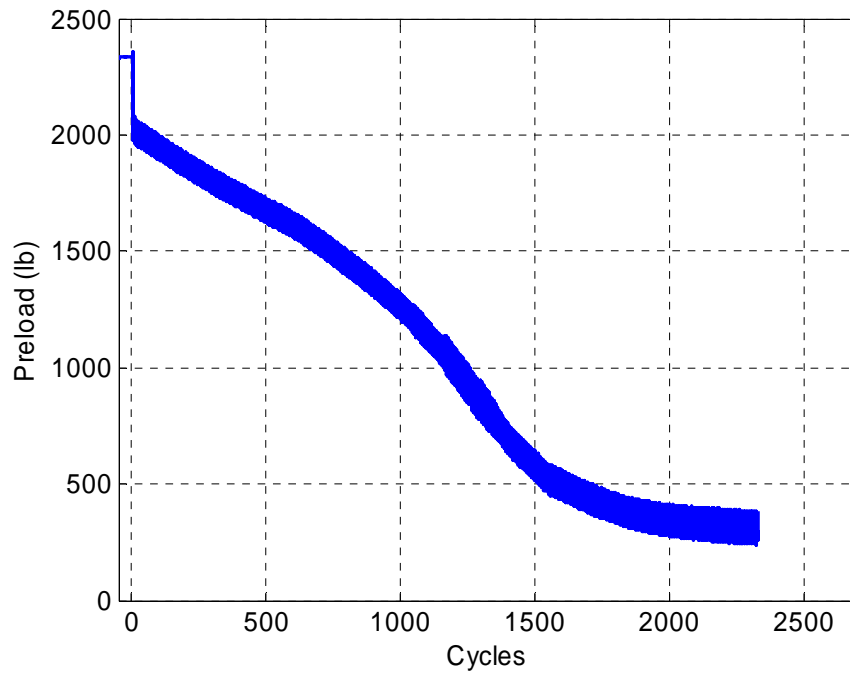


Figure 2.27 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 26.

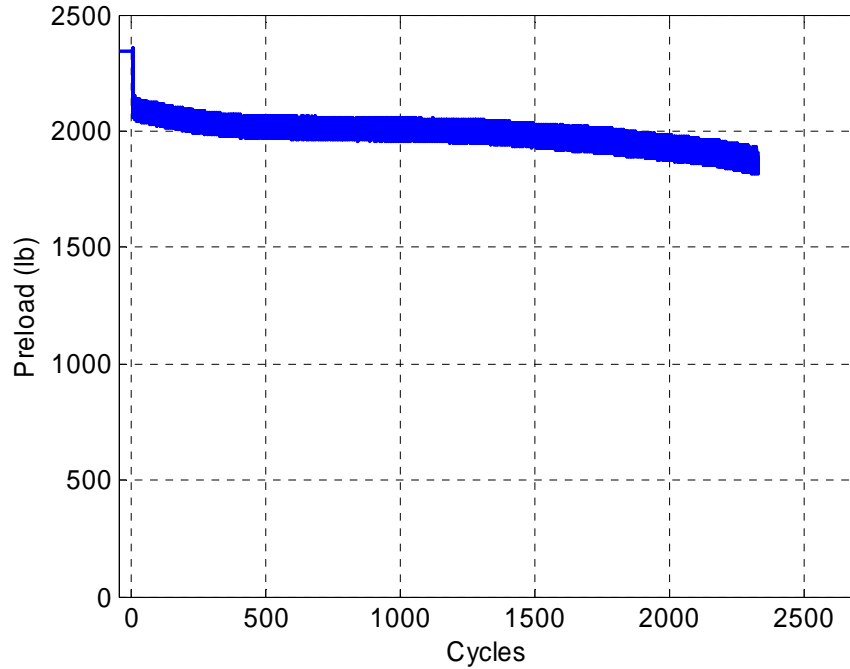


Figure 2.28 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 27.

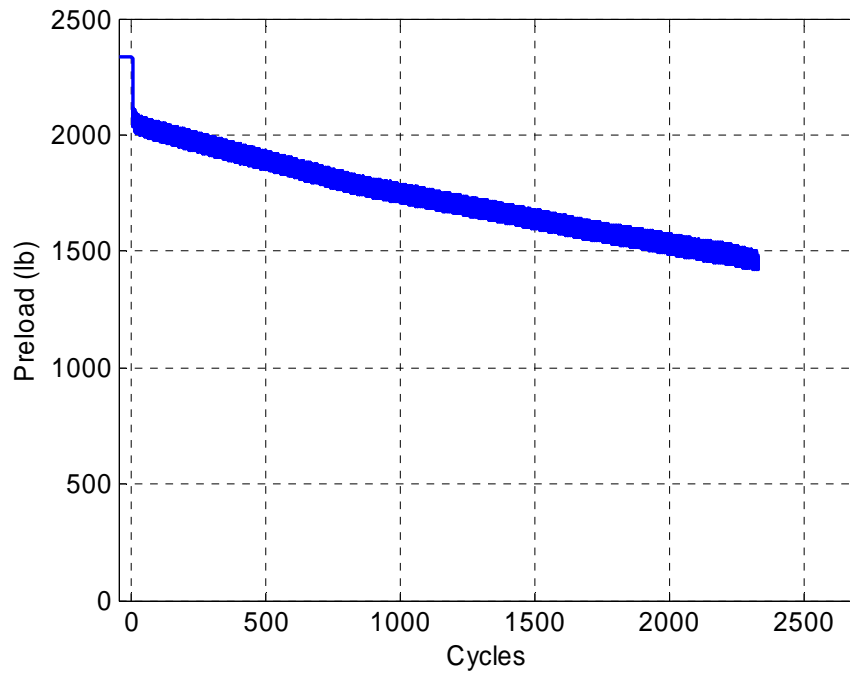


Figure 2.29 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 28.

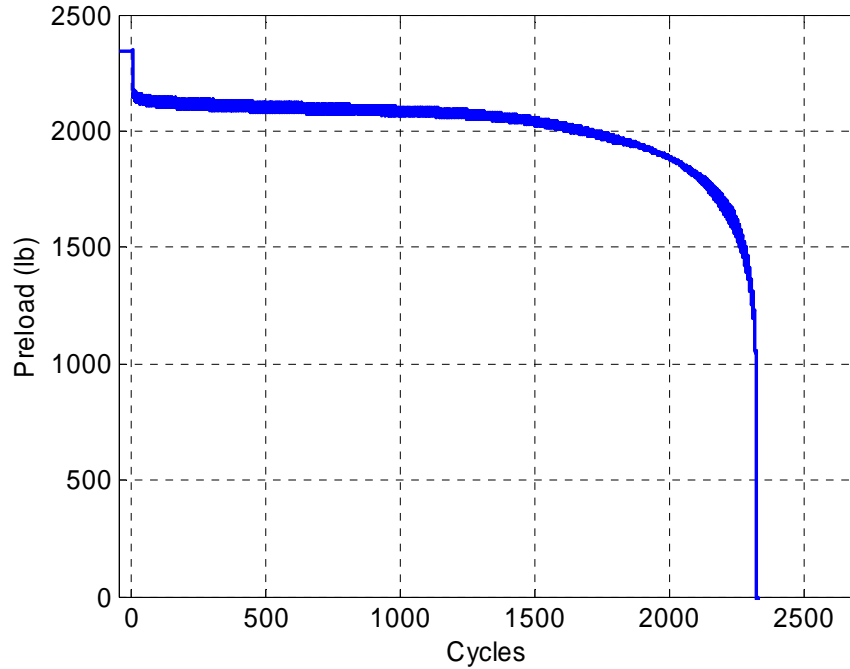


Figure 2.30 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 29.

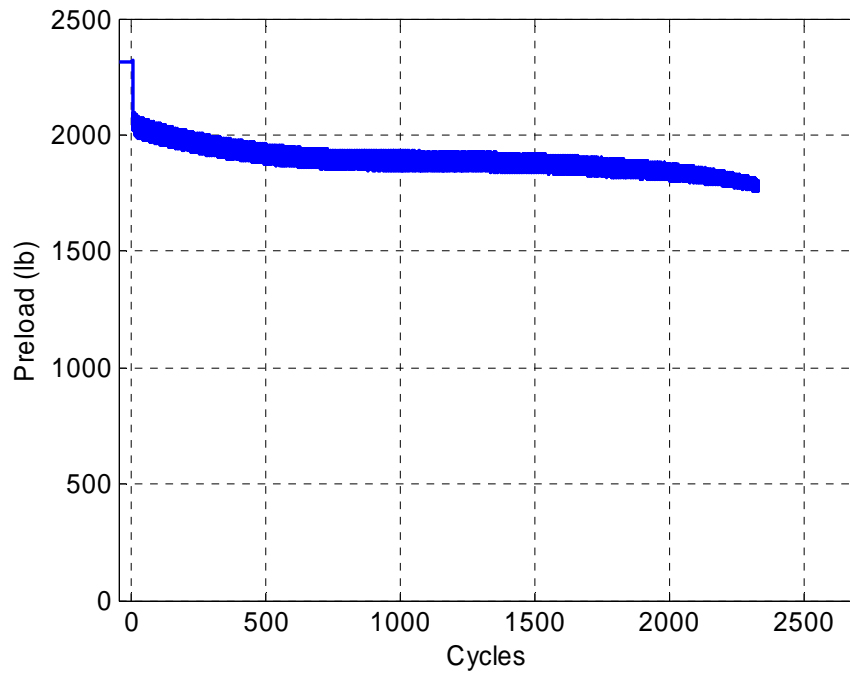


Figure 2.31 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 30.

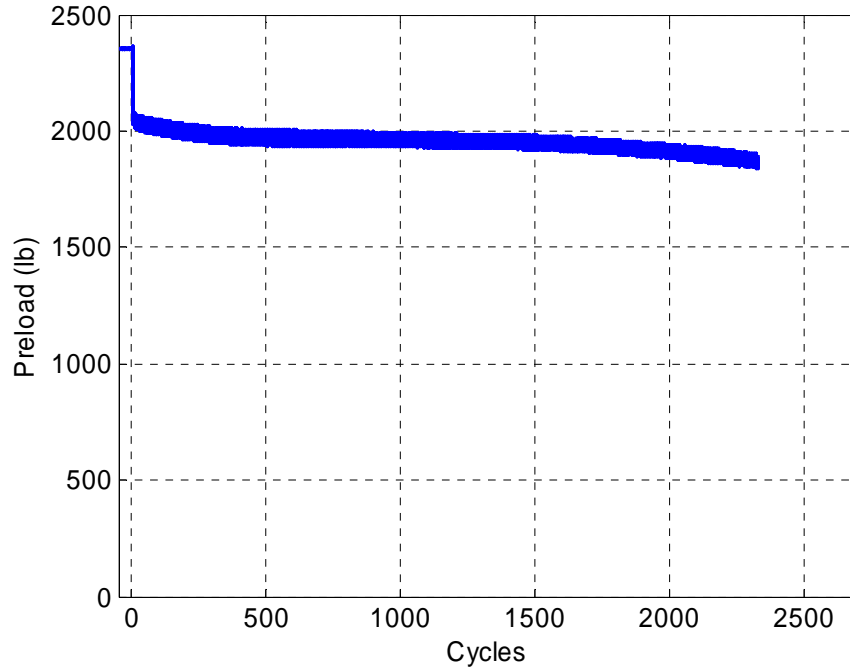


Figure 2.32 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 31.

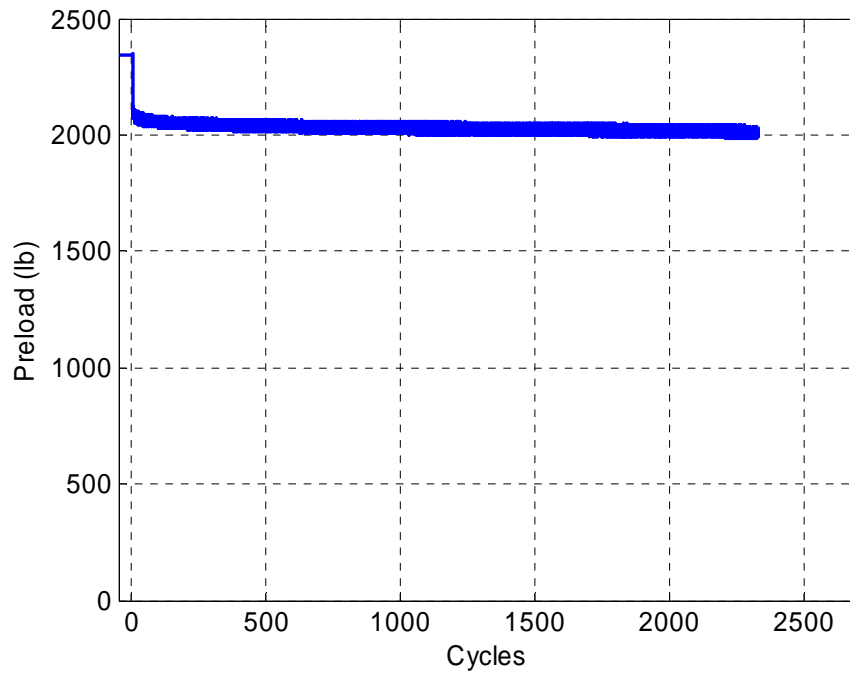


Figure 2.33 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 32.

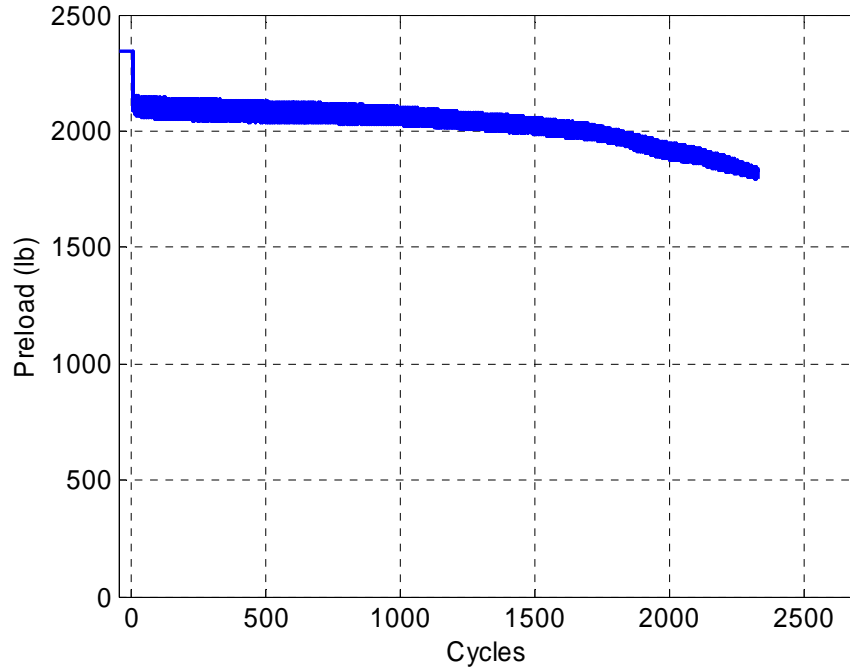


Figure 2.34 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 33.

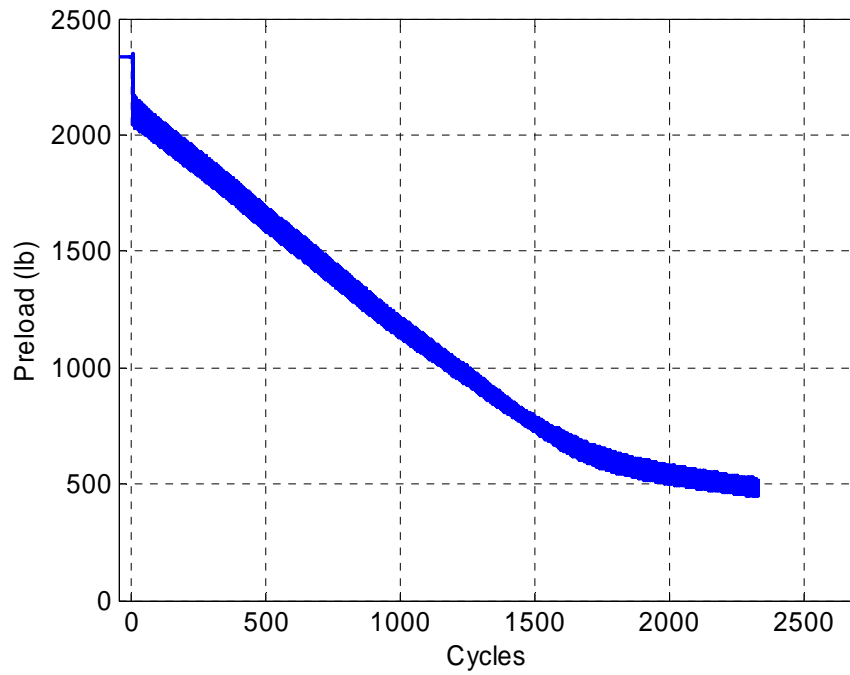


Figure 2.35 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 34.

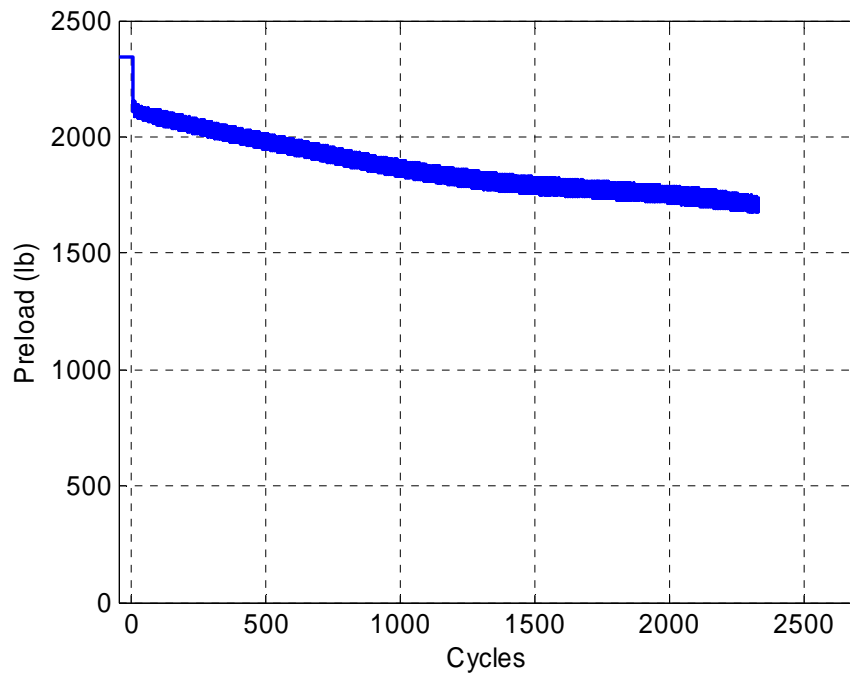


Figure 2.36 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 35.

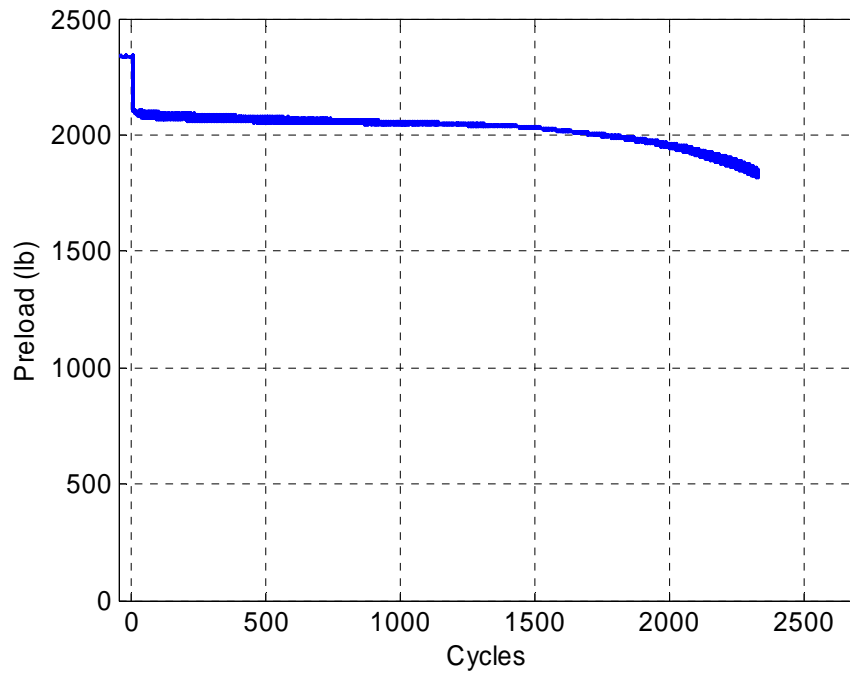


Figure 2.37 Preload versus cycles for “Standard Heli-Coil with Loctite” run number 36.

Chapter 3

Descriptive Analysis

3.1 Introduction

Ordinarily, test data are studied on an individual basis. The minimum preload is found at a predetermined number of cycles for each of the test runs and a statistical analysis is conducted on these minimum preloads. The goal of this thesis is to provide a way to quantify the complete preload versus cycles results of these tests both descriptively and statistically so predictions of how the population is loosening for a given configuration can be created. This chapter describes how methods of condensing the test data were created in order to provide useful quantitative results.

3.2 Minimum preload extraction

Comparing the minimum preloads at a predetermined number of cycles statistically is very useful, but it does not fully explain what is occurring during the full run of these tests. A plot that would show all samples from the same locking level would illustrate the level of repeatability and variability of the loosening in each configuration. Plotting all of the test runs for the same locking level on one plot seems like an obvious solution. However in Figures 3.1-3.3 it can be seen that plots of this kind present results with limited information. The test data overlaps one another with the variation of preload during a cycle, which makes it hard to observe minimum preloads and loosening trends between runs especially with little differences between tests. Since we are generally

interested in the minimum preloads, it would be useful to extract the minimum preloads from each run and plot them against cycles for each run of each locking level. Figures 3.4-3.6 show these plots. M-files for MatLab have been written to create these composite plots, and can be found in Appendix A. These composite plots present the minimum preload versus cycles for the test data for all three locking levels.

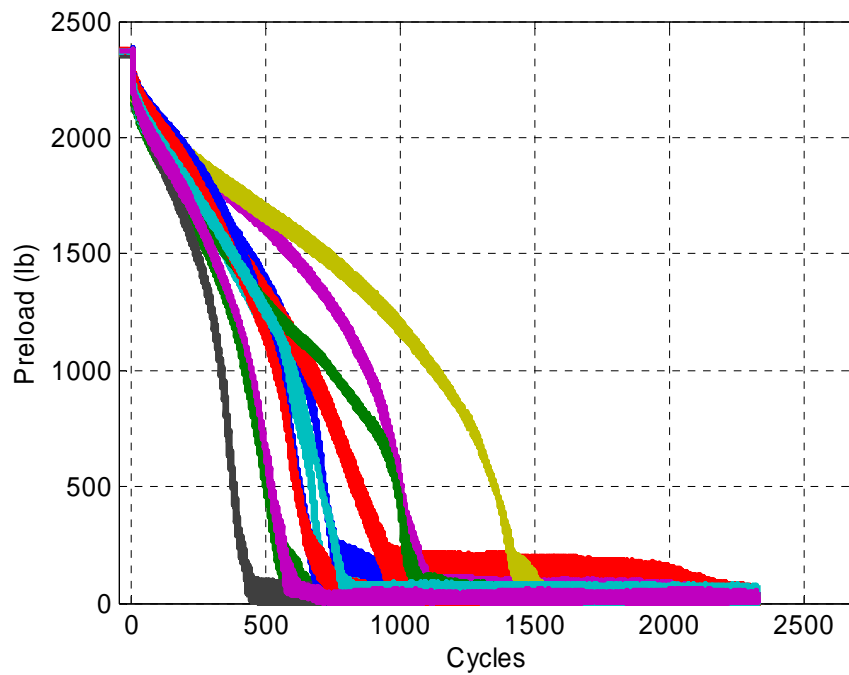


Figure 3.1 Plot of all data for “Standard Heli-Coil with Braycote” runs.

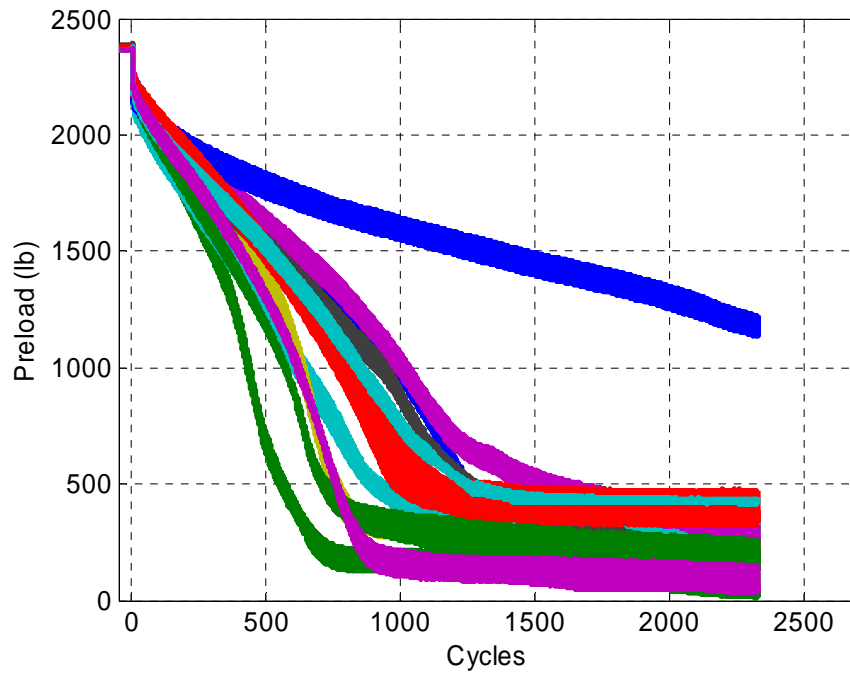


Figure 3.2 Plot of all data for “Locking Heli-Coil with Braycote” runs.

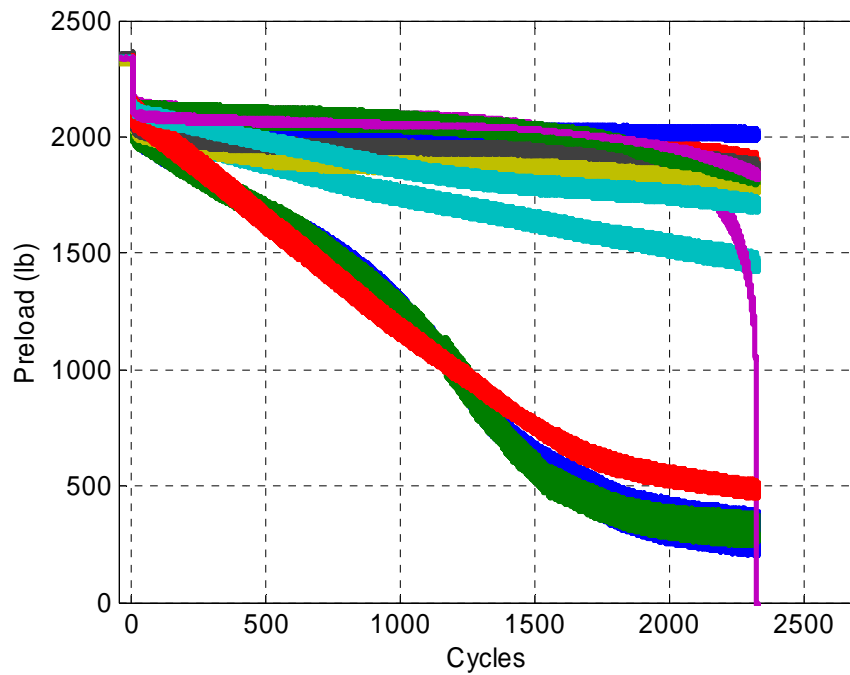


Figure 3.3 Plot of all data for “Standard Heli-Coil with Loctite” runs.

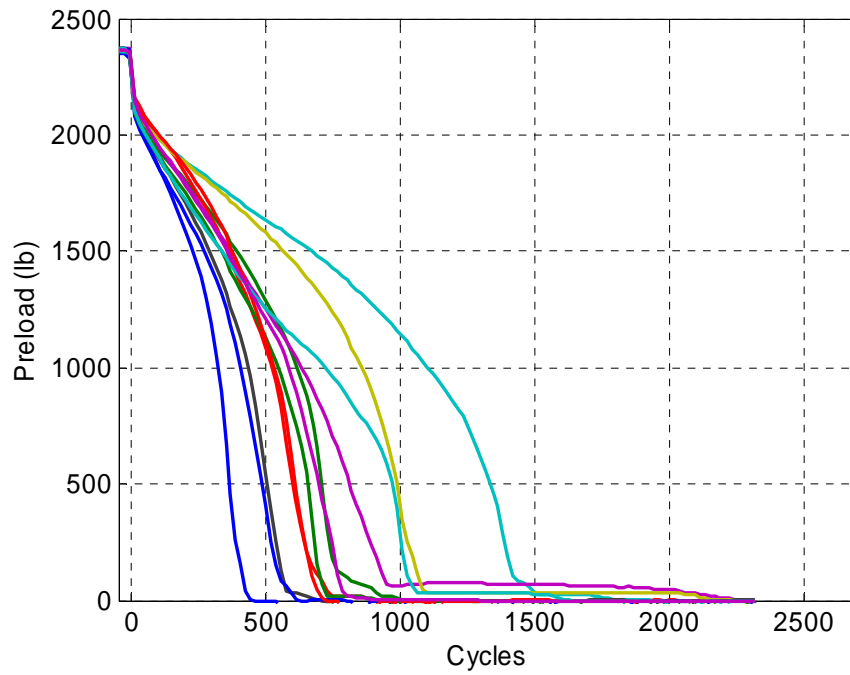


Figure 3.4 Composite plot for “Standard Heli-Coil with Braycote” runs.

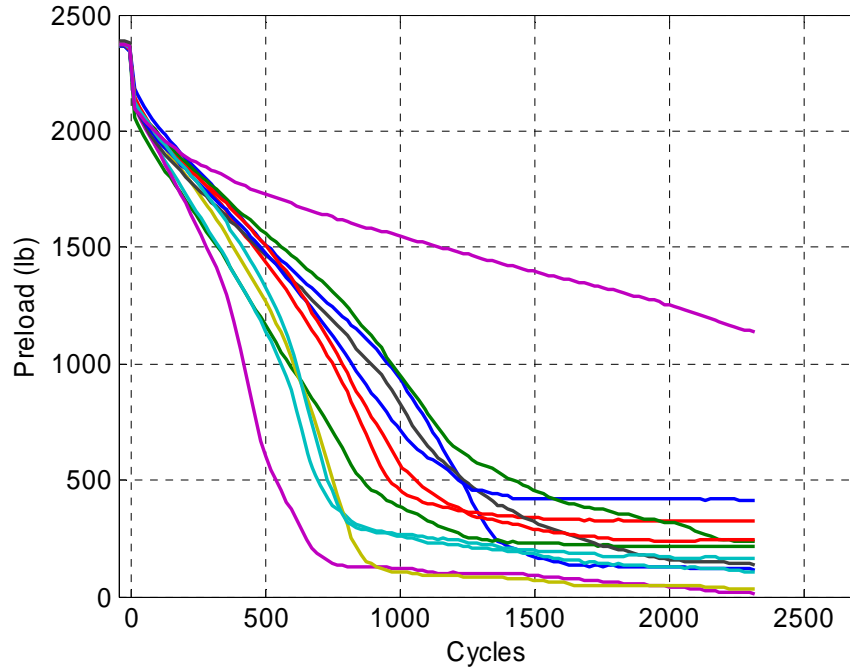


Figure 3.5 Composite plot for “Locking Heli-Coil with Braycote” runs.

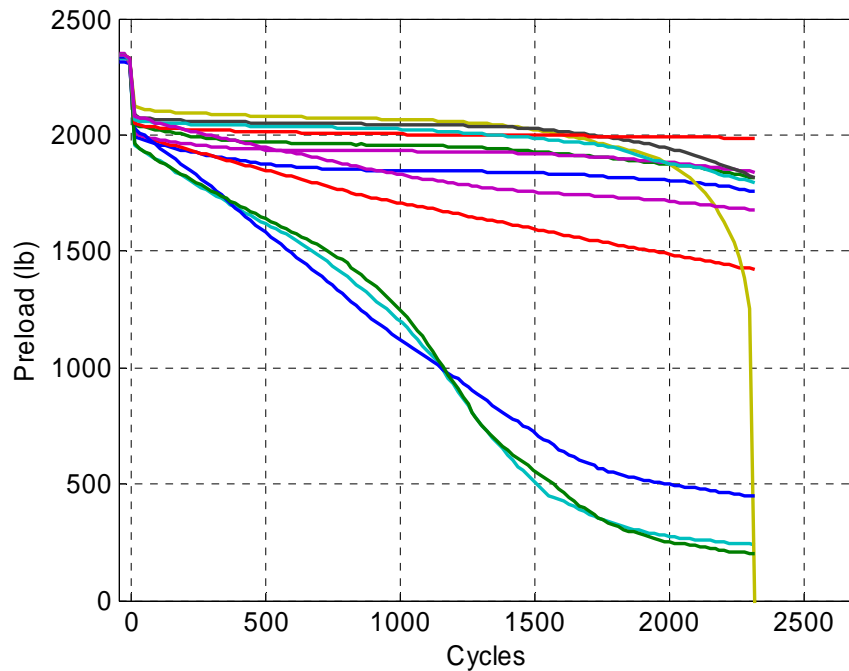


Figure 3.6 Composite plot for “Standard Heli-Coil with Loctite” runs.

The m-file *minstdlines2.m* creates the composite plot for the “Standard Heli-Coils with Braycote” test runs in Figure 3.4, m-file *minlockinglines2.m* creates the composite plot for the “Locking Heli-Coils with Braycote” test runs in Figure 3.5, and m-file *minloctitelines2.m* creates the composite plot for the “Standard Heli-Coil with Loctite” test runs in Figure 3.6. The obtained test data was recorded into data files which can be read by MatLab. These files recorded preload and time at a sample rate of 51.2 samples/second for 160 seconds creating 8,192 data points for preload and time. In the m-files the data files are loaded as arrays. Each m-file loads the data obtained from runs of the same locking level. The minimum preloads for every 18.75 cycles are extracted using “for” loops. Sixty-four data points are grouped into an array 128 times for a total of 8,192 data points for both preload and cycles. The function “min” is used to sort the preload arrays from least to greatest. The first element in the array, which is the minimum preload

for the sixty-four data points, is then put into another array $C_i(j)$ for a particular test where $i=1-12$ and $j=1-128$. The array $C_i(j)$ contains the 128 minimum preloads for every 64 recorded preloads; which is also the minimum preload for every 18.75 cycles. The cycle arrays do not need to be sorted and the median of each array is assigned to the array $D_i(j)$ for a particular test. That is to say every 18.75 cycles makes up the arrays $D_i(j)$. Then all of the preload arrays $C_i(j)$ for a locking level are plotted against the cycle's arrays $D_i(j)$.

Even though the minimum preloads are being plotted against the median of cycles per sixty-four data points, the composite plots show either a very accurate representation or a slightly conservative representation. Figures 3.7-3.12 present comparisons of the actual data and the minimum preload extracted data. Figure 3.7 and Figure 3.8 represent the most conservative minimum preload extraction, Figure 3.9 and Figure 3.10 represent the typical minimum preload extractions, and Figure 3.11 and Figure 3.12 represent one of the best minimum preload extractions. When rates of preload per-cycles loss are larger, the minimum preload extraction plots are less accurate as seen in Figure 3.7 and Figure 3.8. This may be adjusted in the m-file by changing $dpcm$ to be equal to dpc . This would change the median of cycles to equal the end point of every sixty-four cycles. However, in this study we found using the median of cycles acceptable as the conservative representations were only slightly conservative.

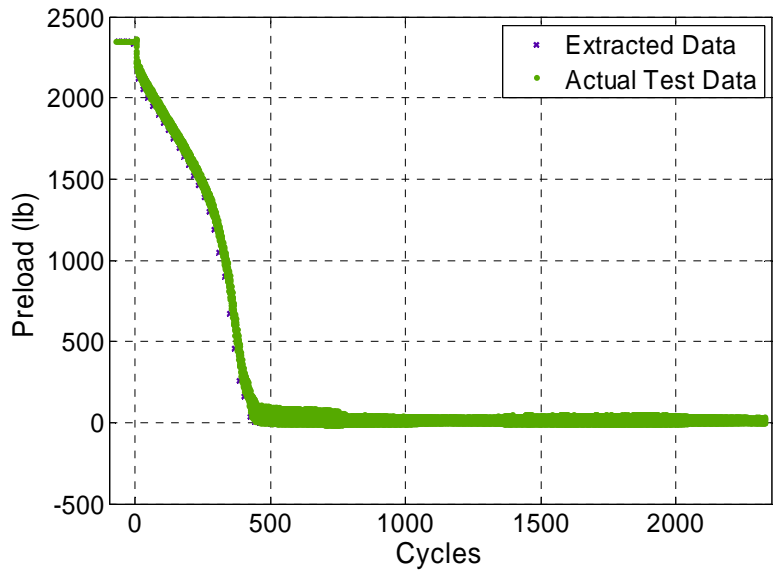


Figure 3.7 Comparison plot of the actual test data and extracted data for run number 7.

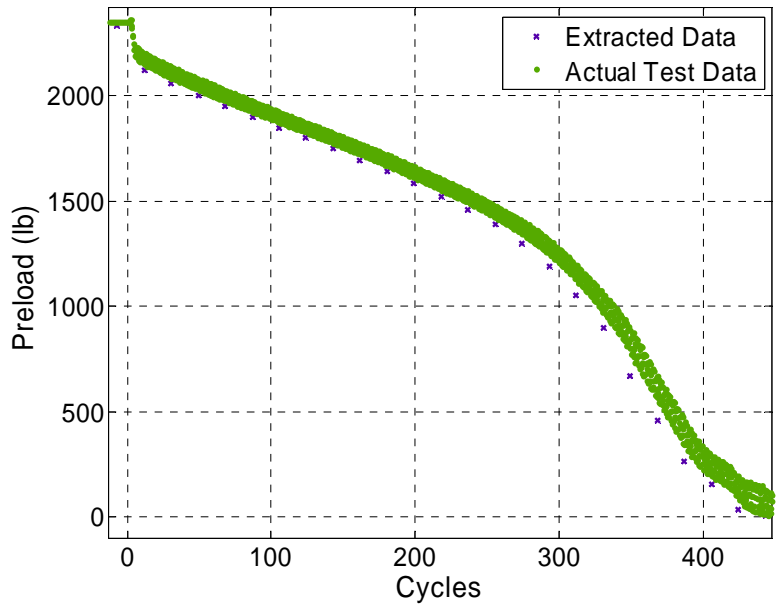


Figure 3.8 Comparison plot of the actual test data and extracted data for run number 7 zoomed in.

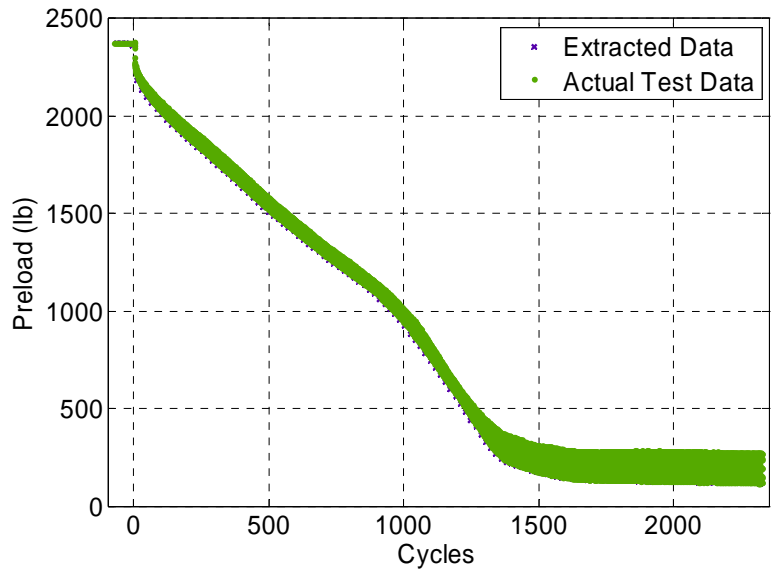


Figure 3.9 Comparison plot of the actual test data and extracted data for run number 13

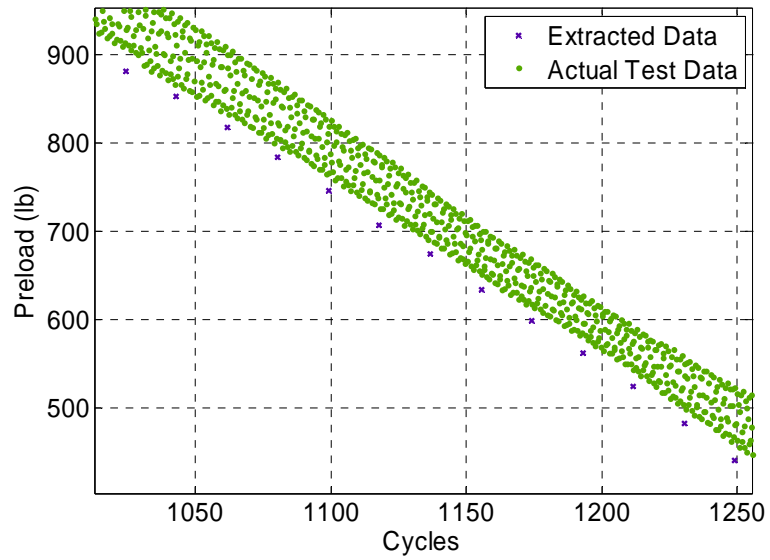


Figure 3.10 Comparison plot of the actual test data and extracted data for run number 13 zoomed in.

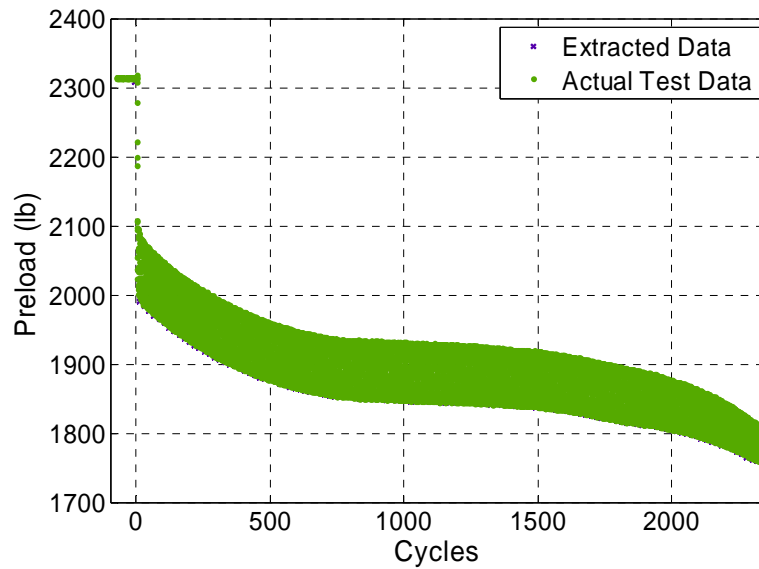


Figure 3.11 Comparison plot of the actual test data and extracted data for run number 30.

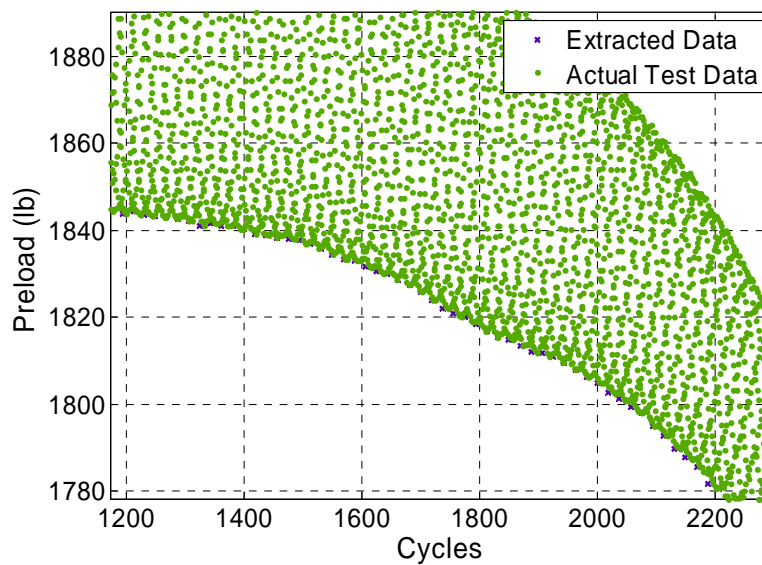


Figure 3.12 Comparison plot of the actual test data and extracted data for run number 30 zoomed in.

Compared to viewing the test data one sample at a time or all composite plots with preload cycle variation, the composite plots of minimum preloads versus cycles provide many advantages. It allows the all samples of a test to be viewed in one plot to

see if the loosening is similar for each related sample. In the case of the obtained test data, these composite plots reduced thirty-six plots down to three. Comparison of the locking performance is then easily evaluated by comparing the three composite plots. Patterns of loosening emerge out of the composite plot, and it sheds insight on the loosening process as a whole. These composite plot m-files also contribute to a technique that will allow the predictions of the populations to be created.

3.3 Sample statistics

In addition to the composite plots, presenting sample statistics in a plot that would summarize test samples by preload versus cycles would help to describe what is happening with the samples as a whole. This will compliment the composite plots and provide a tool to summarize sample test results. The sample statistics that will be evaluated are the sample mean, sample median, sample quartiles, and sample extents. Individual plots of these sample statistics were made in MatLab and are shown in Figures 3.13-3.24.

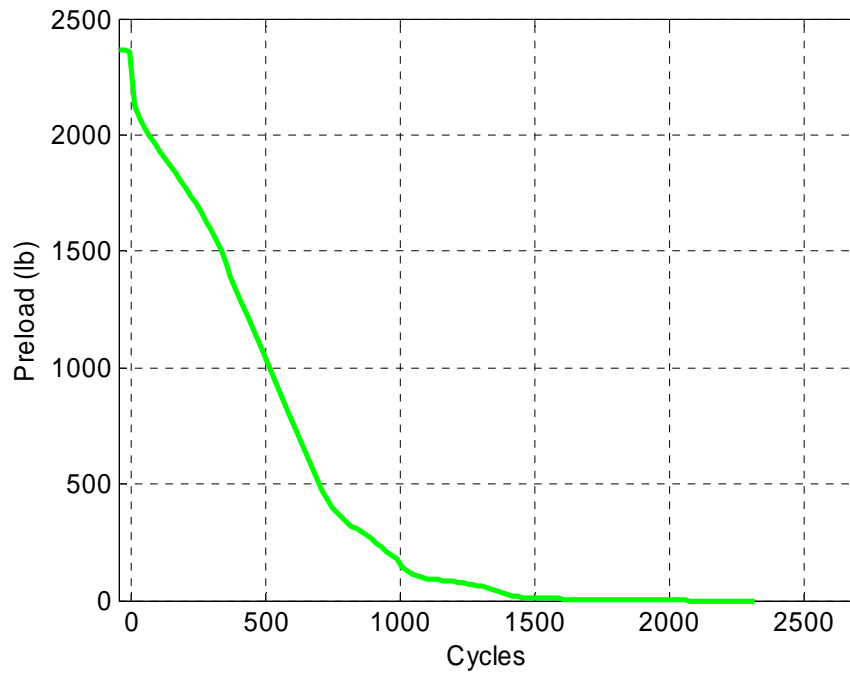


Figure 3.13 Sample mean for “Standard Heli-Coil with Braycote” runs.

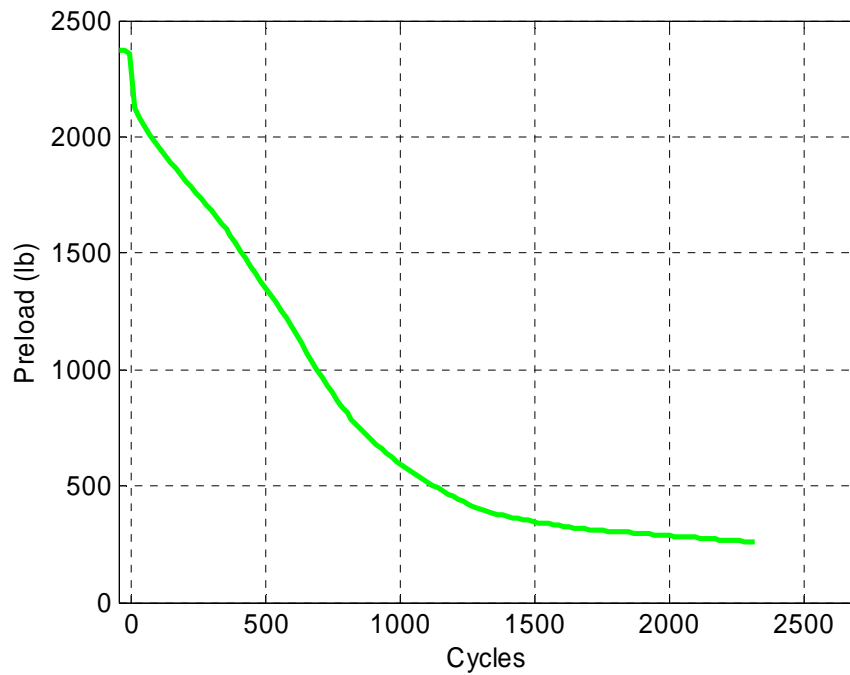


Figure 3.14 Sample mean for “Locking Heli-Coil with Braycote” runs.

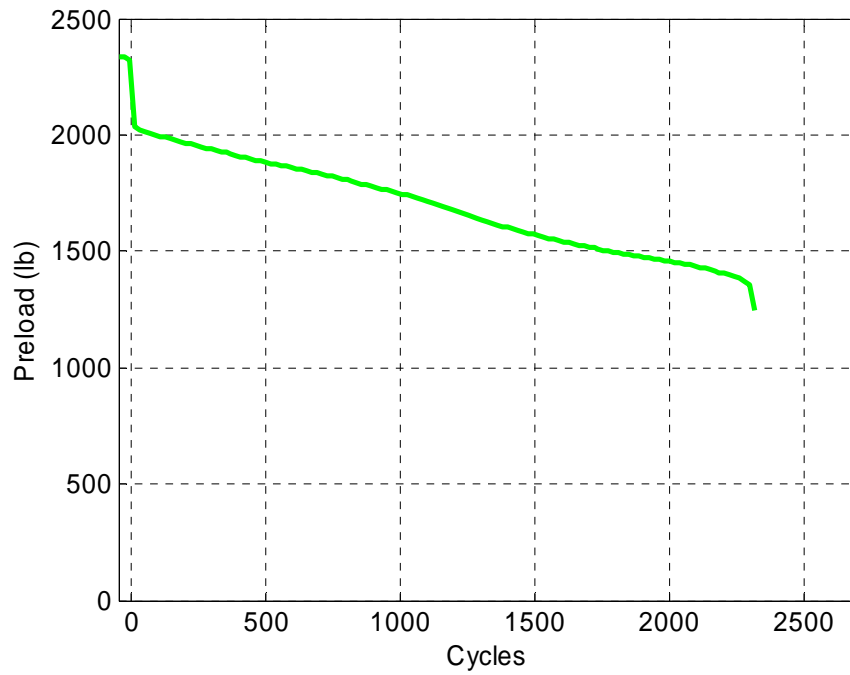


Figure 3.15 Sample mean for “Standard Heli-Coil with Loctite” runs.

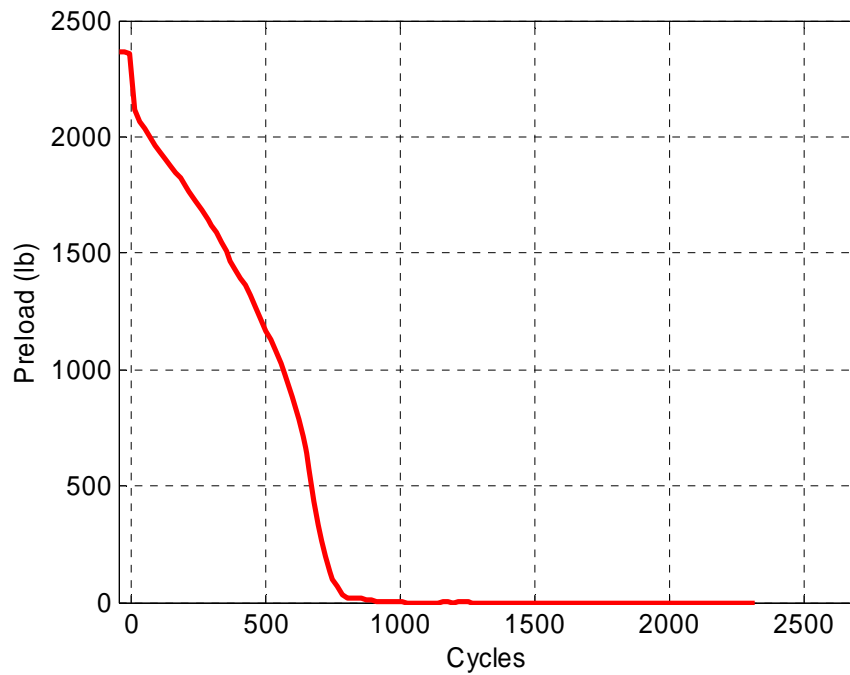


Figure 3.16 Sample median for “Standard Heli-Coil with Braycote” runs.

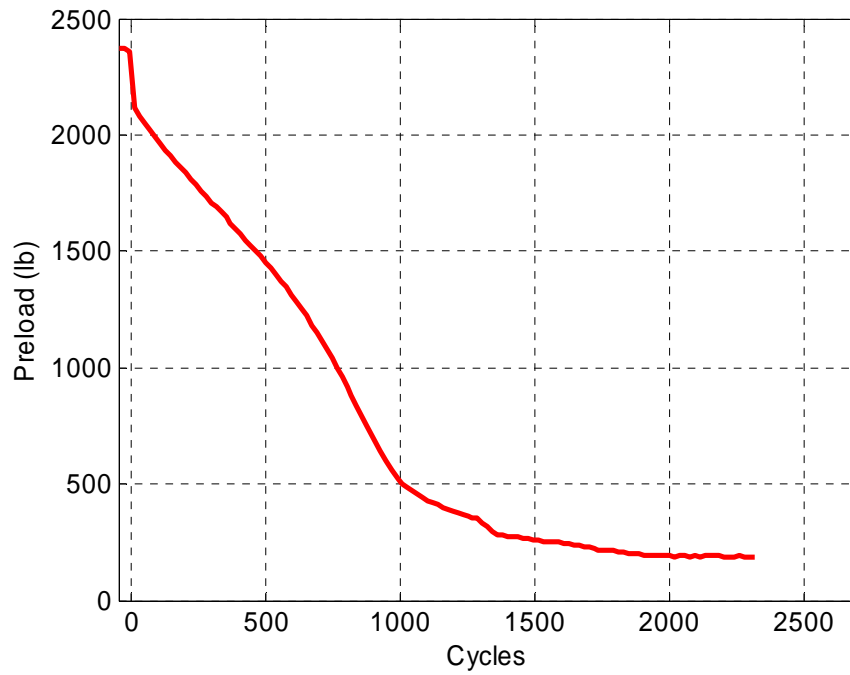


Figure 3.17 Sample median for “Locking Heli-Coil with Braycote” runs.

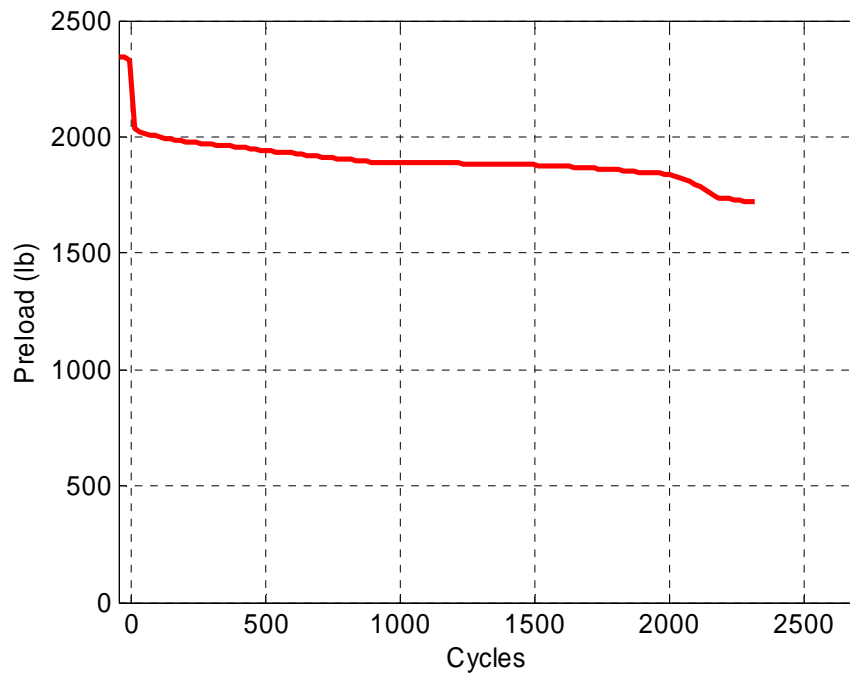


Figure 3.18 Sample median for “Standard Heli-Coil with Loctite” runs.

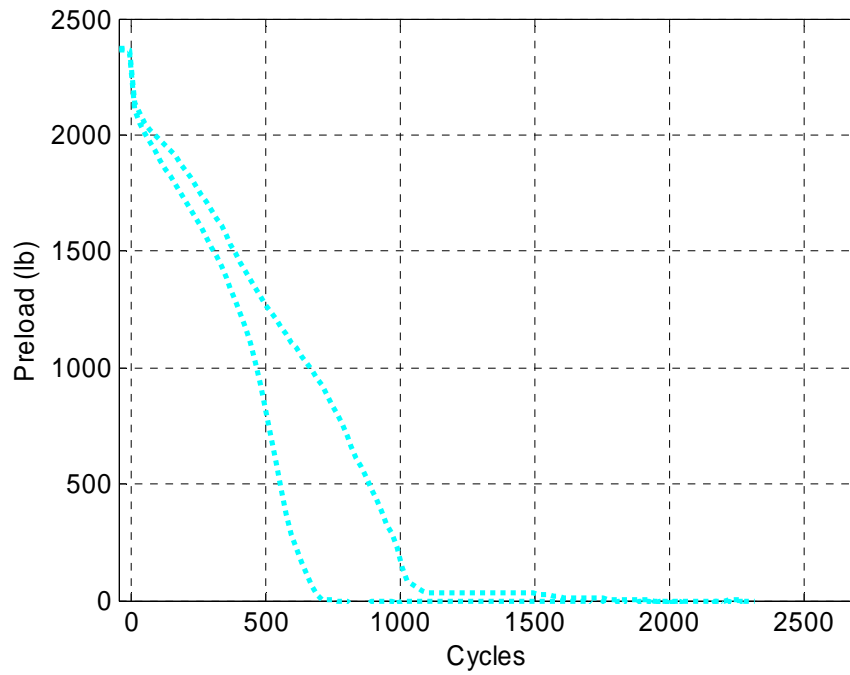


Figure 3.19 Sample upper and lower quartiles for “Standard Heli-Coil with Braycote” runs.

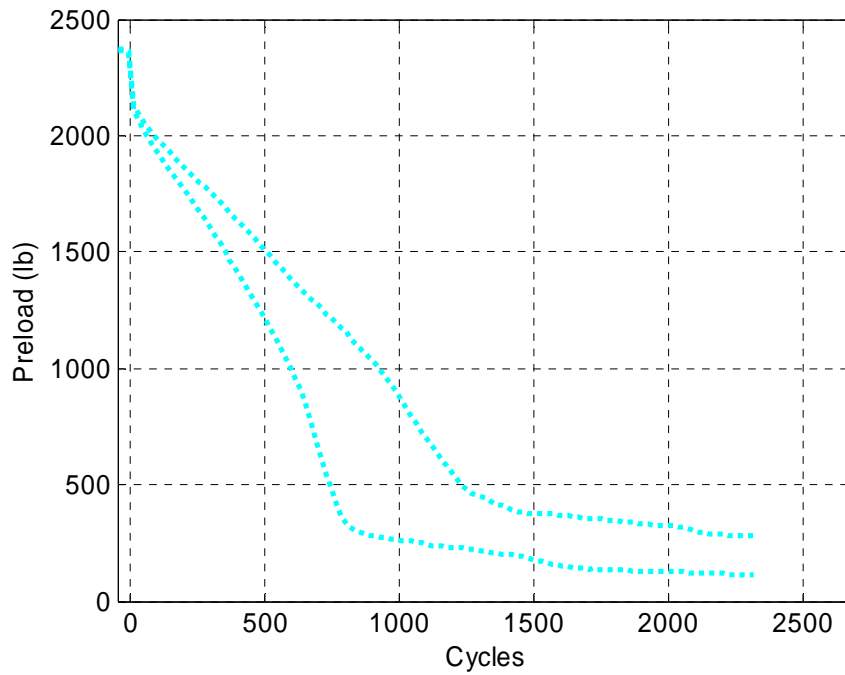


Figure 3.20 Sample upper and lower quartiles for “Locking Heli-Coil with Braycote” runs.

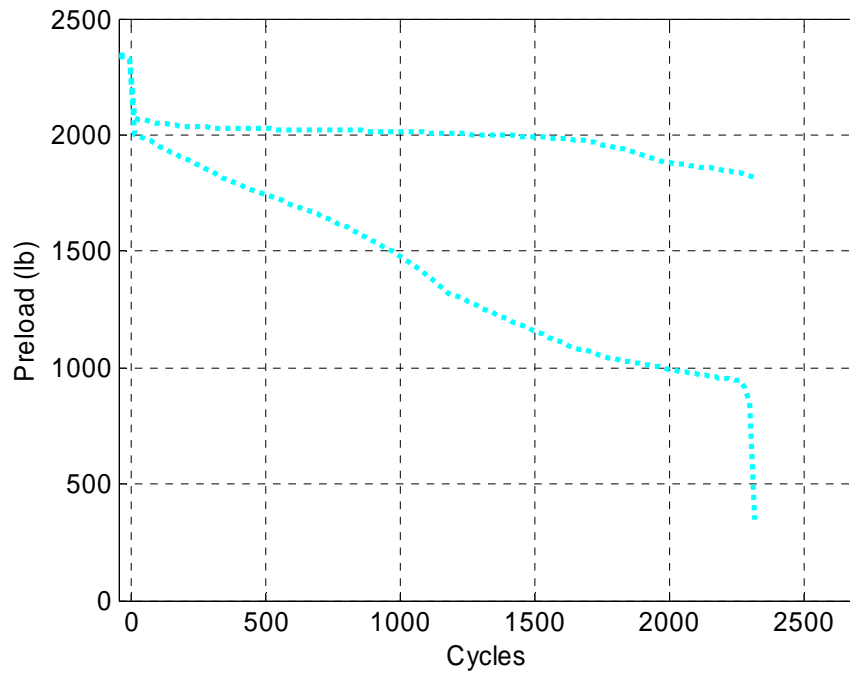


Figure 3.21 Sample upper and lower quartiles for “Standard Heli-Coil with Loctite” runs.

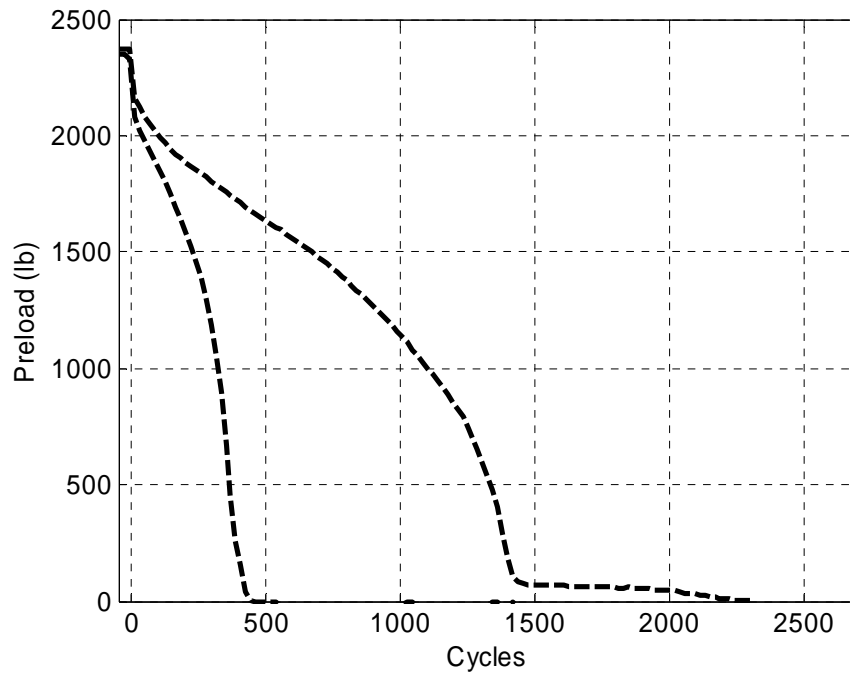


Figure 3.22 Upper and lower extents for “Standard Heli-Coil with Braycote” runs.

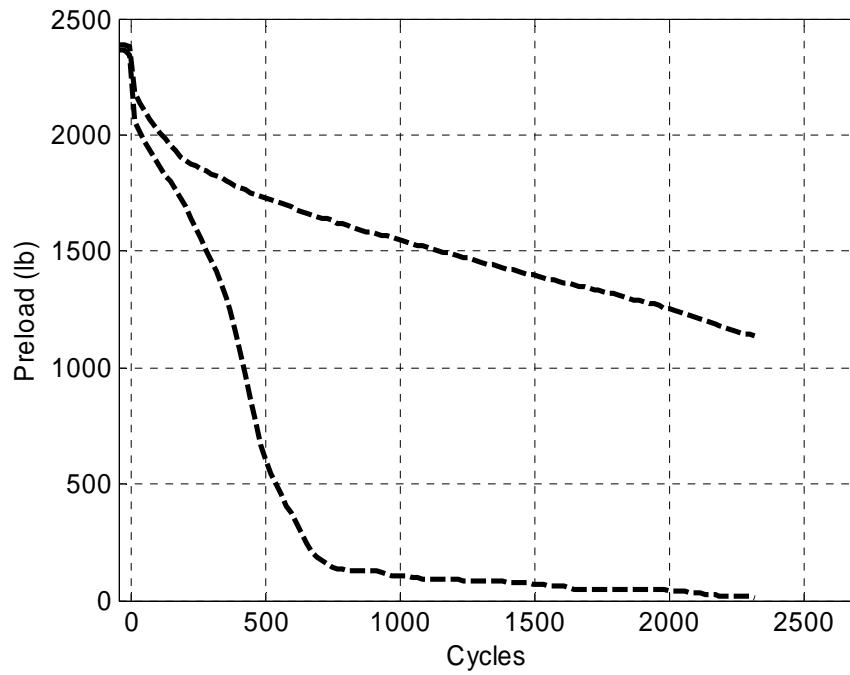


Figure 3.23 Upper and lower extents for “Locking Heli-Coil with Braycote” runs.

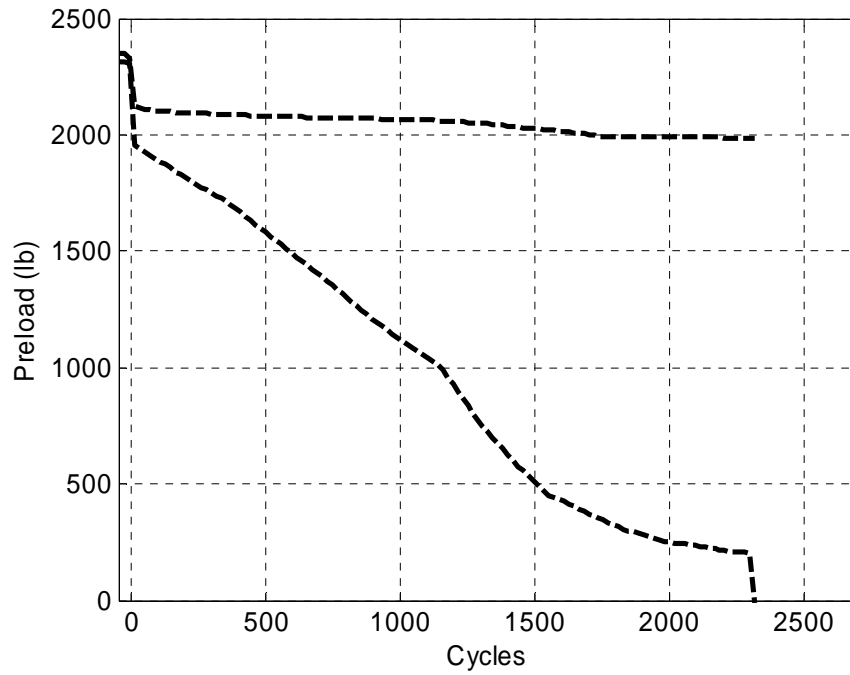


Figure 3.24 Upper and lower extents for “Standard Heli-Coil with Loctite” runs.

Though these sample statistics plots are very descriptive, it is still difficult to evaluate a sample as a whole when trying to compare four figures per sample. To create a tool that would allow all sample statistics to be presented in one plot, another type of m-file has been created and can be viewed in Appendix B. These m-files create preload versus cycle plots of sample means, sample medians, sample quartiles, and upper and lower extents of the tests for each locking level in one plot. Figures 3.25 -3.27 shows these descriptive statistic plots.

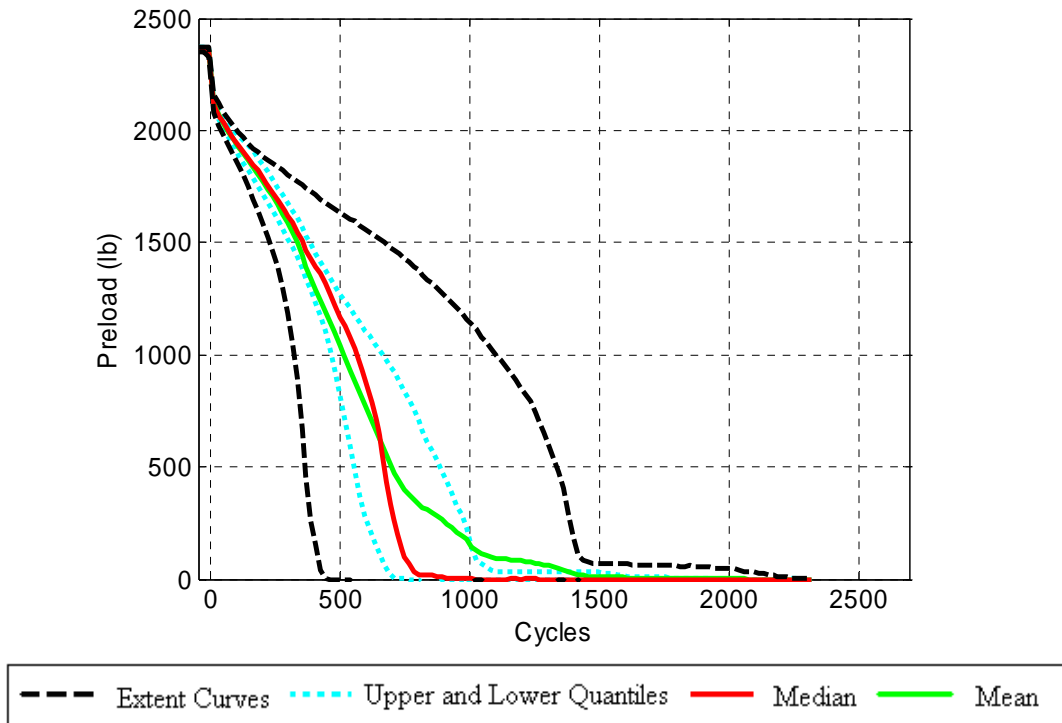


Figure 3.25 Mean, Median, upper and lower quartile, and extent curves for “Standard Heli-Coil with Braycote” runs.

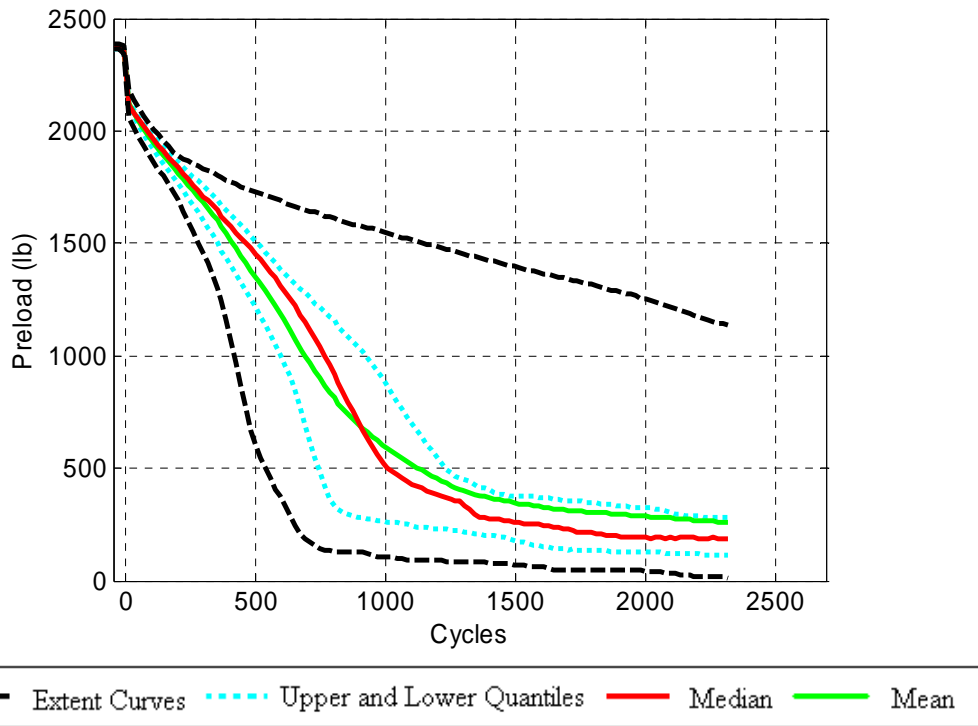


Figure 3.26 Mean, Median, upper and lower quartile, and extent curves for “Locking Heli-Coil with Braycote” runs.

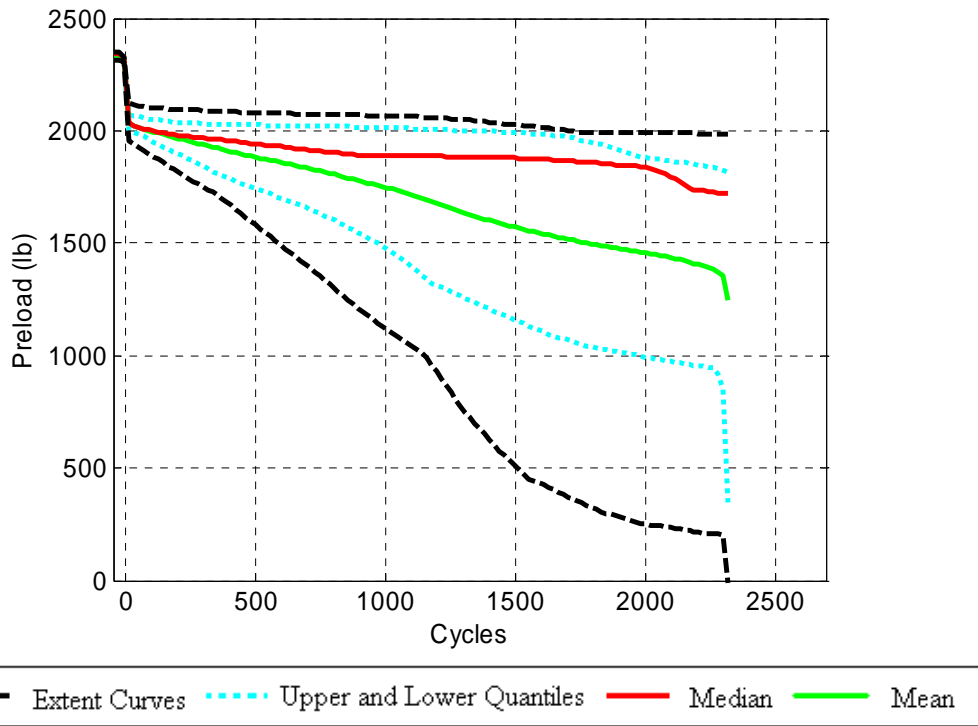


Figure 3.27 Mean, Median, upper and lower quartile, and extent curves for “Standard Heli-Coil with Loctite” runs.

M-files *quanstd2.m*, *quanlocking2.m*, and *quanloctite2.m* create the descriptive statistical plots of minimum preloads versus cycles for the “Standard Heli-Coil with Braycote”, “Locking Heli-Coil with Braycote”, and “Standard Heli-Coil with Loctite” tests data respectively. In the m-files, the minimum preloads and median of cycles are found by the same method used in m-files for the composite plots. That method creates arrays that contain 128 minimum preloads from every 18.75 cycles. Then minimum preloads for all runs are put into matrix td in a manner in which each row is the minimum preload for all runs at a particular cycle. A “for” loop then selects each row of the matrix td and extracts the upper and lower extents, medians, 25% quartile, and 75% quartiles from each row and puts that extracted data into arrays. The medians and quartiles are found using the MatLab function “quantile” and the extents are found using the MatLab function “min” and “max”. The means of the samples are calculated by adding each minimum preload array together and then dividing them by the number of samples. This creates an array containing 128 means of all the minimum preloads. Then the arrays of means, medians, quartiles, and extents are plotted against a cycle array $D_i(j)$.

These m-files provide a tool to aid evaluation of the sample data in a clear and concise manner. At a glance these plots present where 100 percent of the data lies (between upper and lower extents), where 50 percent of the data lies (between the upper and lower quartile), the sample mean, and the sample median. Sample test results then can be easily evaluated.

Chapter 4

Tests for Normality

4.1 Introduction

This work is focused on developing population predictions that quantify the loosening of preloaded bolted joints. In particular this work develops population predictions for three different locking levels of threaded inserts, but could be used for most tests from a transverse vibration machine. In a perfect scenario the results would have shown a single curve for preload versus cycles (i.e., no variation from test to test) for each of the three different locking levels. However, like most experiments, the test data obtained for this work showed variation within individual locking levels. To describe the statistical model of loosening, a graphical model to describe the population is desired. To do this the distribution that the samples belongs to must be determined, and tests for normality will be conducted. Using the minimum preloads at approximately 10, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250 cycles, ten sets of sample size twelve are created for the “Locking Heli-Coil with Braycote” and “Standard Heli-Coil with Loctite” locking levels. The “Standard Heli-Coil with Braycote” locking level only has six sets of sample size twelve using the minimum preloads at 10, 250, 500, 750, 1000, 1250 cycles. This is because only twenty-five percent of the “Standard Heli-Coil with Braycote” minimum preload data has a value greater than zero after 1250 cycles. These sample subsets will be tested for normality, to determine whether or not the samples belong to the Normal Distribution.

4.2 Probability plots

To determine whether the data collected from the three different locking levels are normally distributed, normal probability plots have been developed. If the sample sets are normally distributed then population predictions that describe the loosening of the three different locking levels can be developed using equations for normally distributed sample sets. The normal probability plot is a graphical technique that allows for a quick visual test to determine whether or not the sample data is from the Normal Distribution.

Representative normal probability plots for minimum preloads at 250 cycles for each of the three locking levels are shown in Figures 4.1-4.3. Figures 4.1-4.3 show a good linear relationship on the minimum preloads at 250 cycles. The normal probability plots for all twenty-six sample sets are provided in Appendix C.

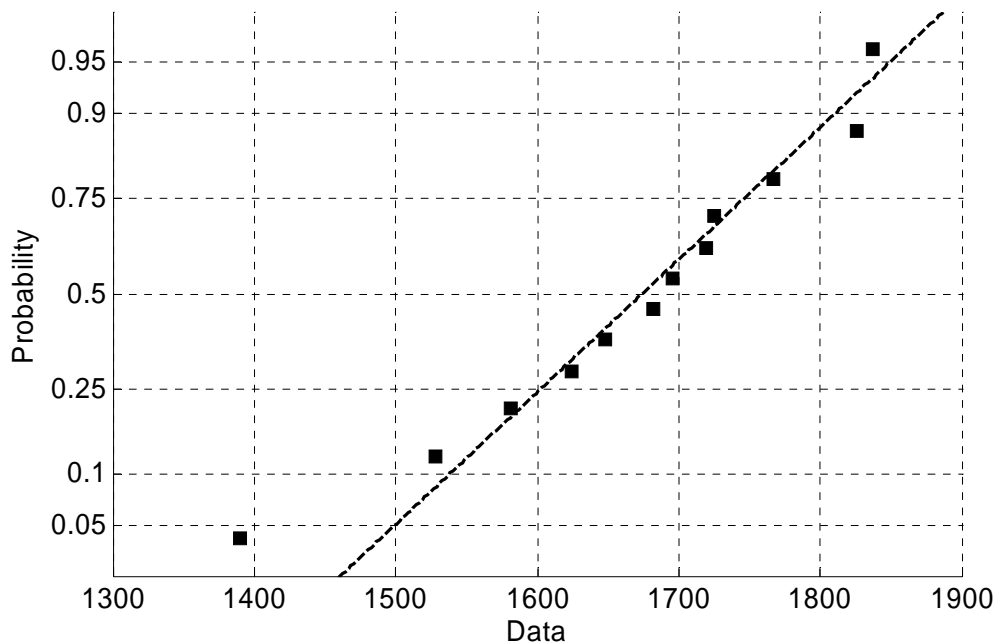


Figure 4.1 Normal plot for “Standard Heli-Coil with Braycote” minimum preloads at 250 cycles.

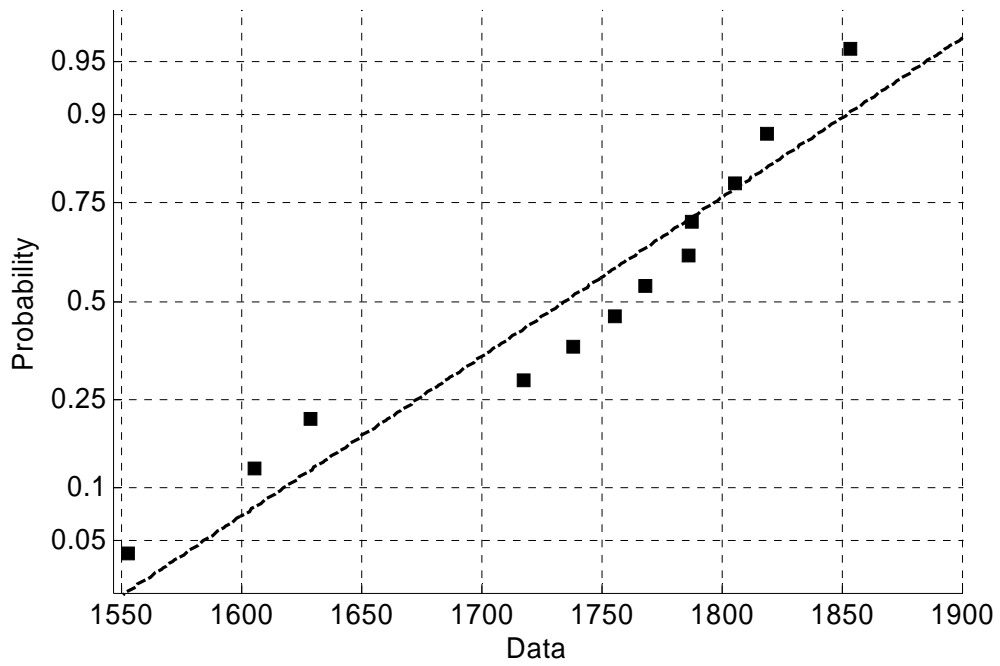


Figure 4.2 Normal plot of “Locking Heli-Coil with Braycote” minimum preloads at 250 cycles.

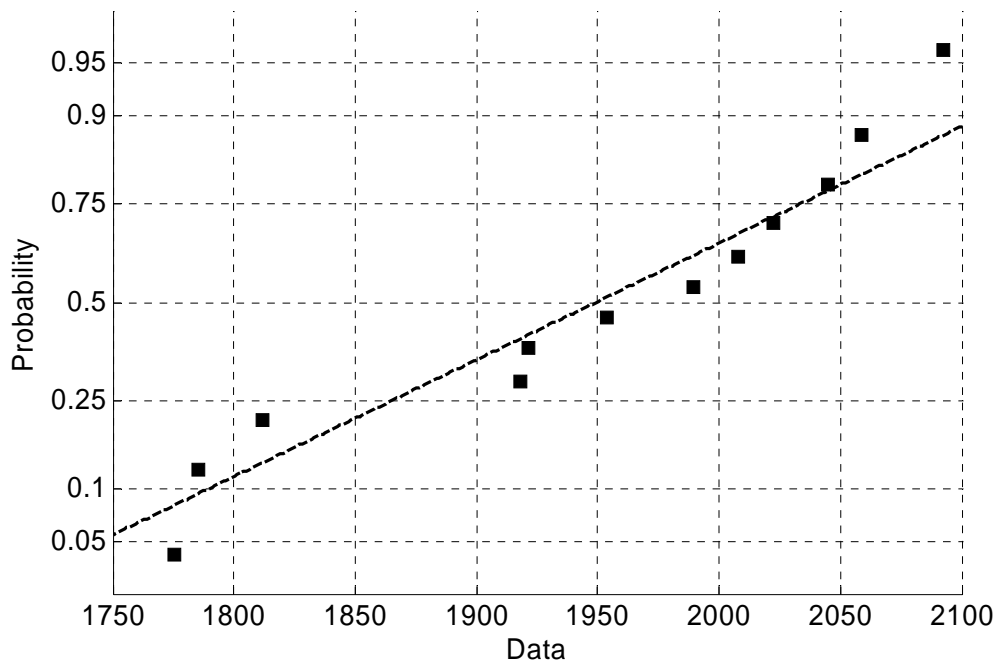


Figure 4.3 Normal plot of “Standard Heli-Coil with Loctite” minimum preloads at 250 cycles.

Normal probability plots were created in MatLab using the command “probplot”. The “probplot” function in MatLab uses one of the most commonly used plotting positions to create Normal Plots. The plotting position is defined by the following equation:

$$p_i = \frac{(i - 0.5)}{n} \quad (4.1)$$

where p_i is the plotting position, n is the sample, and $i=1..n$. When these plots are approximately linear it can be said that the samples belong to the Normal Distribution. Most of the plots appear to have a strong linear relationship, but to see how linearly related they are, a further check for normality is pursued.

4.3 Probability plot correlation coefficient test for normality

A probability plot correlation coefficient (PPCC) test for normality was used to complement the results of the probability plots. One of the most useful features of the PPCC is that it produces a single number that quantifies normality. PPCC tests describe how linear the probability plots are. M-files for MatLab were created to calculate PPCC for the same sample sets as the Probability Plots. In Appendix D, m-file *normstd2.m* calculates the correlation coefficients for “Standard Heli-Coil with Braycote” test data, m-file *normlocking2.m* calculates correlation coefficients for the “Locking Heli-Coil with Braycote” test data, and the m-file *normloctite2.m* calculates the correlation coefficient for “Standard Heli-Coil with Loctite” test data. These m-files calculate a total of twenty-six correlation coefficients, one for each of the normal plots. Thus the same minimum

preloads that were used to create the normal plots were also used in these m-files to run PPCC.

To calculate the correlation coefficient the minimum preload data must be arranged in order from smallest to largest, x_1, x_2, \dots, x_{12} , where x_1 is the smallest and x_{12} is the largest for each sample set. To do this, the same method used in the sample statistic m-files is implemented and the minimum preload arrays are put into a matrix td . Rows 5, 18, 31, 44, 58, 71, 84, 98, 111, 124 of matrix td contain minimum preloads at 10, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250 cycles respectively. Which are the minimum preloads used for creating the normal plots. Then using “for” loop these preloads are put into arrays $m_i(j)$, where $i=1..10$ and $j=1..12$. These arrays are sorted by the function “sort” from least to greatest. The means of these arrays are calculated by the function “mean”, which will be needed later to find the correlation coefficient.

Next the uniform order statistic median is calculated from the Blom plotting position equation (4.2).

$$p_i = (i - 0.375)/(n + 0.25) \quad (4.2)$$

Here i is 1, 2...12 and n is twelve for a sample size of twelve. The Blom plotting position equation (4.2) is used because it has been found to provide a more powerful correlation coefficient test than the Shanpiro-Francia, Filliben tests, and tests using the plotting position equation (4.1) [16]. Then, the normal order statistic median was approximated by equation (4.3).

$$b_i = \frac{(2.515517 + 0.82853 * t + 0.010328 * t^2)}{1 + 1.432788 * t + 0.183269 * t^2 + 0.001308 * t^3} - t + e(p) \quad (4.3)$$

where,

$$t = \sqrt{\ln \frac{1}{p_i}} \quad (4.4)$$

$$e(p) < 4.5 * 10^{-4} \quad (4.5)$$

Finally the correlation coefficient R_p is calculated with equation (4.6).

$$R_p = \frac{\sum (x_i - \bar{x}) * (b_i - \bar{b})}{\sqrt{\sum (x_i - \bar{x})^2 * \sum (b_i - \bar{b})^2}} \quad (4.6)$$

The summations of the correlation coefficient are calculated using “for” loops. The actual correlation coefficients are the output of the m-files and are seen in an array. Elements one through ten represent the correlation coefficients in order for minimum preloads at cycles 10, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250.

Results of the Correlation Coefficients for all thirty samples can be seen in Table 4.1. When the correlation coefficient has a higher value than the critical value the sample set can be said to belong to the Normal Distribution. The critical values for the Blom plotting position are found in tables provide by Looney and Gulledge [16]. Table 4.2 gives a few of these critical values for data with different sample sizes and significant levels.

Table 4.1 Correlation coefficients R_p .

Cycles	Standard Heli-Coil w/ Braycote	Locking Heli-Coil w/ Braycote	Standard Heli-Coil w/ Loctite
10	0.992	0.950	0.993
250	0.977	0.959	0.969
500	0.938	0.920	0.944
750	0.881	0.981	0.931
1000	0.704	0.954	0.904
1250	0.603	0.835	0.875
1500	NA	0.781	0.866
1750	NA	0.778	0.855
2000	NA	0.783	0.854
2250	NA	0.799	0.876

Table 4.2 Empirical percentage points for correlation coefficient test based on Blom's plotting position.

Sample Size	Significant Level					
	0	0.005	0.01	0.025	0.05	0.1
10	0.578	0.862	0.879	0.901	0.918	0.934
11	0.560	0.870	0.886	0.907	0.923	0.938
12	0.544	0.876	0.892	0.912	0.928	0.942
13	0.529	0.885	0.899	0.918	0.932	0.945

The sets of samples are normally distributed with a significance level of 0.005 for all the “Standard Heli-Coils with Braycote” except for the minimum preloads at 1000 and 1250 cycles. This is due to the fact that more than half of the test runs have a preload of zero at 1000 and 1250 cycles, as other runs still record a preload above zero.

Samples for the “Standard Heli-Coils with Loctite” are normally distributed at a significance level 0.005 or higher except for minimum preloads sample sets at cycles 1500 to 2000. This is mainly due to the larger difference in test runs, and small sample size. However, with more than half of the samples coming from the Normal Distribution, a useful model can still be developed for this locking level.

The sets of samples for the “Locking Heli-Coil with Braycote” are normally distributed for minimum preloads at cycles 10 to 1000 at a significance level above 0.025; however, minimum preloads at cycles 1250 to 2250 are not normally distributed at all. Looking at both the normal plots and composite plots for the Locking Heli-Coil with Braycote, it is apparent that Run 20 is a large outlier. Outliers tend to appear at the tails of probability plots where Run 20 is seen in all of the normal plots. Looking at the composite plot in Figure 3.5 it is very noticeable that Run 20 is a sample that is very different from the other eleven samples. So Run 20 was taken out of the samples sets, and

a PPCC test was conducted again for the Locking Heli-Coil with Braycote with a sample size of 11. These new correlation coefficients are present in Table 4.3.

Table 4.3 Correlation coefficients R_p for “Locking Heli-Coil with Braycote” with sample size 11.

Cycles	Locking Heli-Coil w/ Braycote
10	0.935
250	0.945
500	0.887
750	0.965
1000	0.975
1250	0.986
1500	0.987
1750	0.986
2000	0.979
2250	0.981

After removing Run 20 from the “Locking Heli-Coil with Braycote” samples, all samples are normally distributed at a significance level of 0.01 or higher. Now that the most of the data is known to be normally distributed predictions of the population can be created.

Chapter 5

Population Predictions

5.1 Introduction

Now that the sample data has been determined to be normally distributed, as defined in Chapter 4, predictions of the population that describe the loosening of the three different locking levels can be created. These population predictions will use the sample data to produce graphical models that accurately describe the population of the samples. This is important because if the population is identified, conclusions about the loosening process for each of the three locking levels can be made. This chapter presents methods used to describe loosening of the populations themselves.

5.2 Confidence intervals

Confidence intervals are used to provide an interval that will include the true population parameter at a specified probability. This is useful when describing the population. Using the sample means and standard deviations, it is possible to describe the population's mean with a predetermined confidence level. These confidence intervals will provide a useful graphical model of loosening for the three different locking levels. Confidence intervals are created for the minimum preloads at cycles 10, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, and 2250 if the samples were determined to have come from the Normal Distribution for each of the three different locking levels. A sample size of twelve was used for the Standard Heli-Coil with Braycote and the Standard Heli-Coil

with Loctite runs. A sample size of eleven was used for the Locking Heli-Coil with Braycote runs due to the removal of Run 20 from the samples.

M-files for MatLab in Appendix E have been written to calculate the confidence intervals and plot them with their sample means of preload versus cycles for 95% and 99% confidence levels for all three locking levels. The 95% confidence intervals for minimum preloads for “Standard Heli-Coil with Braycote”, “Locking Heli-Coil with Braycote”, and “Standard Heli-Coil with Locite” test data are found using m-files *CI95std2.m*, *CI95locking2.m*, and *CI95locite2.m* respectively, and the 99% confidence intervals for minimum preloads for “Standard Heli-Coil with Braycote”, “Locking Heli-Coil with Braycote”, and “Standard Heli-Coil with Locite” test data are found using m-files *CI99std2.m*, *CI99locking2.m*, and *CI99locite2.m* respectively.

To find the minimum preloads at 10, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, and 2250 cycles, these m-files that create the confidence interval plots use the same technique as the m-files that found the correlation coefficient. See Chapter Four to reference that technique. The sample means and standard deviation of each array of minimum preloads were calculated using the MatLab functions “mean” and “std”. The ten means and ten standard deviations are put into arrays *mm* and *sdm*. Then “for” loops calculate the upper and lower confidence interval using the double sided Student’s t-distribution confidence interval equation

$$\left(\bar{x} - t_{\alpha/2, n-1} * \frac{s}{\sqrt{n}}\right) < \mu < \left(\bar{x} + t_{\alpha/2, n-1} * \frac{s}{\sqrt{n}}\right) \quad (5.1)$$

where \bar{x} is the sample mean, s is the sample standard deviation, n is the sample size, μ is the mean of the population, α is the level of confidence, and $t_{\alpha/2, n-1}$ is the Student’s t-

distribution's critical value found in a Student's t Distribution table as a function of degrees of freedom and confidence level. Critical values from the Student's t-Distribution are: 2.20 for a sample size of twelve and 95% confidence level, 2.23 for a sample size of eleven and 95% confidence level, 3.11 for a sample size twelve and 99% confidence level, and 3.17 for a sample size eleven and 99% confidence level. The upper and lower confidence intervals are then paired with matching cycles and plotted with the sample mean.

The double sided Student's t-Distribution confidence interval equation was used because the samples were normally distributed and the sample sizes were less than twenty. The Student's t-distribution is approximately equal to the Normal Distribution when sample sizes are twenty or larger; however with sample sizes less than twenty, the Student's t-distribution's area under the bell curve increases at the tails. This is because for small sample sizes, the sample standard deviation, s , is not a good estimate of the population's standard deviation σ , and so the Student's t-distribution provides different critical values for different sample sizes; whereas the Normal Distribution uses one critical value for all sample sizes.

Figures 5.1-5.6 present these plots of the confidence intervals. The sample mean is the dashed line, and the confidence interval of minimum preload at cycles 10, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, and 2250 are solid lines.

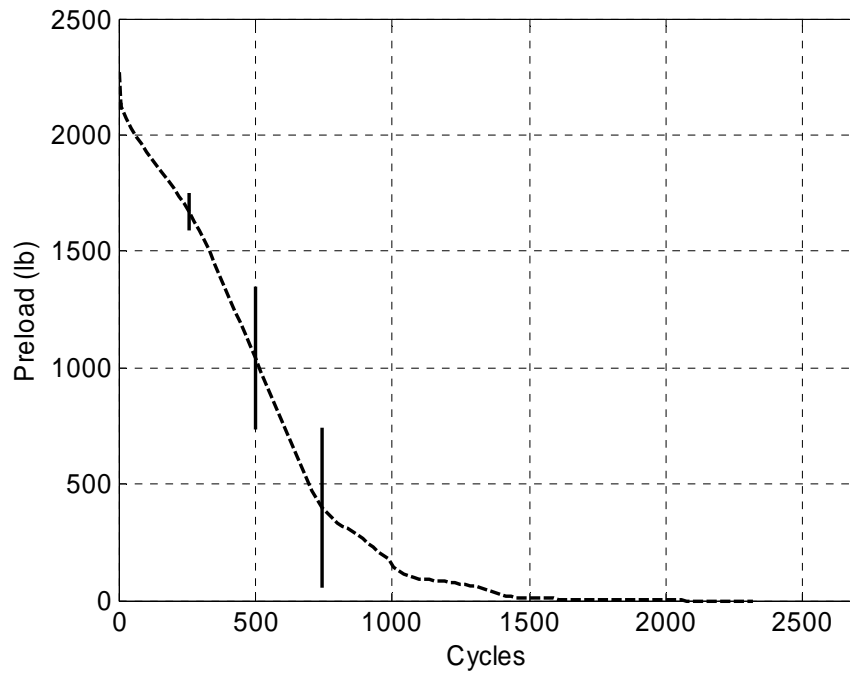


Figure 5.1 95% Confidence Intervals for the population mean of “Standard Heli-Coil with Braycote” runs.

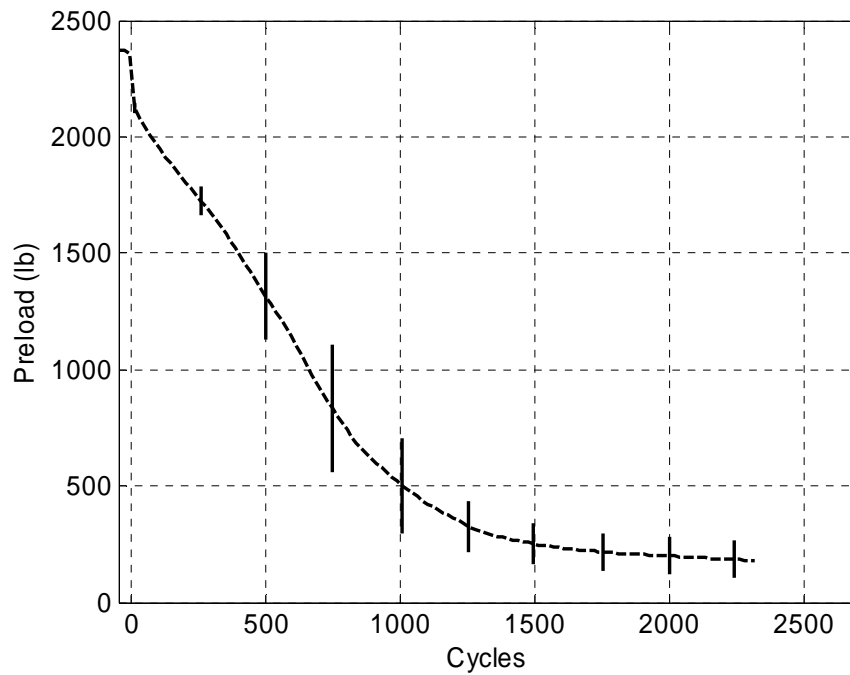


Figure 5.2 95% Confidence Intervals for the population mean of “Locking Heli-Coil with Braycote” runs.

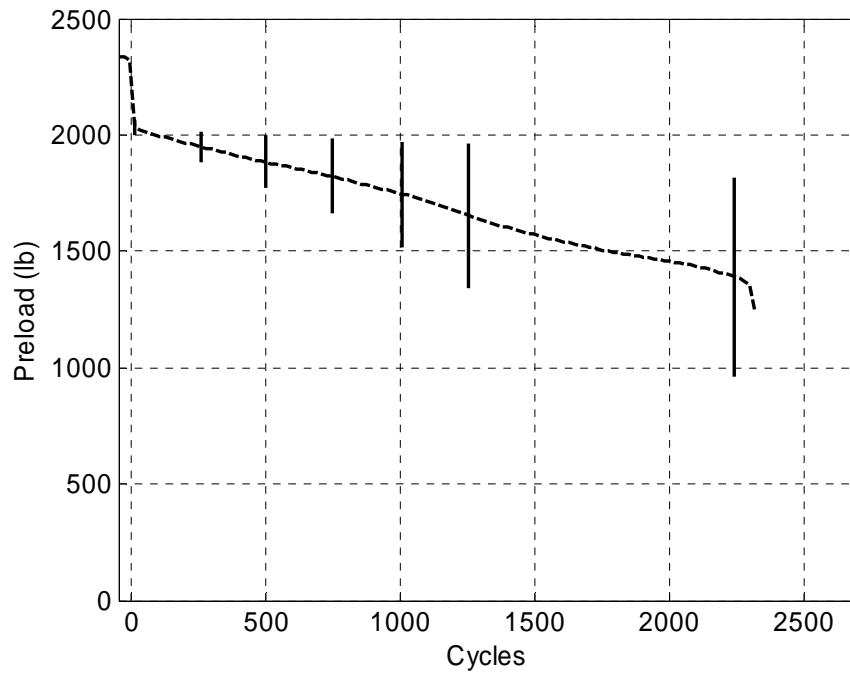


Figure 5.3 95% Confidence Intervals for the population mean of “Standard Heli-Coil with Loctite” runs.

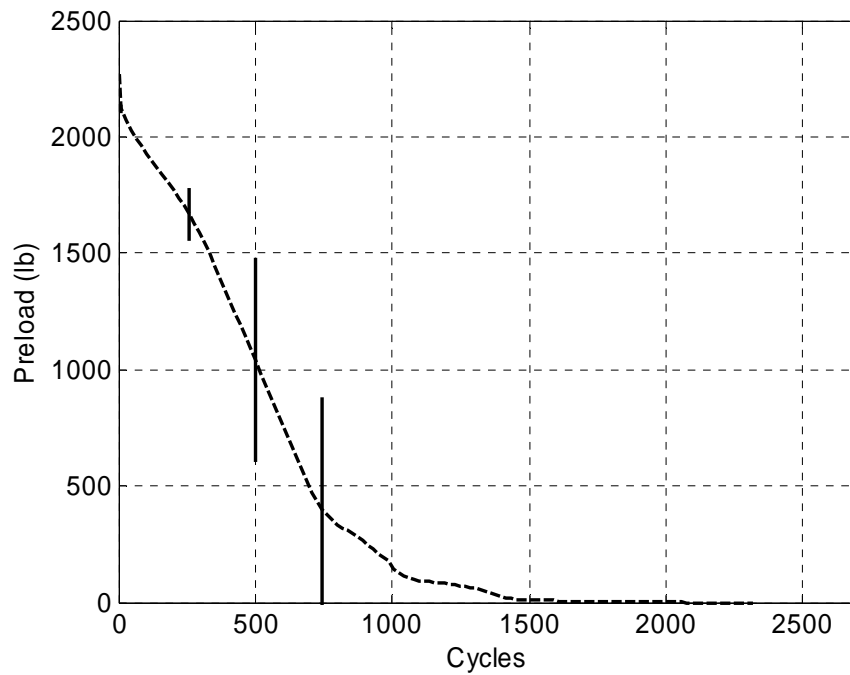


Figure 5.4 99% Confidence Intervals for the population mean of “Standard Heli-Coil with Braycote” runs.

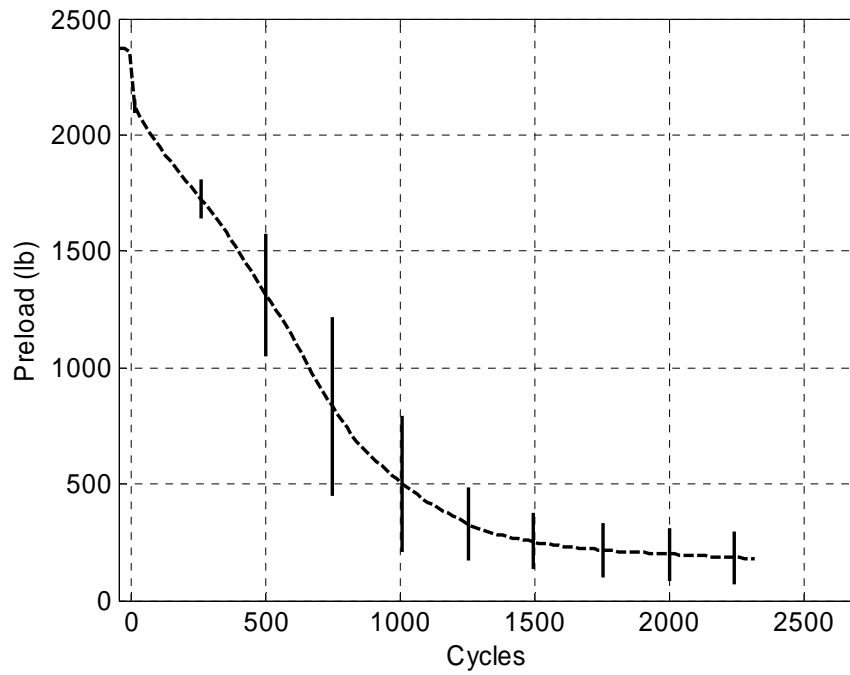


Figure 5.5 99% Confidence Intervals for the population mean of “Locking Heli-Coil with Braycote” runs.

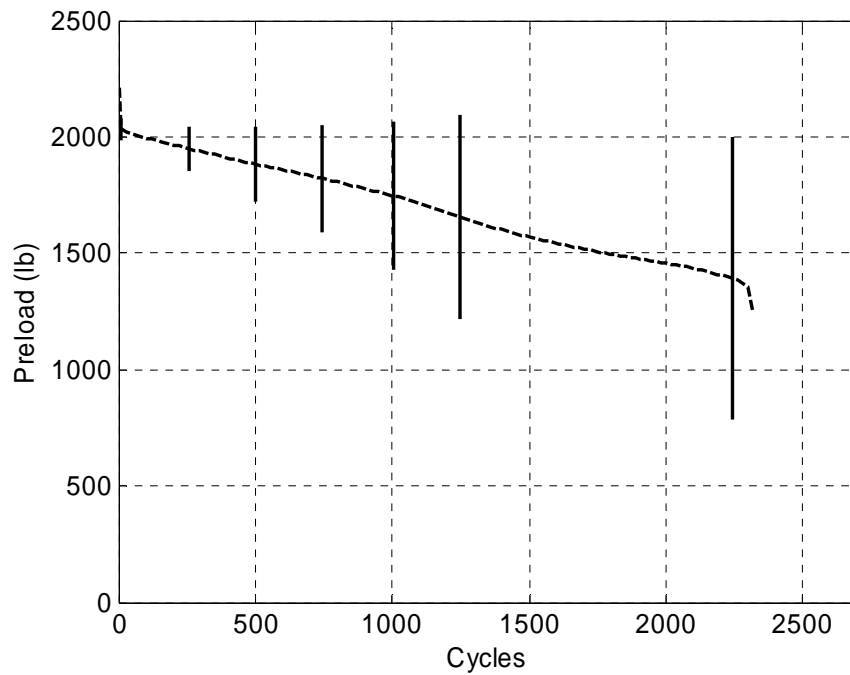


Figure 5.6 99% Confidence Intervals for the population mean of “Standard Heli-Coil with Loctite” runs.

The 95 percent confidence interval defines the population mean range with 95 percent confidence, and the 99 percent confidence interval defines the population mean range with 99 percent confidence.

5.3 Prediction intervals and tolerance intervals

Prediction intervals, sometimes known as forecast intervals, are able to provide an interval that contains a future run within the sample; which is determined from previous runs. In the case of this study, a prediction interval would predict a thirteenth sample for sample sets of twelve and would predict twelfth sample for sample sets of eleven. Using a normal two sided prediction interval, intervals were made for the same minimum preload sample sets as the normal plots and confidence intervals. However, due to larger sample standard deviations and small sample sizes, prediction intervals predicted that in a future sample the preload could increase to larger than the initial preload as cycles increase. This would indicate a possibility of tightening. Due to the fact that all test results show loosening, the results for prediction intervals have been left out of this study.

Tolerance intervals are intervals that contain a specified proportion of the population with a confidence level. This is to say that tolerance intervals contain a certain percentage of the population with a degree of confidence. These intervals are created with sample means, standard deviations, sizes, and critical values from the Normal Distribution and the Chi-Squared distributions. Tolerance intervals were also created for the same minimum preload sample set as the normal plots and confidence intervals. But just like the prediction intervals, small sample sizes and large standard deviations calculated tolerance intervals that suggest that the population may have an increase in

preload instead of decrease when cycles increase. All test results show no evidence of this occurring, instead all results show a loss of preload as cycles increase, therefore the tolerance interval results have been left out of this study.

Chapter 6

Interpretation of Results

Fastener life test data has been statistically quantified in this thesis. In this chapter, the underlying mechanisms of fastener loosening for the tests are examined. Previous analysis and the preload versus cycles data are used to identify the various mechanisms of loosening. The statistical treatments are examined to see if the characteristics inherent of the underlying mechanisms of loosening are preserved or lost. These characteristics include initial drop and changes in rate of preload loss.

This work goes beyond one test sample with twelve test samples and predicts preload means versus cycles for the sample's population. This not only allows one to predict population preload means versus cycles, but it also allows one to predict the characteristics of these curves.

First, comparisons of the confidence interval plots to those of actual sample runs will be made to determine if the confidence intervals accurately represent loosening for a particular locking level. In Figure 6.1, all of the "Standard Heli-Coil with Braycote" plots can be seen. In this figure it is apparent that the 95 percent confidence intervals, (c), and 99 percent confidence intervals, (d), are an excellent estimation for loosening. These confidence intervals display intervals that the population's mean lies within. On average (c) and (d) of Figure 6.1 shows the characteristics inherent the underlying mechanisms of loosening are preserved.

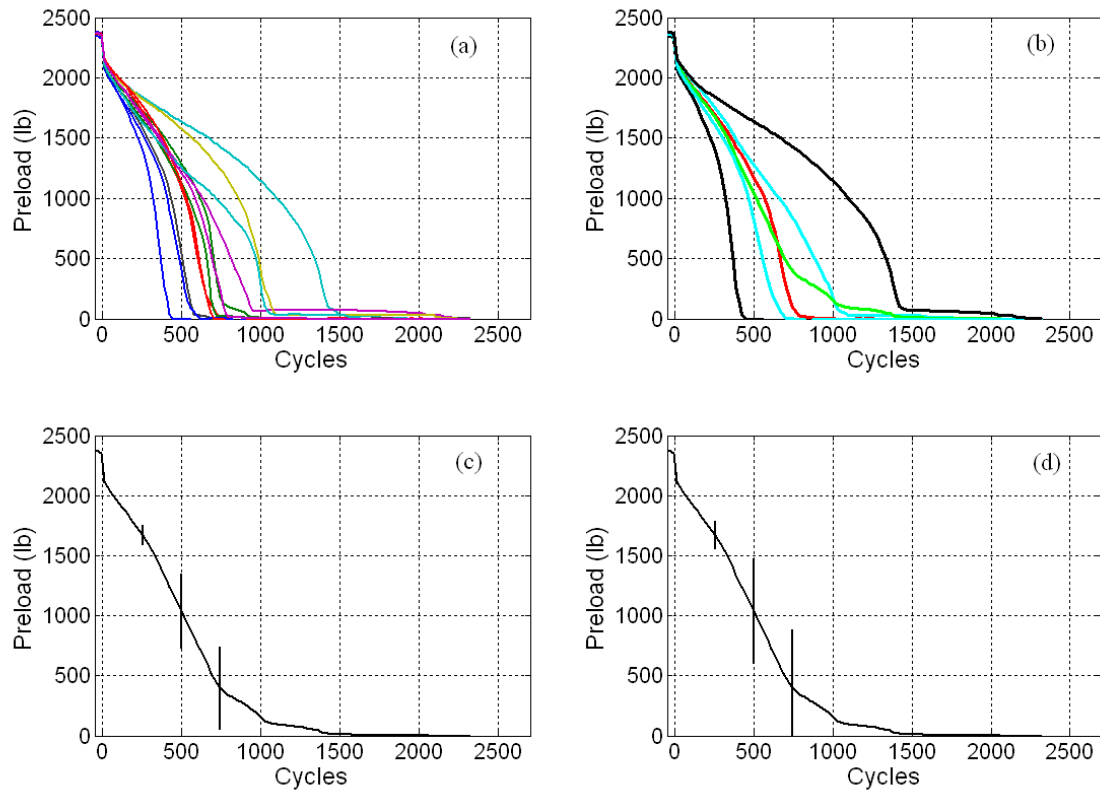


Figure 6.1 Composite Plot (a), Descriptive Statistics Plot (b), 95% Confidence Interval Plot (c), and 99% Confidence Interval Plot (d) for the “Standard Heli-Coil with Braycote” runs.

Similarly to the “Standard Heli-Coil with Braycote” plots, the confidence intervals for “Locking Heli-Coil with Braycote” preserve characteristics of loosening when comparing the individual sample runs from the composite plot, which can be observed in section (a) of Figure 6.2. Apparent in sections (a) and (b) in Figure 6.2, Run 20 lies separate from the rest of the other eleven test runs. The characteristics of loosening for Run 20 are different. With this understood, Run 20 was determined to be an outlier and was left out when calculating the confidence intervals.

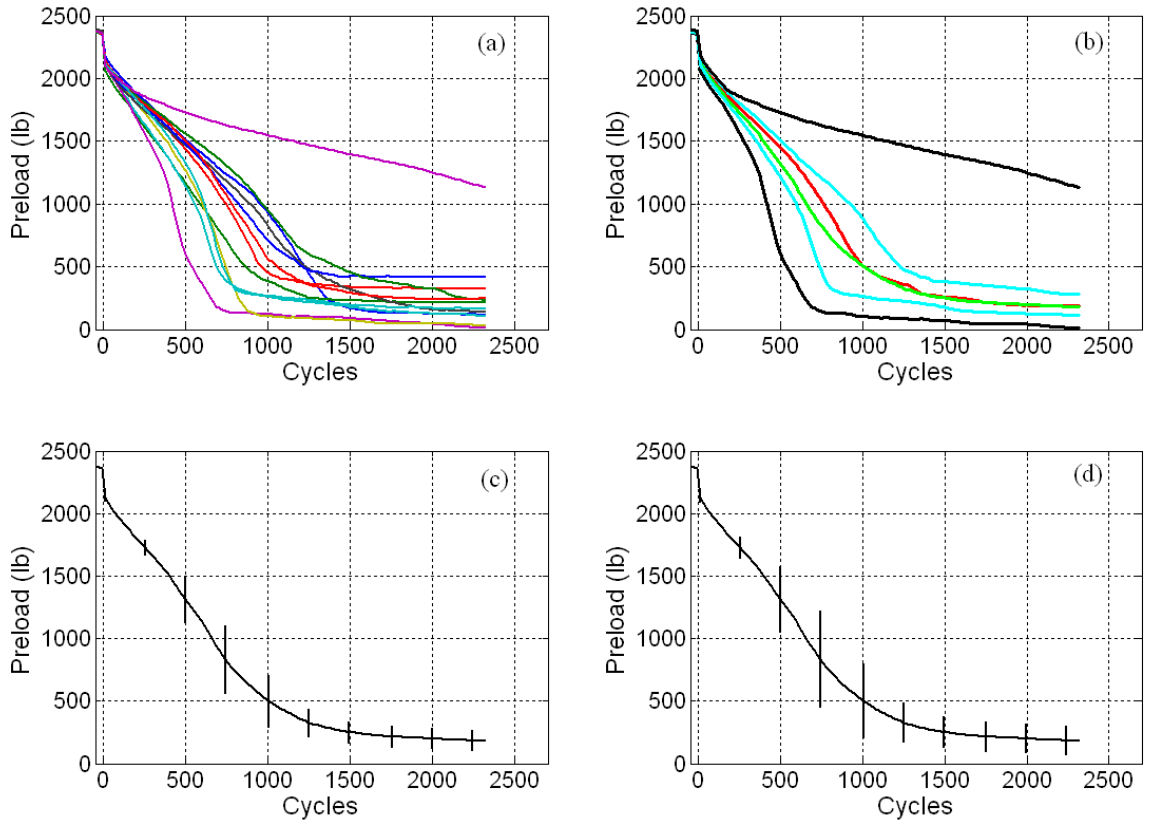


Figure 6.2 Composite Plot (a), Descriptive Statistics Plot (b), 95% Confidence Interval Plot (c), and 99% Confidence Interval Plot (d) for the “Locking Heli-Coil with Braycote” runs.

Figure 6.3 presents similar plots created for the “Standard Heli-Coil with Loctite” runs. When comparing the individual test runs to the confidence interval plots, it is apparent that the confidence intervals represent the loosening of individual samples. Three of the test runs from the “Standard Heli-Coil with Loctite” samples are seen to lose preload faster than the other nine test runs. The confidence intervals account for those different samples and suggest the population’s characteristics are a combination of all samples.

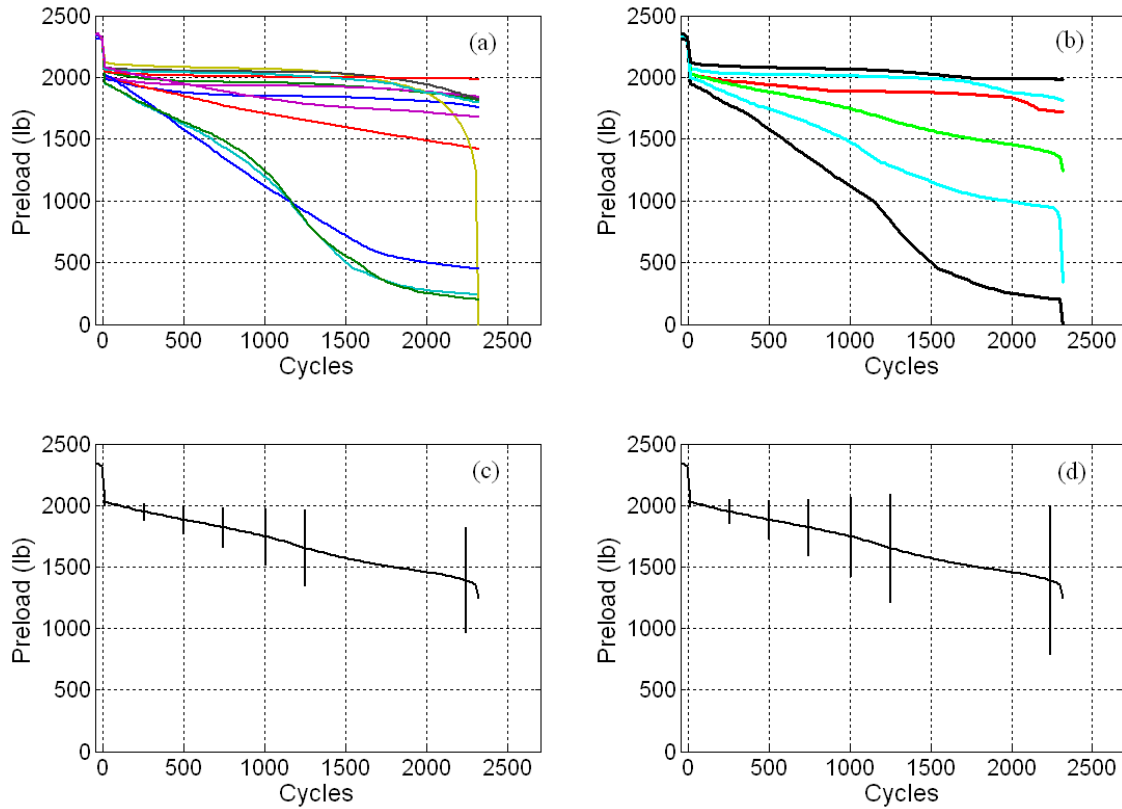


Figure 6.3 Composite Plot (a), Descriptive Statistics Plot (b), 95% Confidence Interval Plot (c), and 99% Confidence Interval Plot (d) for the “Standard Heli-Coil with Loctite” runs.

Now that the confidence intervals have been shown to capture the characteristics of the three different locking levels, estimations of what is occurring during the loosening process can be made. The Composite Plots and Descriptive Statistic Plots will be examined in regions to describe how loosening is occurring for each of the locking levels. This information will also provided insight on which secondary locking feature helps prevent the loss of preload the best.

To help explain how the “Standard Heli-Coil with Braycote” locking level loosens; the plots for the “Standard Heli-Coil with Braycote” will be divided into three regions. Looking at Region I of the “Standard Heli-Coil with Braycote” plots in Figure

6.4 it can be seen that there is 5.1 to 8.8 percent drop of initial preload when the test begins.

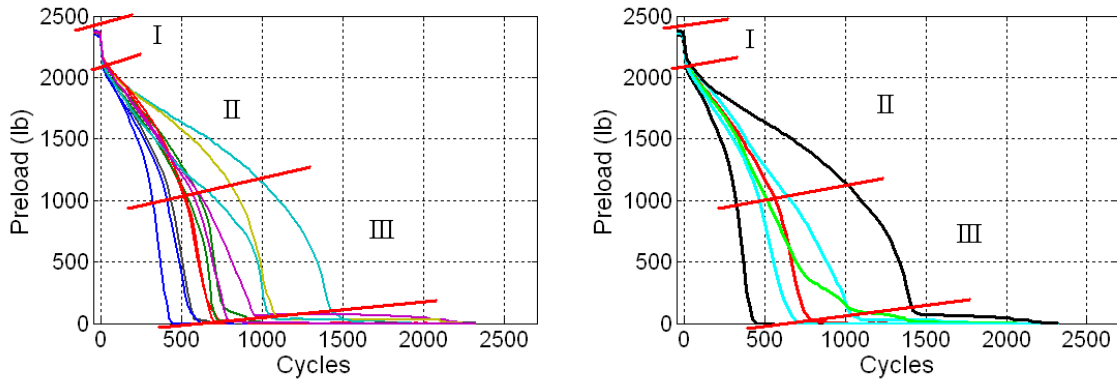


Figure 6.4 Loosening regions for “Standard Heli-Coil with Braycote” plots.

Pai and Hess provided a possible explanation to preload loss at the beginning of vibration tests. Pai and Hess [3] explained that when a tightening torque is applied to the screw head, a portion of that torque is retained by the friction forces at the head and the threads of the screw. However this stored torque also provides a loosening moment at the head of the screw when loosening begins. Region I is at the beginning of the vibration test and the seven percent preload loss occurs in the first few cycles. This moment of loosening at the head created by the applied torque helps in the loss of friction at the head and threads of the bolt resulting in complete head slip with complete or localized thread slip of the bolt at Region I. For such large rates of preload loss to occur in such a short time, the threaded fastener would have to be completely turning.

In Region II of Figure 6.4, a lower rate of preload loss occurs, compared to the loss of preload in Region I. Lower rates of preload loss would indicate that the friction at the head and threads interact more after the initial drop of preload. The preload loss in Region III is seen to be very significant and most of the preload is completely lost well

before 1000 cycles. Without a secondary locking feature the “Standard Heli-Coil with Braycote” relies on the friction at the head and threads of the screw. In Region II of Figure 6.4, the friction is overcome and slip occurs at the head and the threads. As seen in Pai and Hess’ work [3] the rates of preload loss in Region II would be caused by localized slip at the head with complete or localized slip at the threads. With larger rates of preload loss in Region III the slip at the head and threads is more than Region II, and results in complete loss of preload. So with more slip, the localized slip at the head in Region II, changes to complete slip in Region III, while the threads experience complete or localized slip.

The “Locking Heli-Coil with Braycote” plots have been divided into four regions and are presented in Figure 6.5. When comparing Figure 6.4 of the “Standard Heli-Coil with Braycote” plots with Figure 6.5 of the “Locking Heli-Coil with Braycote” plots there is a noticeable similarity in the Regions I, II, and III. This would indicate that loosening for “Locking Heli-Coil with Braycote” is similar to the loosening of the “Standard Heli-Coil with Braycote” for the first three regions of loosening. The difference between the two comes with the emergence of a transition region and Region IV in Figure 6.5. The “Locking Heli-Coil with Braycote” locking level has a transition region after Region III and before a new region, defined as Region IV, where preload loss is significantly changing.

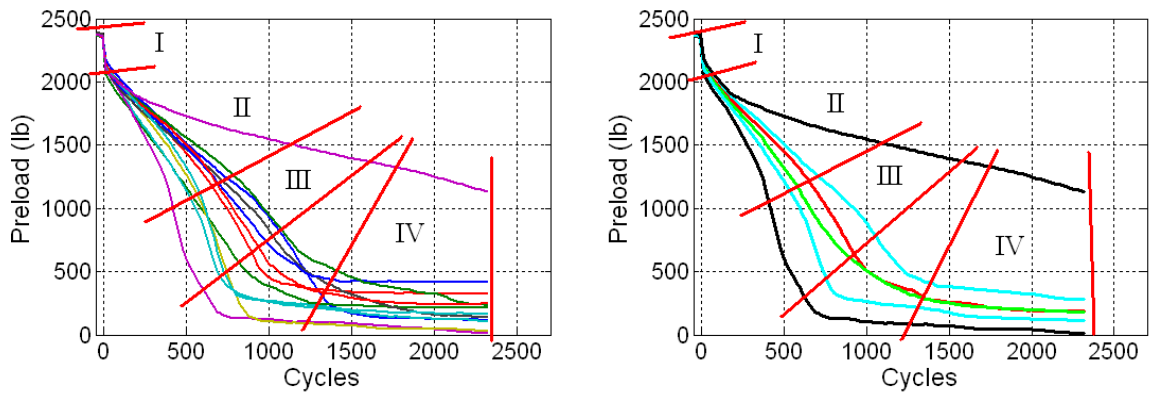


Figure 6.5 Loosening regions for “Locking Heli-Coil with Braycote” plots.

Unlike the “Standard Heli-Coil with Braycote”, the “Locking Heli-Coil with Braycote” has a secondary locking feature that creates a prevailing torque. This prevailing torque is created from a series of straight segments or “chords” in one of the insert coils [13]. When the bolt is inserted into the “grip” coil, seen in Figure 6.6, the chord segments push outward on the bolt and creates pressure on the bolt. Thus the prevailing torque comes from this pressure exerted on the bolt threads. It has been documented by Finkelston [5] that prevailing torque not only reduces the rate of preload loss but it can also stop the loosening process completely when the prevailing torque counteracts with the loosening torque. This is what is occurring in the transition region and Region IV in Figure 6.5. The prevailing torque counteracts the loosening torque and the loss of preload is reduced and stopped.

In Figure 6.5 the transition region as well as Region IV shows when the prevailing torque counteracts with the loosening torque. In the transition region the preload loss is reduced significantly which is the result of the prevailing torque. To help explain this, Figure 6.7 presents a free body diagram shows the moments of a preloaded fastener that

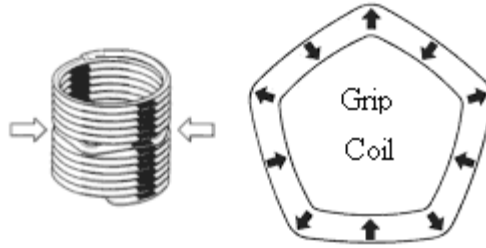


Figure 6.6 Grip Coil on “Locking Heli-Coil with Braycote” provided by [13].

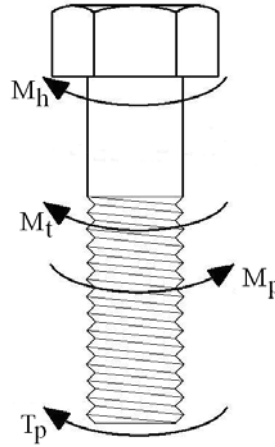


Figure 6.7 Free body diagram of fastener with prevailing torque.

has a prevailing torque. It can be observed that the prevailing torque moment, T_p , acts in the same direction as the moments created by friction at the head, M_h , and threads, M_t . M_p is a moment acting against the other moments and is created from the pitch. For this reason it is understood that the prevailing torque works against loosening.

The prevailing torque helps reduce the loss of preload throughout the fastener vibrations life test, but seems to be more effective near the end of the test. Reasons for this can be developed with a simplified version of the long-form torque equation.

$$T_{off} = F_p \left(-\frac{p}{2\pi} + \frac{\mu_t r_t}{\cos \beta} + \mu_n r_n \right) + T_p \quad (6.1)$$

Here T_{off} is the net torque tending to lessen the fastener, F_p is the preload, p is the thread pitch, μ_t is coefficient of friction in the thread, μ_n is coefficient of friction in the head, r_t is effective radii of the thread, r_n is effective radii of the head, T_p is the prevailing torque, and β is the half-angle of thread tooth. This equation shows the prevailing torque acts against the removal torque. Pai and Hess [3] use this equation to show the condition for maintaining preload in the absence of external loads.

$$F_p \frac{p}{2\pi} < F_p \frac{\mu_t r_t}{\cos(\beta)} + F_p \mu_n r_n + T_p \quad (6.2)$$

By dividing equation (6.2) through with the preload it can be observed that the prevailing torque's effectiveness is proportional to the preload.

$$\frac{p}{2\pi} > \frac{\mu_t r_t}{\cos(\beta)} + \mu_n r_n + \frac{T_p}{F_p} \quad (6.3)$$

Equation (6.3) shows the condition when loosening occurs in the absence of external loads and how the prevailing torque is affected by the preload. Though this equation is only valid in absence of external loads, it demonstrates how preload can affect the performance of the prevailing torque. It can be seen in equation (6.3) that at high preloads, F_p , the prevailing torque, T_p , contribute less, and at lower preloads, the prevailing torque will contribute more and help slow or stop loosening.

Looking at Figure 6.8, this is more apparent. Figure 6.8 shows the difference in preload of the “Locking Heli-Coil with Braycote” and the “Standard Heli-Coil with Braycote” mean and median versus the loss of preload for “Locking Heli-Coil with Braycote”. This figure shows that the prevailing torque locking feature is only more effective, than no secondary locking feature, after significant preload is lost.

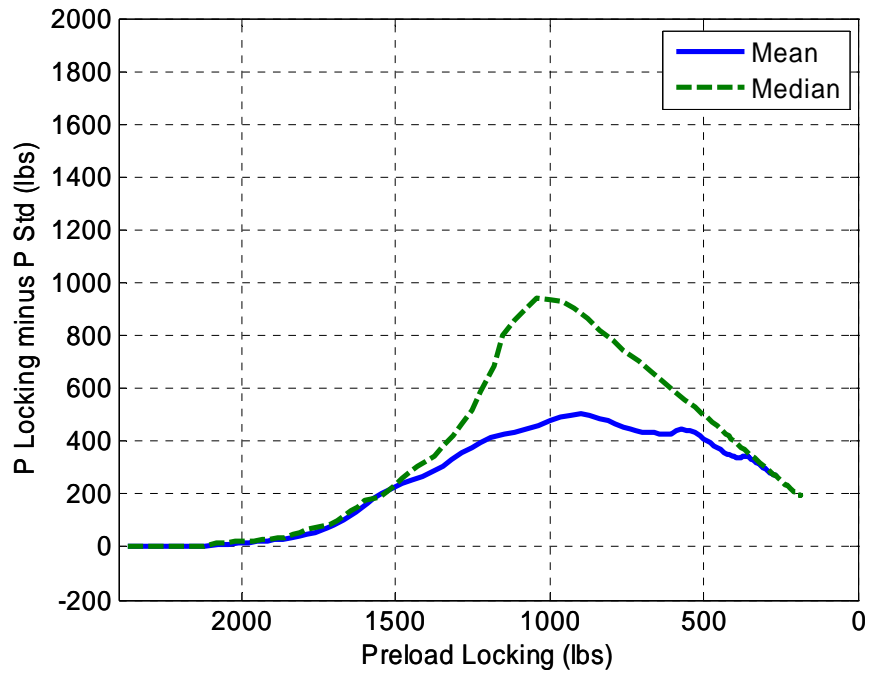


Figure 6.8 Locking minimum preload minus Std minimum preload versus loss of Locking preload.

In the transition region, the preload reaches a low enough value for the prevailing torque to be more effective, and helps slow the loss of preload. As the loosening moment counteracts the prevailing torque, complete slip, seen in Region III, becomes partial slip in the transition region. In Region IV the preload loss in some individual runs has stopped completely, but in others the preload is still lost at very slow rates. When it is apparent that preload has reached a steady value, there is neither slip at the head or threads. For the test runs that still lose preload, there is localized slip at the head and threads of the fastener.

When analyzing the “Standard Heli-Coil with Loctite” plots the loosening process was divided into three sections, which can be observed in Figure 6.9. Just like the other two locking levels, the “Standard Heli-Coil with Loctite” has an initial loss in preload in

Region I. For the “Standard Heli-Coil with Braycote” plots and the “Locking Heli-Coil with Braycote” plots, Region I of Figure 6.9 showed a loss of initial preload to be 4.6 to 9.2 percent, but the loss of initial preload of the “Standard Heli-Coil with Loctite” plots is higher at 7.9 to 14.4 percent. This gross loss of preload in Region I for all three locking levels can be contributed to complete slip at the head and complete or localized slip at the threads.

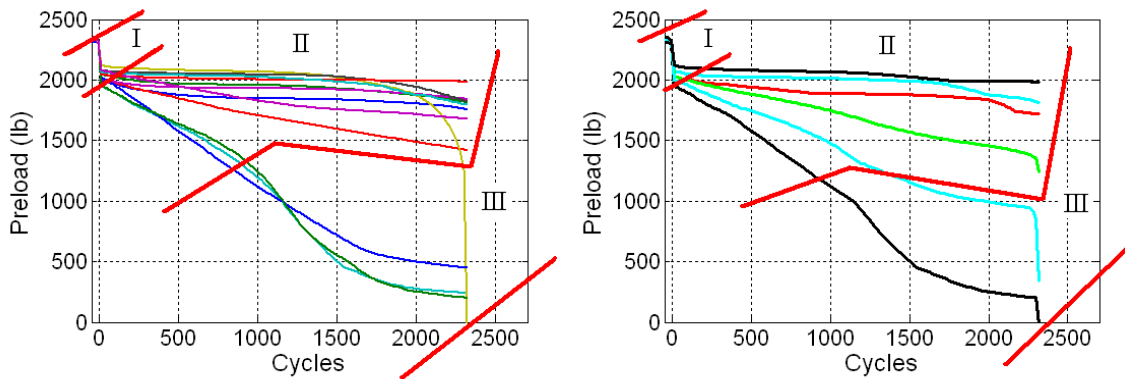


Figure 6.9 Loosening regions for “Standard Heli-Coil with Loctite” plots.

In Region II in Figure 6.9, it can be observed that rates of preload loss are slow, which would indicate that the complete slip in Region I has become localized slip. For most of the test runs in this sample, localized slip at the heads and threads occurs from Region I through Region II. An increase in preload loss rate occurs for three of the twelve tests as identified by Region III in Figure 6.9. For these three tests a transition to complete head and/or thread slip occurs. Some of the reasons that these three tests have different characteristics may be due to: curing issues of the adhesive, cleaning issues, and assembly issues.

The secondary locking feature for the “Standard Heli-Coil with Loctite” locking level is in the form of a thread locking adhesive. This adhesive “Loctite” is applied to the

threads during assembly of each sample of the “Standard Heli-Coil with Loctite” samples. Advertised by Henkel Corporation [18], Loctite threadlocker is an anaerobic liquid that cures to a hard thermoset plastic that locks the threads together. In the “Standard Heli-Coil with Loctite” test runs, all runs loosen to some degree, which suggest that this threadlocker does not completely lock the threads together. When the threadlocker cures into a solid, the voids and gaps where air would usually be present without threadlocker, are now filled with a solid. Just like tighter thread tolerances, the voids and gaps are filled at the threads, which improves the resistance to vibration-induced loosening [19]. This explains the low rate of preload loss in Region II.

Figure 6.9, similar to Figure 6.8, shows the difference in preload of the “Standard Heli-Coil with Loctite” and the “Standard Heli-Coil with Braycote” mean and median versus the loss of preload for “Standard Heli-Coil with Loctite”. In this figure it appearant that the adhesive secondary locking feature helps prevent the loss of preload near intial preload when compaired to no locking feature or the prevailing torque secondary locking feature. After the intial preload drop the adhesive maintains more preload which can be seen in Figure 6.9. As the difference in preload increase the loss of preload for “Standard Heli-Coil with Loctite” samples remains relatively slow.

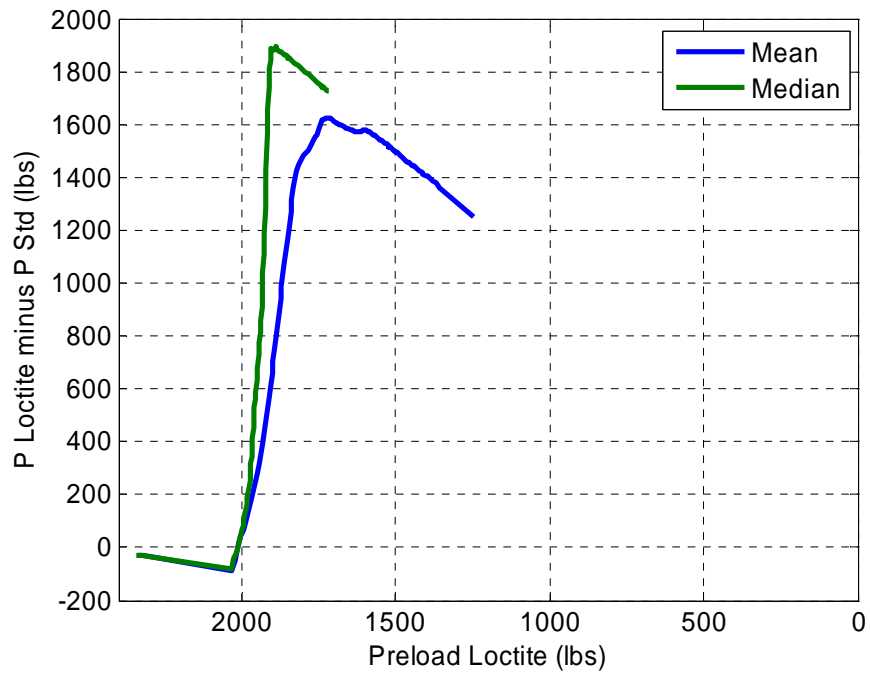


Figure 6.10 Loctite minimum preload minus Std minimum preload versus loss of Loctite preload.

The overall assessment of the three locking levels can be developed by looking at the Confidence Interval plots. Figure 6.9 and Figure 6.10 respectively show the 95 percent Confidence Intervals and the 99 percent Confidence Intervals for all three locking levels. These figures indicate that the “Standard Heli-Coil with Loctite” locking level is the best when trying to retain initial preload. The next best locking level is the “Locking Heli-Coil with Braycote” followed by the “Standard Heli-Coil with Braycote” locking level. These figures also show how the programs created in this thesis can aid in the comparison of different threaded fasteners.

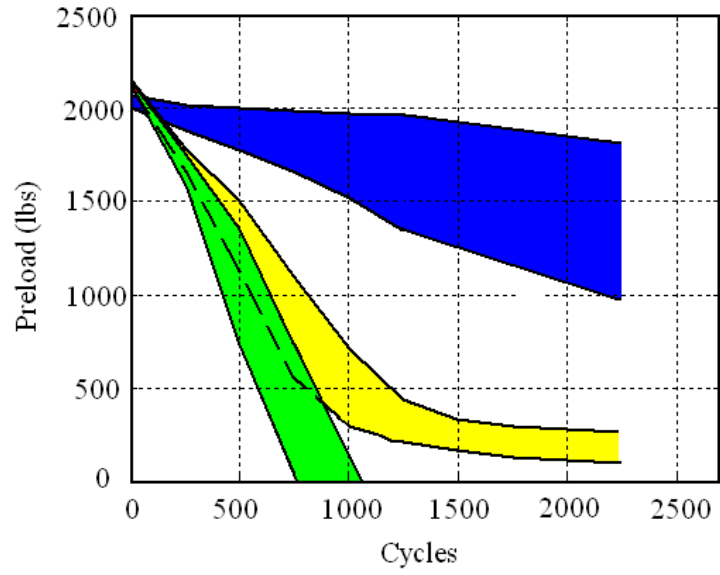


Figure 6.11 95 percent Confidence Intervals for all three locking levels.

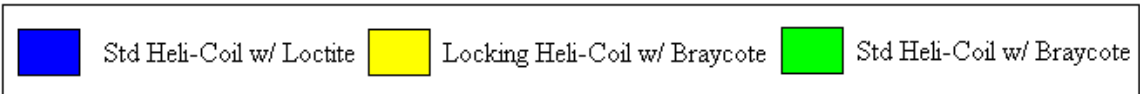
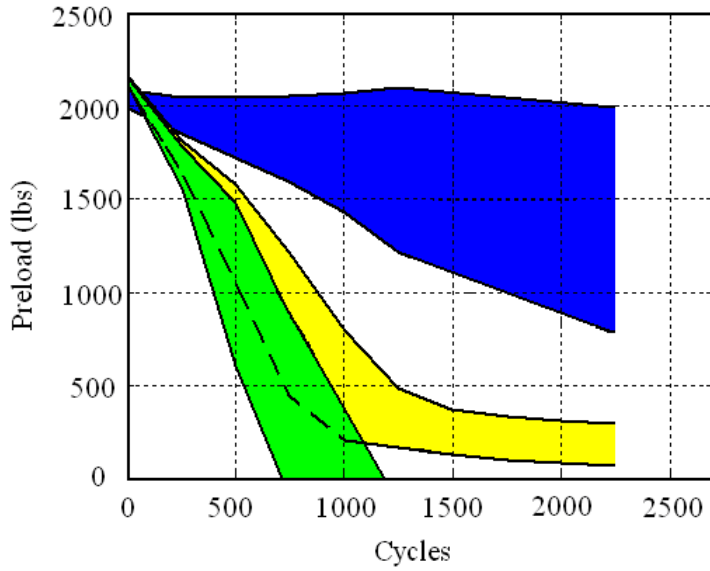


Figure 6.12 99 percent Confidence Intervals for all three locking levels.

Chapter 7

Conclusion

The methods developed in this work provide quantified and statistical test results for tests of threaded fasteners subjected to vibration life tests. The tools in this thesis allow one to quantify multiple test samples from vibration life tests. In previous work life tests are presented individually. In this work, fastener vibration life test data of threaded inserts with different secondary locking features was obtained to help develop these methods of quantification.

First, methods to summarize the test data of vibration life test were developed. This was done by plotting the minimum preloads versus cycles for all runs in the same sample in one plot. The minimum preloads are usually of most interest in fastener vibration life test, which is the main reason the minimum preload was extracted. In addition to sample composite plots, sample statistics include sample mean, sample median, sample quartiles, and sample extents were determined.

Predictions for sample population were estimated. To do this the distribution of the sample data was assessed. Using normal plots and PPCC test it was found that the data is approximately normal. This allowed predictions of the sample population to be predicted. Using a double sided Student's t-distribution confidence interval, intervals for which the population's mean lies have been created for the minimum preload versus cycle data. It has been shown that these confidence intervals do preserve the characteristics

inherent of the underlying mechanisms of loosening when comparing them to individual test runs.

Finally in Chapter 6, the shapes of the loosening curves are examined. It was found that the plots generated in this work do preserve the characteristics inherent of the underlying mechanisms of loosening. These plots were then used to aid in the evaluation of the loosening characteristics for the three locking levels. It was also found that having a secondary locking feature does help retain preload loss. The prevailing torque secondary locking feature for the “Locking Heli-Coil with Braycote” test runs is found to be more effective as the preload is lost. The best secondary locking feature was found to be the Loctite Threadlocker.

References

1. Hess, D. P., "Vibration- and Shock- Induced Loosening," Chapter 40 in *Handbook of Bolts and Bolted Joints*, Marcel Dekker Inc., New York, pp. 757-824, 1998a.
2. Junker, G. H., "New Criteria for Self-Loosening of Fasteners Under Vibration," *Society of Automotive Engineers Transactions*, Vol. 78, pp. 314-335, 1969.
3. Pai, N.P. and Hess, D.P., "Experimental Study of Loosening of Threaded Fasteners Due to Dynamic Shear Load," *Journal of Sound and Vibration*, Vol. 253, pp. 585-692, 002.
4. Pai, N.P. and Hess, D.P., "Three-Dimensional Finite Element analysis of threaded fastener loosening due to dynamic shear load," *Engineering Failure Analysis*, Vol. 9, pp. 383-402, 2002.
5. Finkelston, R. F., "How Much Shake Can Bolted Joints Take," *Machine Design*, pp. 122–125, 1972.
6. Sanclemente, J.A. and Hess, D.P., "Parametric Study of Threaded Fastener Loosening Due to Cyclic Transverse Load," *Engineering Failure Analysis*, Vol. 14, pp. 239-249, 2007.
7. "Nord-Lock Bolt securing system", <http://www.nord-lock.com>, Accessed 2007.
8. Faroni, C. C., "Maintaining Tightness of Threaded Fasteners", Amerace Corporation, ESNA Division, Union, NJ, 1967.
9. "A Statistical Look at Bolt Preload, Friction and Relaxation", SME Technical Paper, 1979.
10. Collins, J.A., *Mechanical Design of Machine Elements and Machines: A Failure Prevention Perspective*, New York: John Wiley & Sons, Inc., 2003.
11. NAS 1003 thru 1020, National Aerospace Standard, pp. 1-3, 1991.
12. NAS 1149, National Aerospace Standard, pp. 1-6, 1994.
13. Emhart Teknologies Heli-Coil Bulletin, <http://www.emhart.com>, 2003

14. MIL-STD-1312-7A, 1984, Fastener Test Methods - Method 7: Vibration, U.S. Department of Defense, Washington D.C.
15. Montgomery, D. C., *Design and Analysis of Experiments*, 6th ed., New Jersey: John Wiley & Sons, Inc., 2005.
16. Looney S. W. and Gullledge Jr. T. R., "Use of the Correlation Coefficient with Normal Probability Plots", *The American Statistician*, Vol. 39, No. 1, pp. 75-79, 1985.
17. Bickford, J.H., *An Introduction to the Design and Behavior of Bolted Joints*, 3rd ed., Marcel Dekker, 1995.
18. *The Adhesive Sourcebook*, Vol. 7, Henkel Technologies-Loctite, 2007.
19. Hess, D. P., "Mechanism of Vibration-Induced Loosening," *Fastener Technology International*, Vol. XXII, 1999, pp. 86-87.

Bibliography

1. Abramowitz, M. and Stegun, I. A., *National Bureau of Standards Applied Mathematics Series, 55: Handbook of Mathematical Functions*, Washington, DC: US Government Printing Office, 1964.
2. Daily, J. W., Riley, W. F., and McConnell, K. G., *Instrumentation for Engineering Measurements*, 2nd ed. New Jersey: John Wiley & Sons, Inc, 1993.
3. Filliben, J. J., "The Probability Plot Correlation Coefficient Test for Normality," *Technometrics*, Vol. 17, No. 1, pp. 111-117, 1975.
4. Krishnamoorthy, K., *Handbook of Statistical Distributions with Applications*, Florida: Chapman & Hall/CRC Press, 2006.
5. Patel, J. K. and Read, C. B., *Handbook of the Normal Distribution*, 2nd ed. New York: Marcel Dekker, Inc, 1996.

Appendices

Appendix A: M-Files for Composite Plots

The M-files *minstdlines2.m*, *minlockinglines2.m*, and *minloctitelines2.m* create a composite plot of all runs for each of the locking levels. These composite plots show the minimum preload versus cycles for the runs. The raw data from the runs are recorded into files. These M-files load the raw data files and assigns time, preload, and cycles to proper calibrated values. With a sampling rate of 51.2 samples/second there are a total of 8192 data points for time, preload, and cycles. The minimum preloads for every 18.75 cycles are found using “for” loops. Every 64 data points are grouped into arrays 128 times for a total of 8192 data points for both preload and cycles. The function “min” is used to sort the preload array from least to greatest. The first element in the array is then put into another array $C_i(j)$. The cycle arrays do not need to be sorted and the median of the cycles array is assigned to the array $D_i(j)$. Then the preload arrays $C_i(j)$ are plotted against the cycle’s arrays $D_i(j)$.

M-File *minstdlines2.m* begins here.

```
% Minimum Preloads of Test data from Junker machine

% "mdp" is the total number of data points to be plotted. "dpc"
% is the data point count. It controls the range of data points
% to be selected to choose a specific data point. "dpcm" is the
% data point count median. It is use to select the median cycles
% to beplotted against the lowest data point from each group.
% "dpcm" could be change to be the end lowest cycles as well by
% making it equal to "dpc".

clear all

mdp=128;
dpc=8192/mdp;
dpcm=dpc/2;
```

Appendix A: (Continued)

```
% C arrays are the minimum preloads for each range of data  
% points.  
% D arrays are the medians of cycles for each range of data  
% points.
```

```
load test1.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C1(i)=p(1);  
d=cycles(a:b);  
D1(i)=d(dpcm);  
end;
```

```
load test3.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C2(i)=p(1);  
d=cycles(a:b);  
D2(i)=d(dpcm);  
end;
```


Appendix A: (Continued)

```
load test4.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
D3(i)=d(dpcm);
end;

load test7.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
end;

load test8.vna -mat;
```

Appendix A: (Continued)

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C5(i)=p(1);  
d=cycles(a:b);  
D5(i)=d(dpcm);  
end;
```

```
load test11.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C6(i)=p(1);  
d=cycles(a:b);  
D6(i)=d(dpcm);  
end;
```

```
load test15.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;
```

Appendix A: (Continued)

```
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);

p=min(p);
C7(i)=p(1);
d=cycles(a:b);
D7(i)=d(dpcm);
end;

load test21.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C8(i)=p(1);
d=cycles(a:b);
D8(i)=d(dpcm);
end;

load test25.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
```

Appendix A: (Continued)

```
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C9(i)=p(1);
d=cycles(a:b);
D9(i)=d(dpcm);
end;

load test26.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C10(i)=p(1);
d=cycles(a:b);
D10(i)=d(dpcm);
end;

load test28.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;
```

Appendix A: (Continued)

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C11(i)=p(1);
d=cycles(a:b);
D11(i)=d(dpcm);
end;

load test33.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;

% Plots of all Runs are plotted together. Plots show the minimum
% preload versus cycles.

axes('FontSize',16);
plot(D1,C1,D2,C2,D3,C3,D4,C4,D5,C5,D6,C6,D7,C7,D8,C8,D9,C9,D10,C1
0,D11,C11,D12,C12,'LineWidth',2);
axis([-50 2700 0 2500]);
ylabel('Preload (lb)','FontSize',18);
```

Appendix A: (Continued)

```
xlabel('Cycles','FontSize',18);  
legend('Test 1','Test 3','Test 4','Test 7','Test 8','Test  
11','Test 15','Test 21','Test 25','Test 26','Test 28','Test 33');  
grid;
```

M-File *minstdlines2.m* ends here.

M-File *minlockinglines2.m* begins here.

```
% Minimum Preloads of Test data from Junker machine  
  
% "mdp" is the total number of data points to be plotted. "dpc"  
% is the data point count. It controls the range of data points  
% to be selected to choose a specific data point. "dpcm" is the  
% data point count median. It is use to select the median cycles  
% to be plotted against the lowest data point from each group.  
% "dpcm" be change to be the end lowest cycles as well by making  
% it equal to "dpc".
```

```
clear all
```

```
mdp=128;  
dpc=8192/mdp;  
dpcm=dpc/2;
```

```
% C arrays are the minimum preloads for each range of data  
% points.  
% D arrays are the medians of cycles for each range of data  
% points.
```

```
load test2.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C1(i)=p(1);
```

Appendix A: (Continued)

```
d=cycles(a:b);
D1(i)=d(dpcm);
end;

load test9.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C2(i)=p(1);
d=cycles(a:b);
D2(i)=d(dpcm);
end;

load test10.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
D3(i)=d(dpcm);
end;
```

Appendix A: (Continued)

```
load test14.vna -mat;

time = SLm.tdxvec;

preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
end;

load test16.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C5(i)=p(1);
d=cycles(a:b);
D5(i)=d(dpcm);
end;
```


Appendix A: (Continued)

```
load test18.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);

p=min(p);
C6(i)=p(1);
d=cycles(a:b);
D6(i)=d(dpcm);
end;

load test24.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C7(i)=p(1);
d=cycles(a:b);
D7(i)=d(dpcm);
end;
```

Appendix A: (Continued)

```
load test27.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C8(i)=p(1);
d=cycles(a:b);
D8(i)=d(dpcm);
end;

load test30.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C9(i)=p(1);
d=cycles(a:b);
D9(i)=d(dpcm);
end;

load test32.vna -mat;
```

Appendix A: (Continued)

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C10(i)=p(1);  
d=cycles(a:b);  
D10(i)=d(dpcm);  
end;
```

```
load test35.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C11(i)=p(1);  
d=cycles(a:b);  
D11(i)=d(dpcm);  
end;
```

```
load test36.vna -mat;
```

```
time = SLM.tdxvec;
```

Appendix A: (Continued)

```
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;

% Plots of all Runs are plotted together. Plots show the minimum
% preload versus cycles.

axes('FontSize',16);
plot(D1,C1,D2,C2,D3,C3,D4,C4,D5,C5,D6,C6,D7,C7,D8,C8,D9,C9,D10,C10,D11,C11,D12,C12,'LineWidth',2);
axis([-50 2700 0 2500]);
ylabel('Preload (lb)','FontSize',18);
xlabel('Cycles','FontSize',18);
legend('Test 2','Test 9','Test 10','Test 14','Test 16','Test 18','Test 24','Test 27','Test 30','Test 32','Test 35','Test 36');
grid;

M-File minlockinglines2.m ends here.

M-File minloctitelines2.m begins here.

% Minimum Preloads of Test data from Junker machine

% "mdp" is the total number of data points to be plotted. "dpc"
% is the data point count. It controls the range of data points
% to be selected to choose a specific data point. "dpcm" is the
% data point count median. It is use to select the median cycles
% to be plotted against the lowest data point from each group.
% "dpcm" could be change to be the end lowest cycles as well by
% making it equal to "dpc".

clear all
```

Appendix A: (Continued)

```
mdp=128;
dpc=8192/mdp;
dpcm=dpc/2;

% C arrays are the minimum preloads for each range of data
% points.
% D arrays are the medians of cycles for each range of data
% points.

load test5.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C1(i)=p(1);
d=cycles(a:b);
D1(i)=d(dpcm);
end;

load test6.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
```

Appendix A: (Continued)

```
end;
p=preload(a:b);
p=min(p);
C2(i)=p(1);
d=cycles(a:b);
D2(i)=d(dpcm);
end;

load test12.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
D3(i)=d(dpcm);
end;

load test13.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
```

Appendix A: (Continued)

```
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
end;

load test17.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C5(i)=p(1);
d=cycles(a:b);
D5(i)=d(dpcm);
end;

load test19.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C6(i)=p(1);
d=cycles(a:b);
D6(i)=d(dpcm);
end;
```

Appendix A: (Continued)

```
load test20.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C7(i)=p(1);
d=cycles(a:b);
D7(i)=d(dpcm);
end;

load test22.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C8(i)=p(1);
d=cycles(a:b);
D8(i)=d(dpcm);
end;

load test23.vna -mat;
```


Appendix A: (Continued)

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C9(i)=p(1);  
d=cycles(a:b);  
D9(i)=d(dpcm);  
end;
```

```
load test29.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C10(i)=p(1);  
d=cycles(a:b);  
D10(i)=d(dpcm);  
end;
```

```
load test31.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

Appendix A: (Continued)

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C11(i)=p(1);
d=cycles(a:b);
D11(i)=d(dpcm);
end;

load test34.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;
% Plots of all Runs are plotted together. Plots show the minimum
% preload versus cycles.
axes('FontSize',16);
plot(D1,C1,D2,C2,D3,C3,D4,C4,D5,C5,D6,C6,D7,C7,D8,C8,D9,C9,D10,C1
0,D11,C11,D12,C12,'LineWidth',2);
axis([-50 2700 0 2500]);
ylabel('Preload (lb)','FontSize',18);
xlabel('Cycles','FontSize',18);
legend('Test 5','Test 6','Test 12','Test 13','Test 17','Test
19','Test 20','Test 22','Test 23','Test 29','Test 31','Test 34');
grid;
```

M-File *minloctitelines2.m* ends here.

Appendix B: M-files for Sample Statistic Plots

M-files *quanstd2.m*, *quanlocking2.m*, and *quanloctite2.m* plot sample means, medians, 25% quartiles, 75% quartiles, and extents of minimum preloads versus cycles for all three locking levels. The data for each run is loaded, and minimum preloads for every 18.75 cycles are found using “for” loops. Every 64 data points are grouped into arrays 128 times for a total of 8192 data points for both preload and cycles. The function “min” is used to sort the preload array from least to greatest. The first element in the array is then put into another array $C_i(j)$. The cycle arrays do not need to be sorted and the median of the cycle’s array is assigned to the array $D\#(i)$. Then minimum preloads for all runs are put into matrix td in a manner in which each row is the minimum preload for all runs at a particular cycle. Then another for loop selects each row of the matrix td and extracts the extents, medians, 25% quartile, and 75% quartiles from each row and puts them into arrays. The medians and quartiles are found using the MatLab function “quantile” and the extents are found using the MatLab function “min” and “max”. Then the arrays are plotted against a cycle’s array $D(i)$.

M-file *quanstd2.m* begins here.

```
% Mean, median, quartiles, and extents of minimum preloads.

% "mdp" is the total number of data points to be plotted. "dpc"
% is the data point count. It controls the range of data points
% to be selected to choose a specific data point. "dpcm" is the
% data point count median. It is use to select the median cycles
% to be plotted against the lowest data point from each group.
% "dpcm" could be change to be the end lowest cycles as well by
% making it equal to "dpc".

clear all

mdp=128;
```

Appendix B: (Continued)

```
dpc=8192/mdp;
dpcm=dpc/2;

% C arrays are the minimum preloads for each range of data
% points.
% D arrays are the medians of cycles for each range of data
% points.

load test1.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C1(i)=p(1);
d=cycles(a:b);
D1(i)=d(dpcm);
end;

load test3.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C2(i)=p(1);
```

Appendix B: (Continued)

```
d=cycles(a:b);
D2(i)=d(dpcm);
end;

load test4.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
D3(i)=d(dpcm);
end;

load test7.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
```

Appendix B: (Continued)

```
end;

load test8.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C5(i)=p(1);
d=cycles(a:b);
D5(i)=d(dpcm);
end;

load test11.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C6(i)=p(1);
d=cycles(a:b);
D6(i)=d(dpcm);
end;
```

Appendix B: (Continued)

```
load test15.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C7(i)=p(1);
d=cycles(a:b);
D7(i)=d(dpcm);
end;

load test21.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C8(i)=p(1);
d=cycles(a:b);
D8(i)=d(dpcm);
end;
```

Appendix B: (Continued)

```
load test25.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C9(i)=p(1);
d=cycles(a:b);
D9(i)=d(dpcm);
end;

load test26.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C10(i)=p(1);
d=cycles(a:b);
D10(i)=d(dpcm);
end;

load test28.vna -mat;
```


Appendix B: (Continued)

```
time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C11(i)=p(1);
d=cycles(a:b);
D11(i)=d(dpcm);
end;

load test33.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;

td=[C1;C2;C3;C4;C5;C6;C7;C8;C9;C10;C11;C12];
td=transpose(td);

sum=C1+C2+C3+C4+C5+C6+C7+C8+C9+C10+C11+C12;
mean1=sum/12;
```

Appendix B: (Continued)

```
for i=1:mdp
m=[td(i,1);td(i,2);td(i,3);td(i,4);td(i,5);td(i,6);td(i,7);td(i,8)
);td(i,9);td(i,10);td(i,11);td(i,12)];
    med = quantile(m, 0.5);
    upq = quantile(m, 0.75);
    loq = quantile(m, 0.25);
    mp=min(m);
    mpl=max(m);
    medd(i)=med;
    upqd(i)=upq;
    loqd(i)=loq;
    mind(i)=mp(1);
    maxd(i)=mpl(1);
end;
    medd = transpose(medd);
    upqd = transpose(upqd);
    lowq = transpose(loqd);
    mind=transpose(mind);
    maxd=transpose(maxd);
```

```
axes('FontSize',16);
plot(D1,mind,'--k',D1,upqd,':c',D1,medd,'-r',D1,mean1,'-g',D1,
loqd,':c',D1,maxd,'--k','LineWidth',2.5);
axis([-50 2700 0 2500]);
ylabel('Preload (lb)','FontSize',18);
xlabel('Cycles','FontSize',18);
%legend('Extent Curves','Upper and Lower
Quartiles','Median','Mean');
grid;
```

M-file *quanstd2.m* ends here.

M-file *quanlocking2.m* begins here.

```
% Mean, median, quartiles, and extents of minimum preloads.
```

```
% "mdp" is the total number of data points to be plotted. "dpc"
% is the data point count. It controls the range of data points
% to be selected to choose a specific data point. "dpcm" is the
% data point count median. It is use to select the median cycles
% to be plotted against the lowest data point from each group.
% "dpcm" could be change to be the end lowest cycles as well by
% making it equal to "dpc".
```

```
clear all
```

Appendix B: (Continued)

```
mdp=128;
dpc=8192/mdp;
dpcm=dpc/2;

% C arrays are the minimum preloads for each range of data
% points.
% D arrays are the medians of cycles for each range of data
% points.

load test2.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C1(i)=p(1);
d=cycles(a:b);
D1(i)=d(dpcm);
end;

load test9.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
```

Appendix B: (Continued)

```
p=preload(a:b);
p=min(p);
C2(i)=p(1);
d=cycles(a:b);
D2(i)=d(dpcm);
end;

load test10.vna -mat;

time = SLm.tdxvec;

preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
D3(i)=d(dpcm);
end;

load test14.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
```

Appendix B: (Continued)

```
p=min(p);
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
end;

load test16.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C5(i)=p(1);
d=cycles(a:b);
D5(i)=d(dpcm);
end;

load test18.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C6(i)=p(1);
```

Appendix B: (Continued)

```
d=cycles(a:b);
D6(i)=d(dpcm);
end;

load test24.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C7(i)=p(1);
d=cycles(a:b);
D7(i)=d(dpcm);
end;

load test27.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C8(i)=p(1);
d=cycles(a:b);
D8(i)=d(dpcm);
```

Appendix B: (Continued)

```
end;

load test30.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C9(i)=p(1);
d=cycles(a:b);
D9(i)=d(dpcm);
end;

load test32.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C10(i)=p(1);
d=cycles(a:b);
D10(i)=d(dpcm);
end;
```

Appendix B: (Continued)

```
load test35.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C11(i)=p(1);
d=cycles(a:b);
D11(i)=d(dpcm);
end;

load test36.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;

td=[C1;C2;C3;C4;C5;C6;C7;C8;C9;C10;C11;C12];
td=transpose(td);
```


Appendix B: (Continued)

```
sum=C1+C2+C3+C4+C5+C6+C7+C8+C9+C10+C11+C12;
mean1=sum/12;

for i=1:mdp

m=[td(i,1);td(i,2);td(i,3);td(i,4);td(i,5);td(i,6);td(i,7);td(i,8
);td(i,9);td(i,10);td(i,11);td(i,12)];
    med = quantile(m, 0.5);
    upq = quantile(m, 0.75);
    loq = quantile(m, 0.25);
    mp=min(m);
    mpl=max(m);
    medd(i)=med;
    upqd(i)=upq;
    loqd(i)=loq;
    mind(i)=mp(1);
    maxd(i)=mpl(1);
end;
    medd = transpose(medd);
    upqd = transpose(upqd);
    lowq = transpose(loqd);
    mind=transpose(mind);
    maxd=transpose(maxd);

axes('FontSize',16);
plot(D1,mind,'--k',D1,upqd,':c',D1,medd,'-r',D1,mean1,'-g',D1,
loqd,':c',D1,maxd,'--k','LineWidth',2.5);
axis([-50 2700 0 2500]);
ylabel('Preload (lb)','FontSize',18);
xlabel('Cycles','FontSize',18);
%legend('Extent Curves','Upper and Lower
Quartiles','Median','Mean');
grid;
```

M-file *quanlocking2.m* ends here.

M-file *quanloctite2.m* begins here.

```
% Mean, median, quartiles, and extents of minimum preloads.
```

```
% "mdp" is the total number of data points to be plotted. "dpc"
% is the data point count. It controls the range of data points
% to be selected to choose a specific data point. "dpcm" is the
% data point count median. It is use to select the median cycles
```

Appendix B: (Continued)

```
% to be plotted against the lowest data point from each group.  
% "dpcm" could be change to be the end lowest cycles as well by  
% making it equal to "dpc".
```

```
clear all
```

```
mdp=128;  
dpc=8192/mdp;  
dpcm=dpc/2;
```

```
% C arrays are the minimum preloads for each range of data  
% points.  
% D arrays are the medians of cycles for each range of data  
% points.
```

```
load test5.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C1(i)=p(1);  
d=cycles(a:b);  
D1(i)=d(dpcm);  
end;
```

```
load test6.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;
```

Appendix B: (Continued)

```
        b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C2(i)=p(1);
d=cycles(a:b);
D2(i)=d(dpcm);
end;

load test12.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
D3(i)=d(dpcm);
end;

load test13.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
```

Appendix B: (Continued)

```
        a=i*dpc-dpc;  
        b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C4(i)=p(1);  
d=cycles(a:b);  
D4(i)=d(dpcm);  
end;
```

```
load test17.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C5(i)=p(1);  
d=cycles(a:b);  
D5(i)=d(dpcm);  
end;
```

```
load test19.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;
```

Appendix B: (Continued)

```
p=preload(a:b);  
p=min(p);  
C6(i)=p(1);  
d=cycles(a:b);  
D6(i)=d(dpcm);  
end;
```

```
load test20.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;
```

```
p=preload(a:b);  
p=min(p);  
C7(i)=p(1);  
d=cycles(a:b);  
D7(i)=d(dpcm);  
end;
```

```
load test22.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;
```

```
p=preload(a:b);  
p=min(p);
```

Appendix B: (Continued)

```
C8(i)=p(1);
d=cycles(a:b);
D8(i)=d(dpcm);
end;

load test23.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C9(i)=p(1);
d=cycles(a:b);
D9(i)=d(dpcm);
end;

load test29.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C10(i)=p(1);
d=cycles(a:b);
```

Appendix B: (Continued)

```
D10(i)=d(dpcm);
end;

load test31.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C11(i)=p(1);
d=cycles(a:b);
D11(i)=d(dpcm);
end;

load test34.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;
```

Appendix B: (Continued)

```
td=[C1;C2;C3;C4;C5;C6;C7;C8;C9;C10;C11;C12];
td=transpose(td);

sum=C1+C2+C3+C4+C5+C6+C7+C8+C9+C10+C11+C12;
mean1=sum/12;

for i=1:mdp

m=[td(i,1);td(i,2);td(i,3);td(i,4);td(i,5);td(i,6);td(i,7);td(i,8)
);td(i,9);td(i,10);td(i,11);td(i,12)];
    med = quantile(m, 0.5);
    upq = quantile(m, 0.75);
    loq = quantile(m, 0.25);
    mp=min(m);
    mpl=max(m);
    medd(i)=med;
    upqd(i)=upq;
    loqd(i)=loq;
    mind(i)=mp(1);
    maxd(i)=mpl(1);
end;
    medd = transpose(medd);
    upqd = transpose(upqd);
    lowq = transpose(loqd);
    mind=transpose(mind);
    maxd=transpose(maxd);

axes('FontSize',16);
plot(D1,mind,'--k',D1,upqd,':c',D1,medd,'-r',D1,mean1,'-g',D1,
loqd,':c',D1,maxd,'--k','LineWidth',2.5);
axis([-50 2700 0 2500]);
ylabel('Preload (lb)','FontSize',18);
xlabel('Cycles','FontSize',18);
%legend('Extent Curves','Upper and Lower
Quartiles','Median','Mean');
grid;
```

M-file *quanloctite2.m* ends here.

Appendix C: Normal Plots

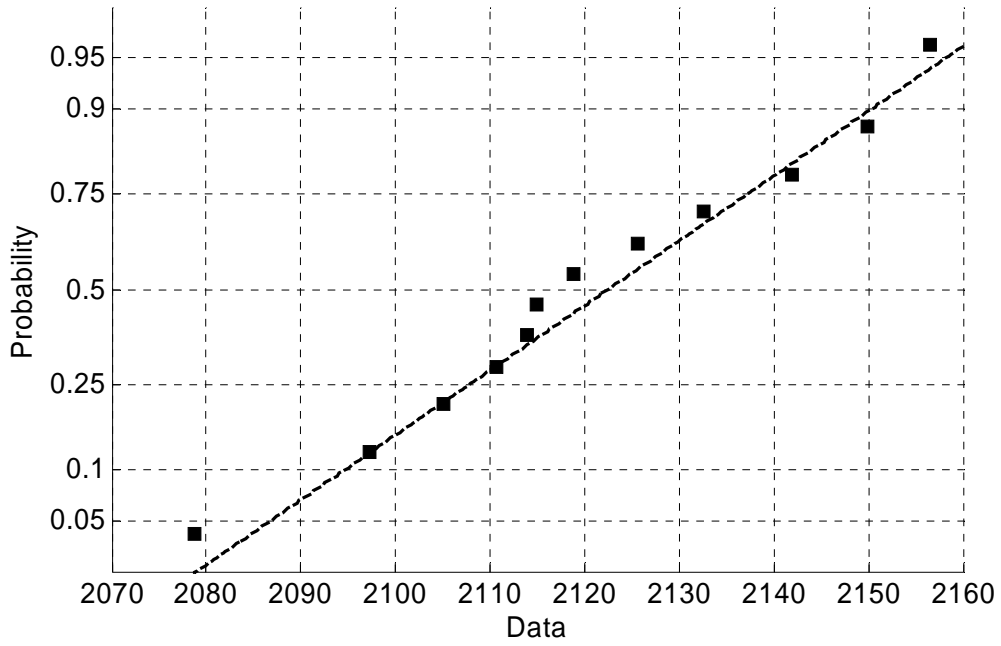


Figure C.1 Normal plot for “Standard Heli-Coil with Braycote” preloads at 10 cycles.

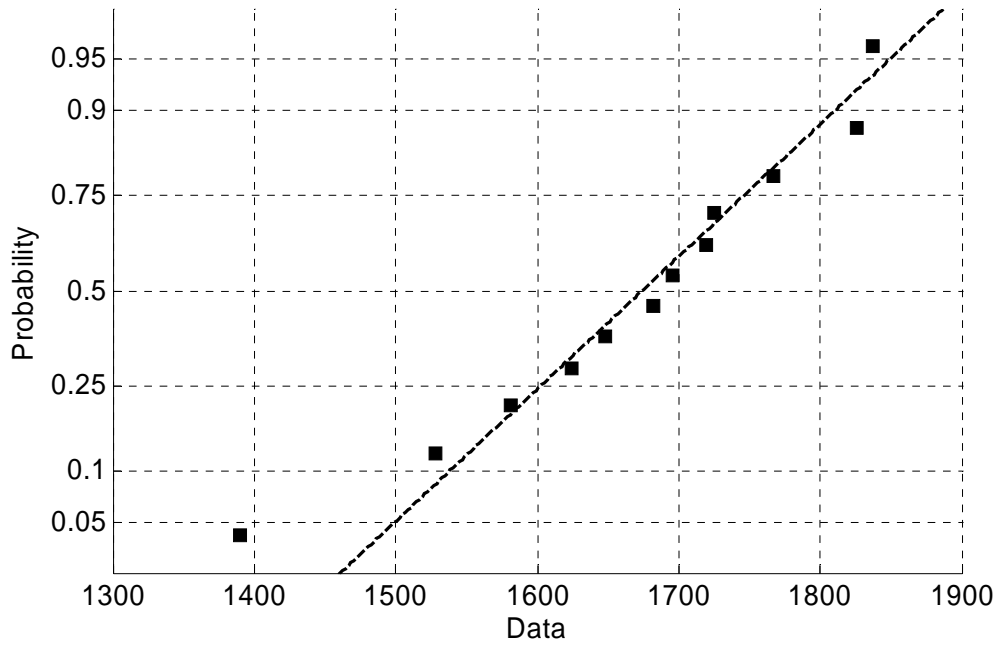


Figure C.2 Normal plot for “Standard Heli-Coil with Braycote” preloads at 250 cycles.

Appendix C: (Continued)

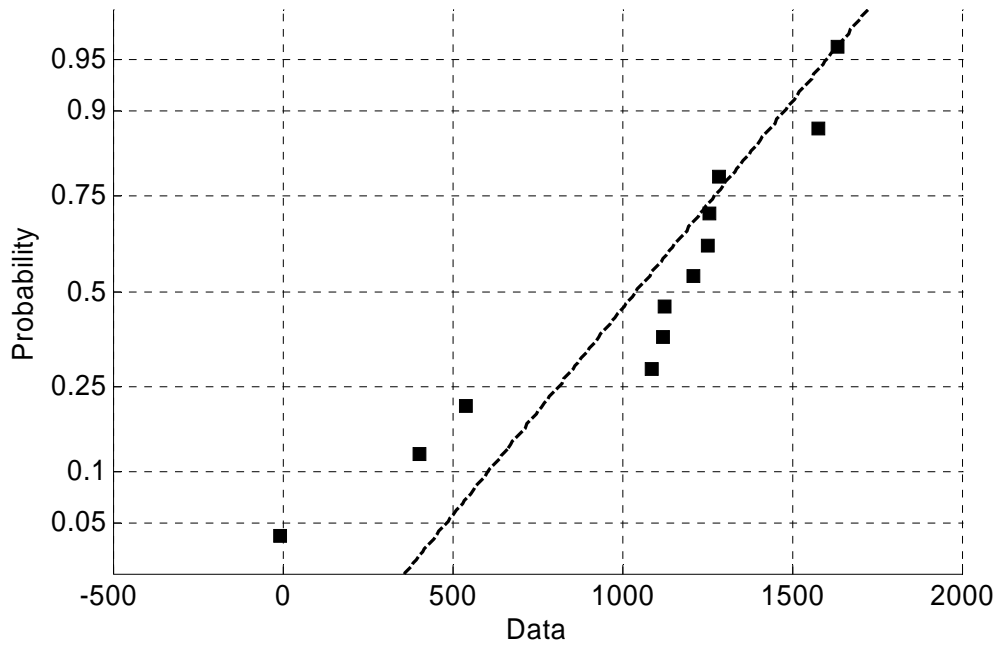


Figure C.3 Normal plot for “Standard Heli-Coil with Braycote” preloads at 500 cycles.

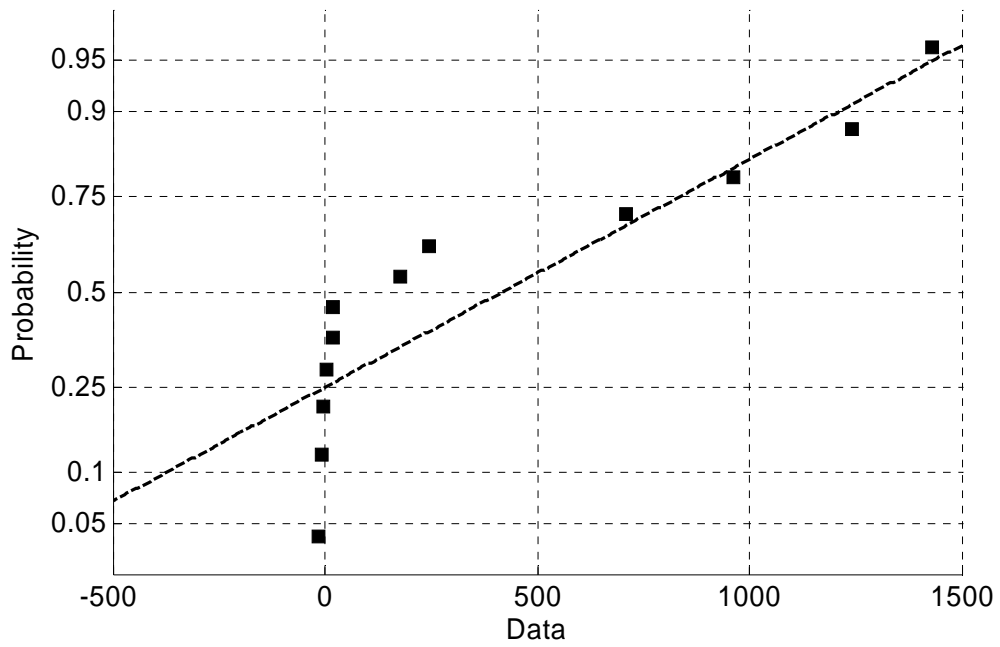


Figure C.4 Normal plot for “Standard Heli-Coil with Braycote” preloads at 750 cycles.

Appendix C: (Continued)

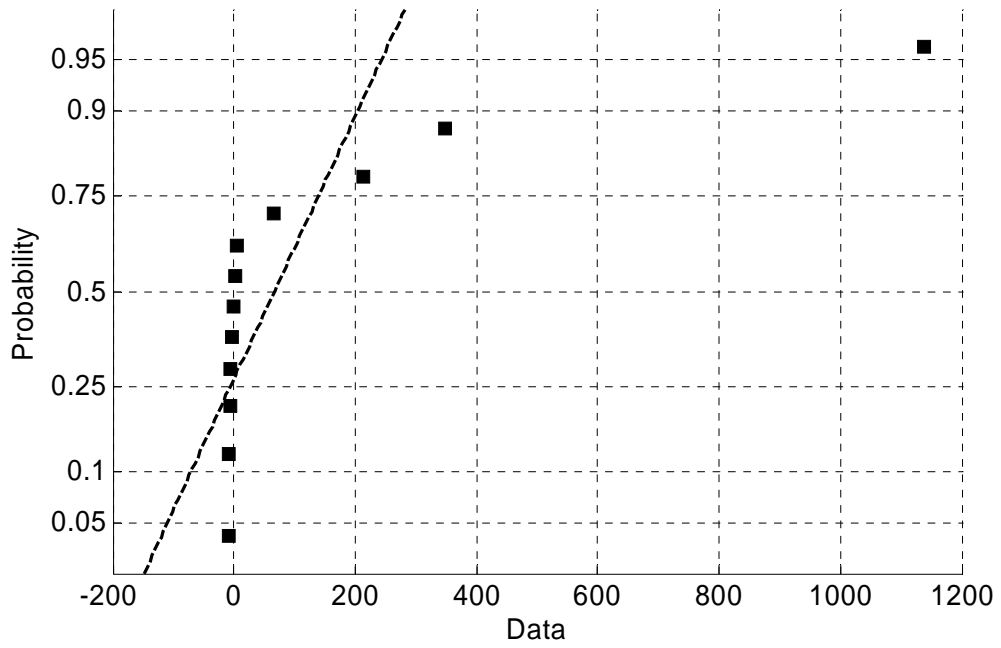


Figure C.5 Normal plot for “Standard Heli-Coil with Braycote” preloads at 1000 cycles.

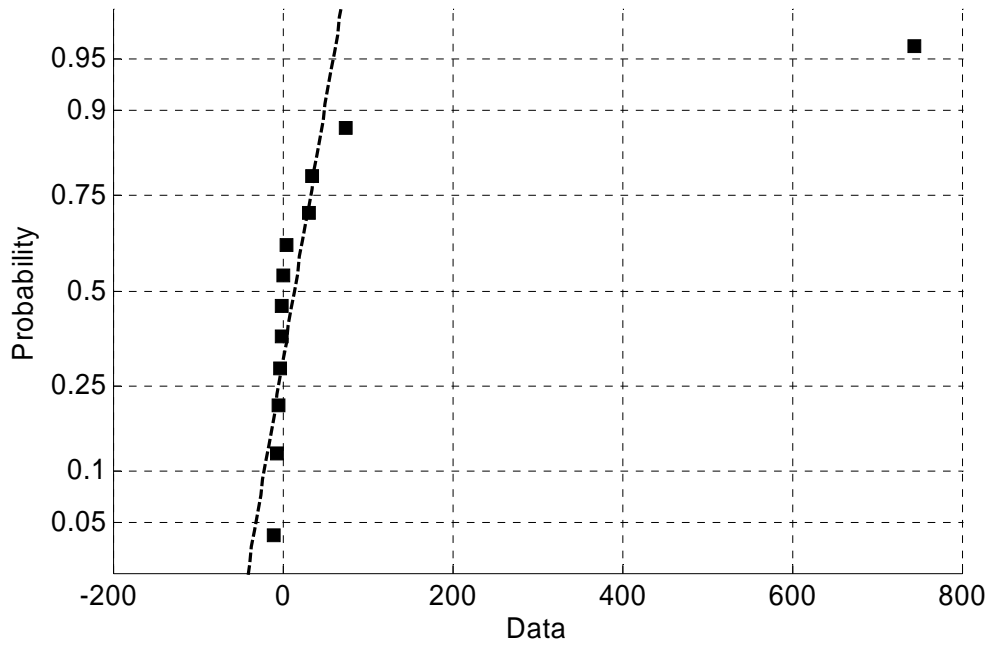


Figure C.6 Normal plot for “Standard Heli-Coil with Braycote” preloads at 1250 cycles.

Appendix C: (Continued)

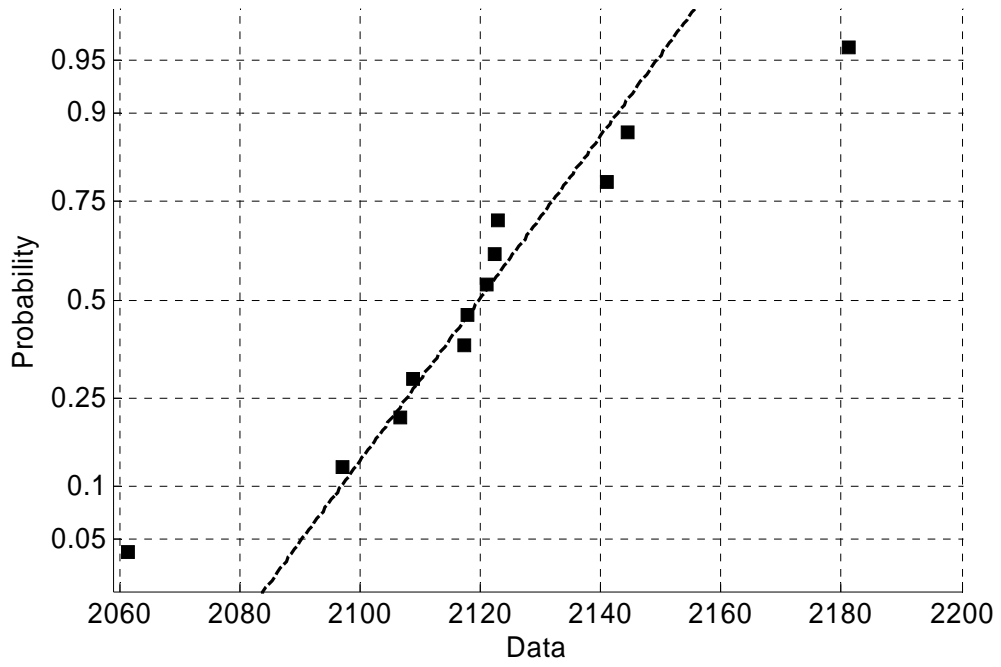


Figure C.7 Normal plot for “Locking Heli-Coil with Braycote” preloads at 10 cycles.

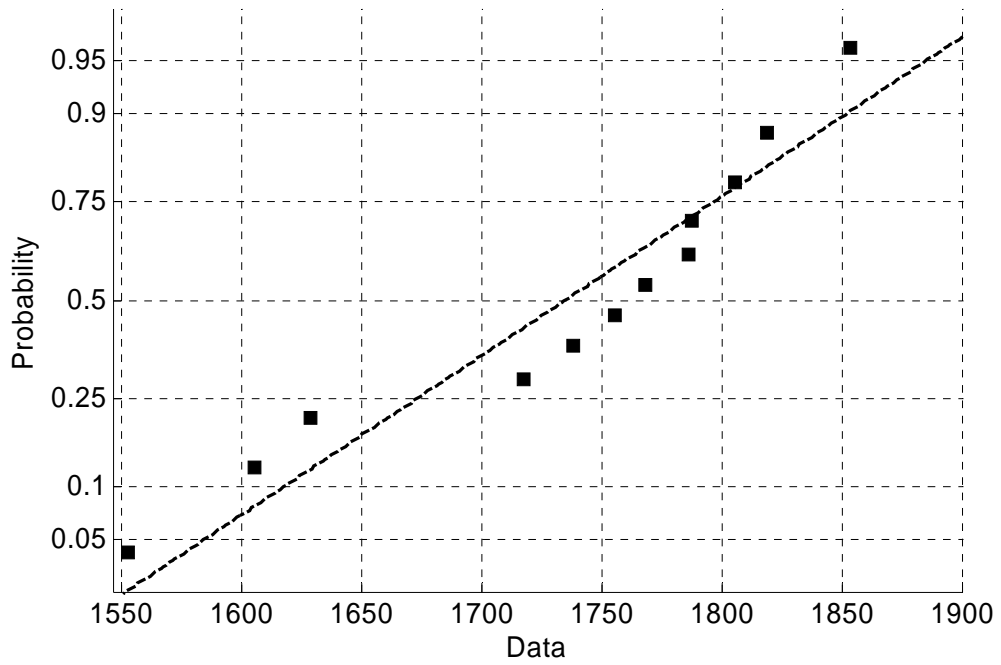


Figure C.8 Normal plot for “Locking Heli-Coil with Braycote” preloads at 250 cycles.

Appendix C: (Continued)

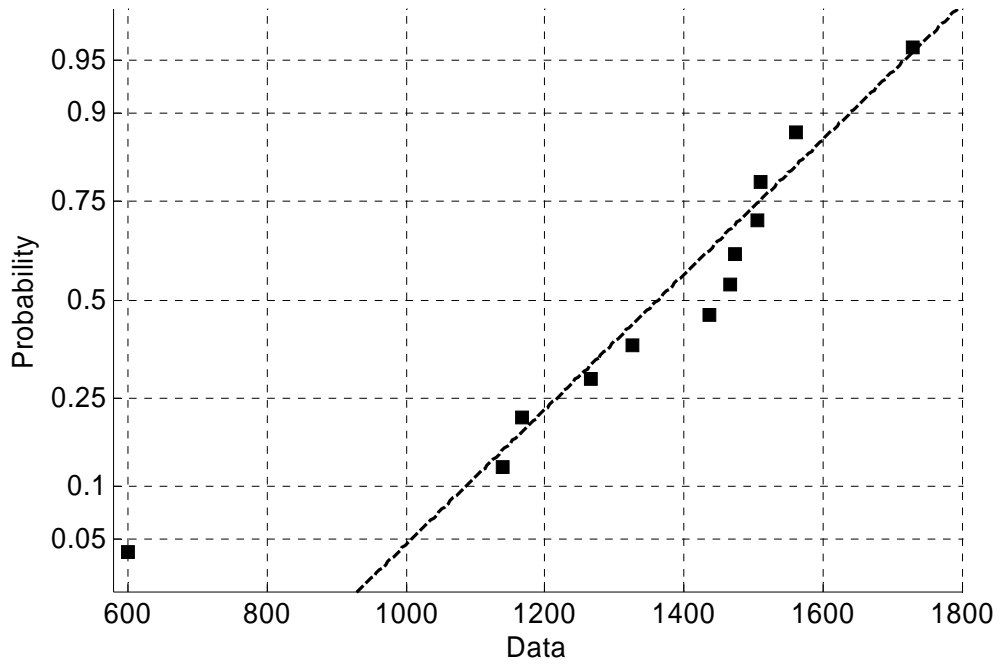


Figure C.9 Normal plot for “Locking Heli-Coil with Braycote” preloads at 500 cycles.

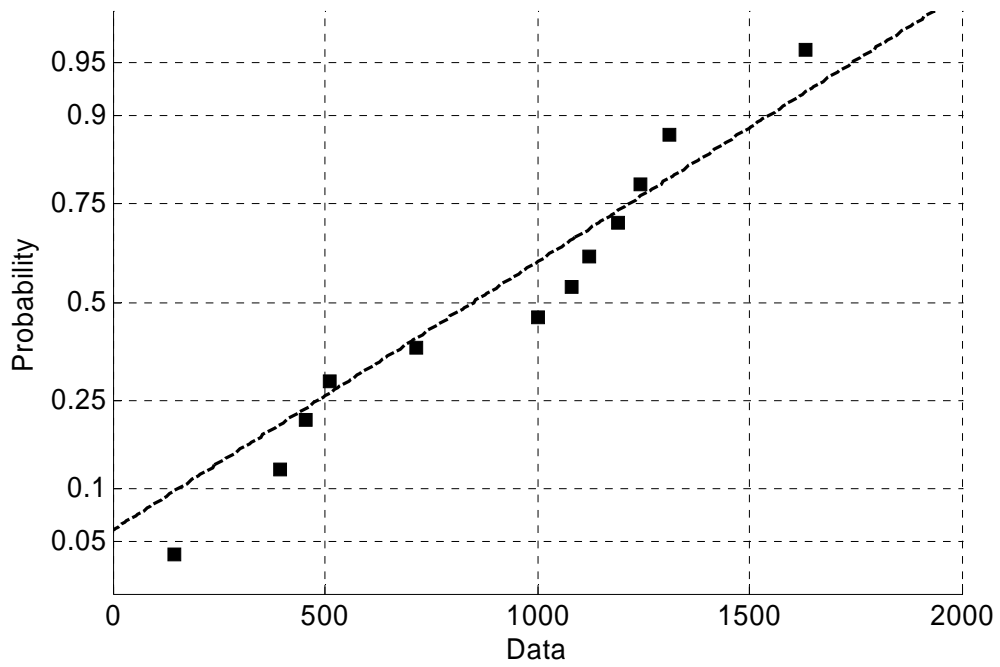


Figure C.10 Normal plot for “Locking Heli-Coil with Braycote” preloads at 750 cycles.

Appendix C: (Continued)

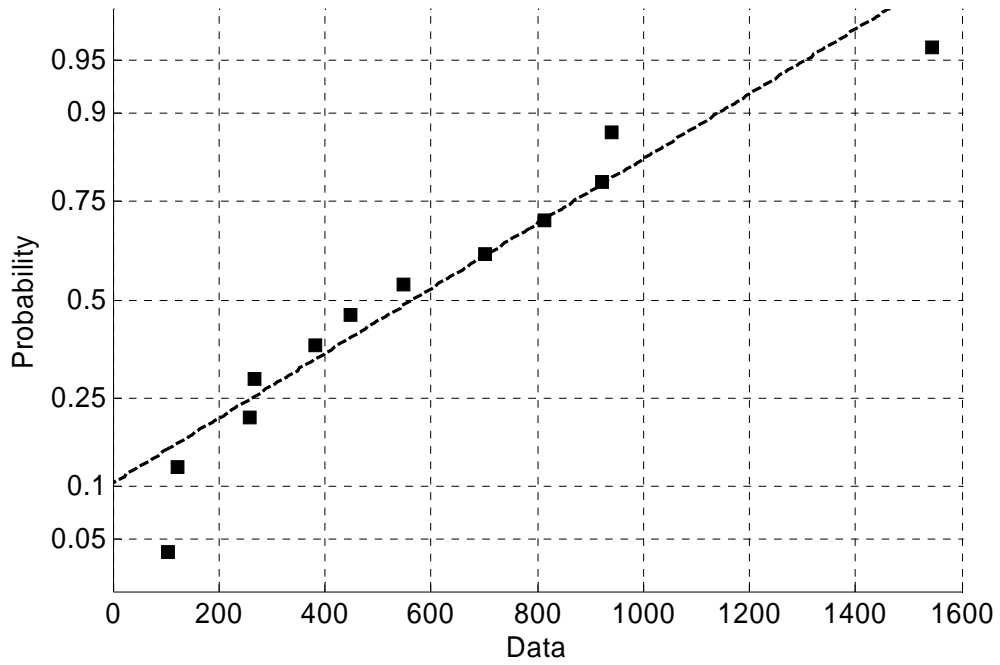


Figure C.11 Normal plot for “Locking Heli-Coil with Braycote” preloads at 1000 cycles.

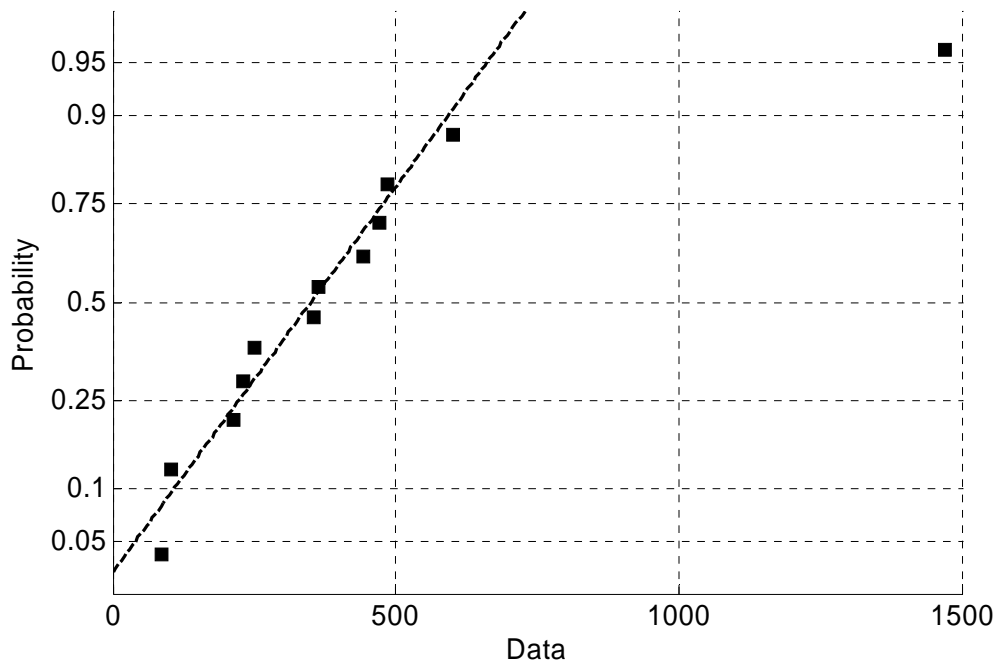


Figure C.12 Normal plot for “Locking Heli-Coil with Braycote” preloads at 1250 cycles.

Appendix C: (Continued)

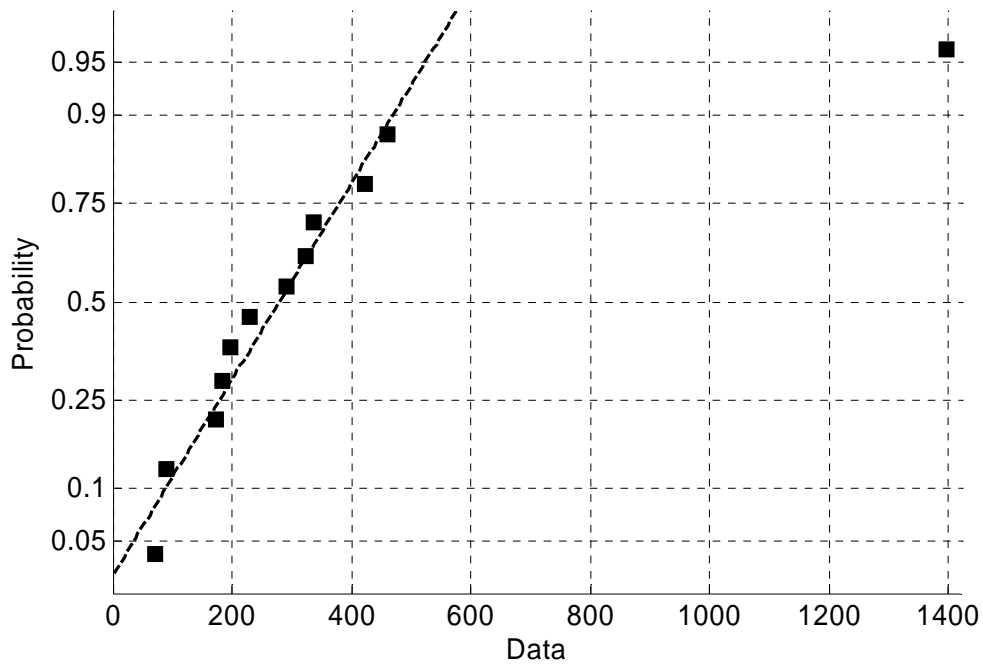


Figure C.13 Normal plot for “Locking Heli-Coil with Braycote” preloads at 1500 cycles.

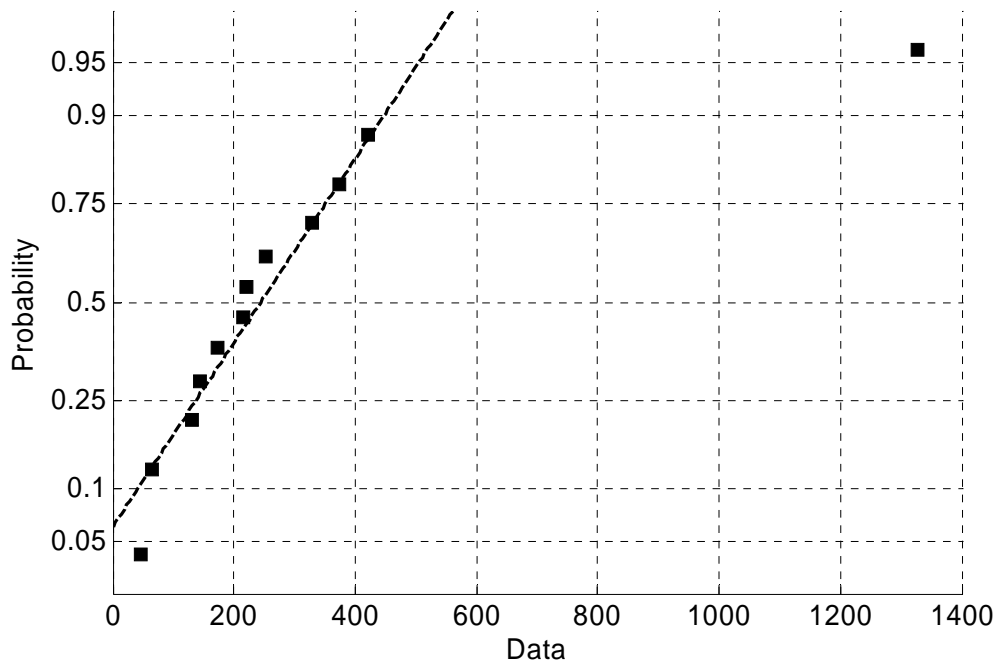


Figure C.14 Normal plot for “Locking Heli-Coil with Braycote” preloads at 1750 cycles.

Appendix C: (Continued)

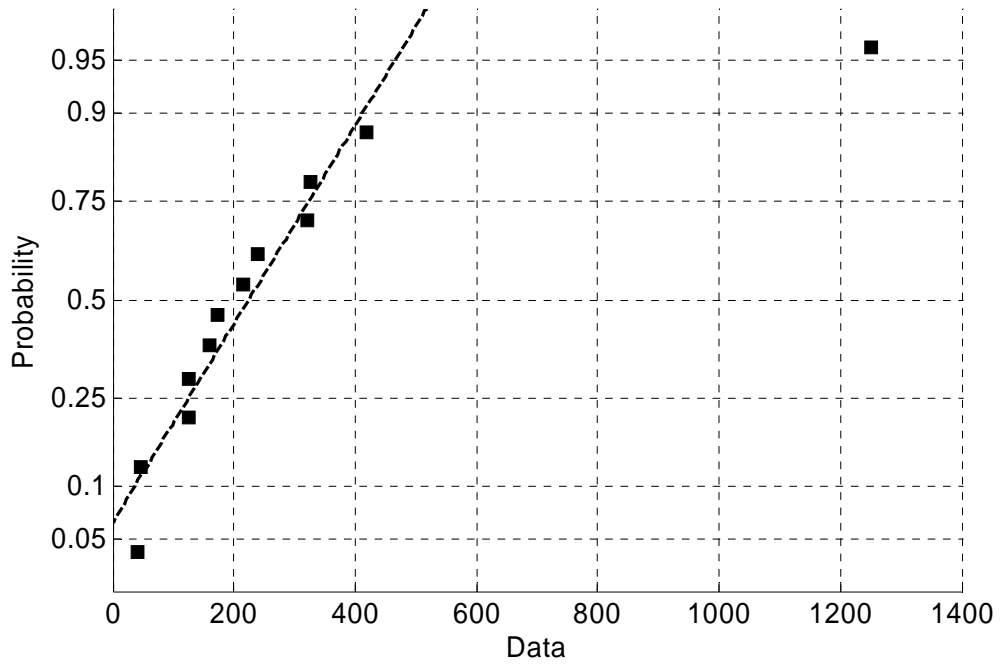


Figure C.15 Normal plot for “Locking Heli-Coil with Braycote” preloads at 2000 cycles.

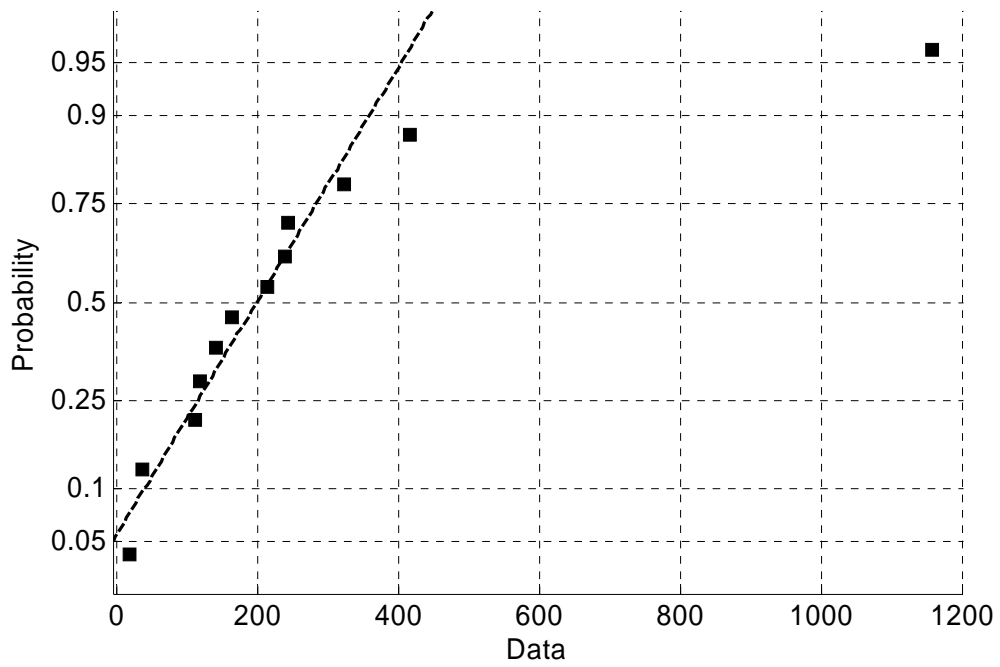


Figure C.16 Normal plot for “Locking Heli-Coil with Braycote” preloads at 2250 cycles.

Appendix C: (Continued)

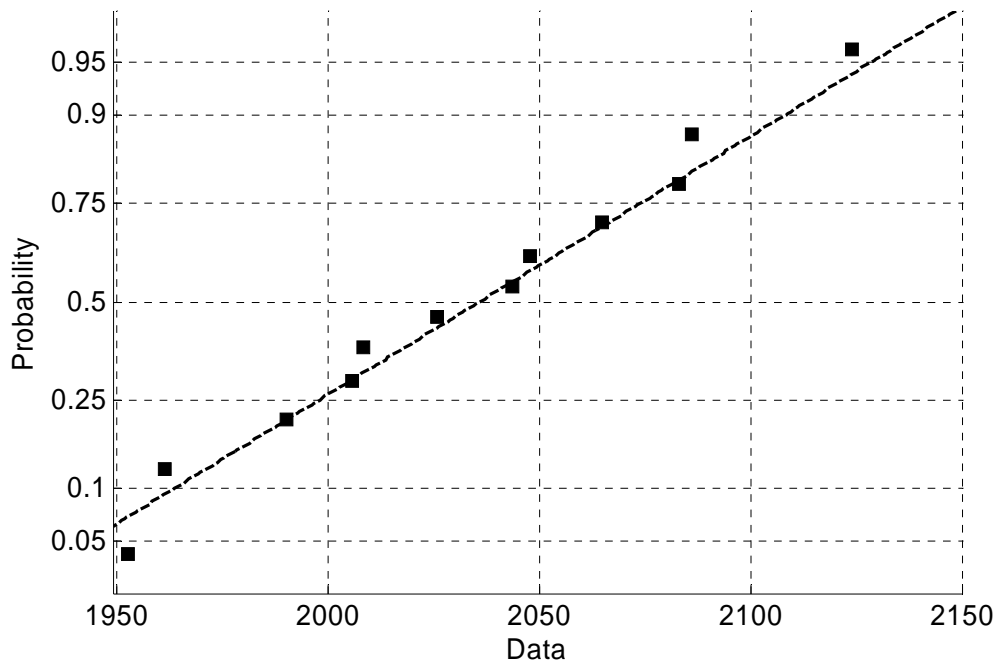


Figure C.17 Normal plot for “Standard Heli-Coil with Loctite” preloads at 10 cycles.

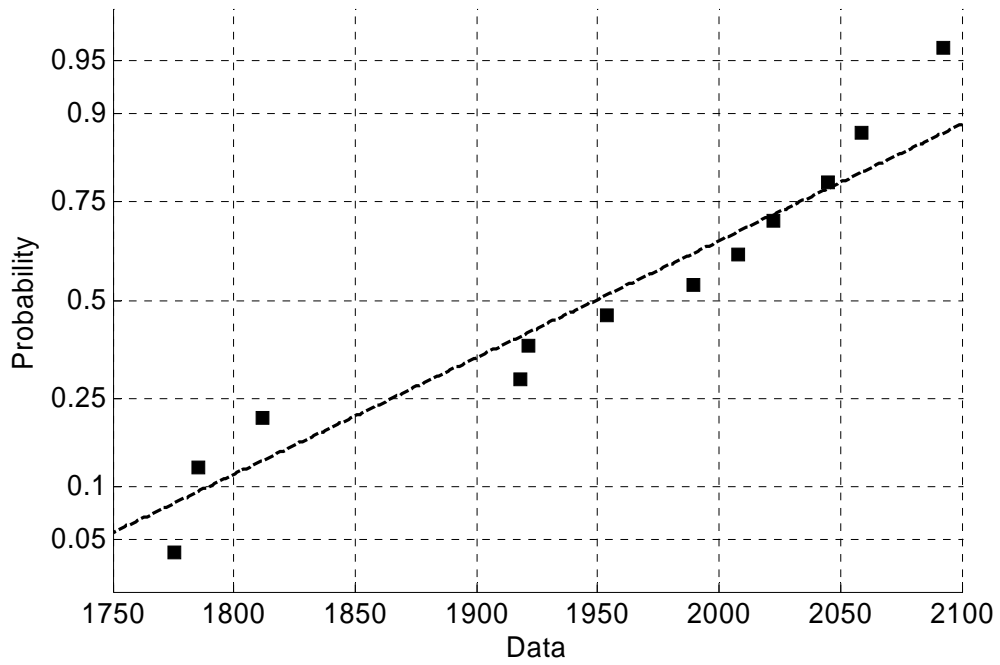


Figure C.18 Normal plot for “Standard Heli-Coil with Loctite” preloads at 250 cycles.

Appendix C: (Continued)

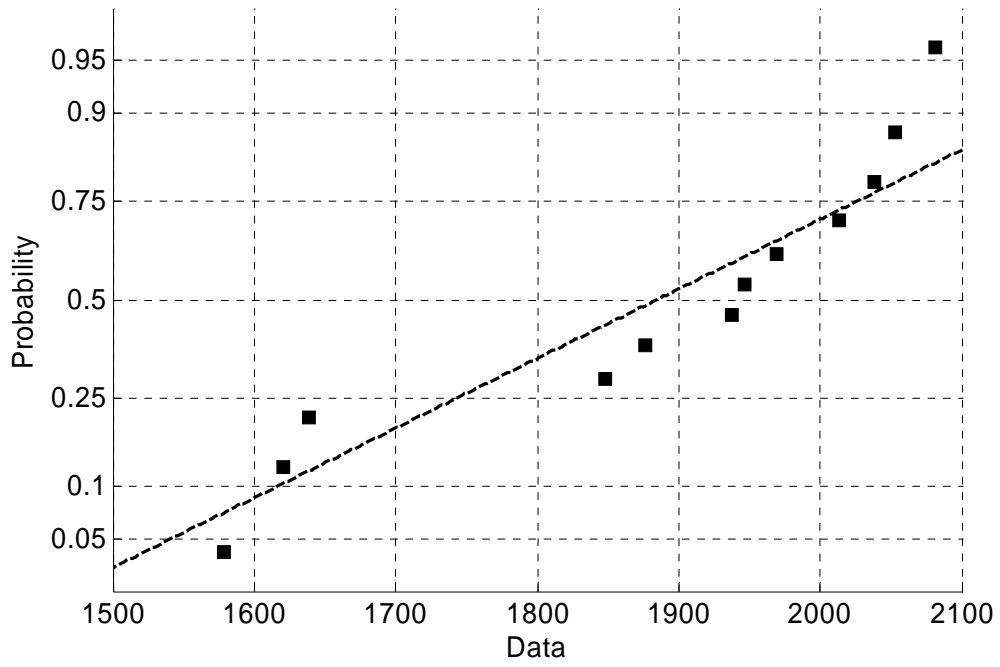


Figure C.19 Normal plot for “Standard Heli-Coil with Loctite” preloads at 500 cycles.

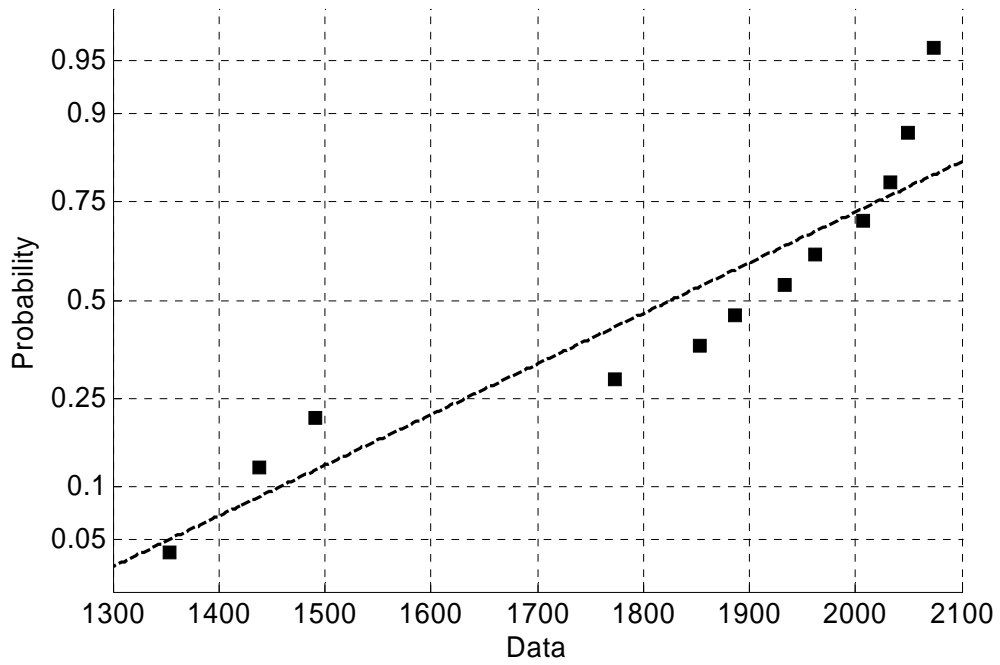


Figure C.20 Normal plot for “Standard Heli-Coil with Loctite” preloads at 750 cycles.

Appendix C: (Continued)

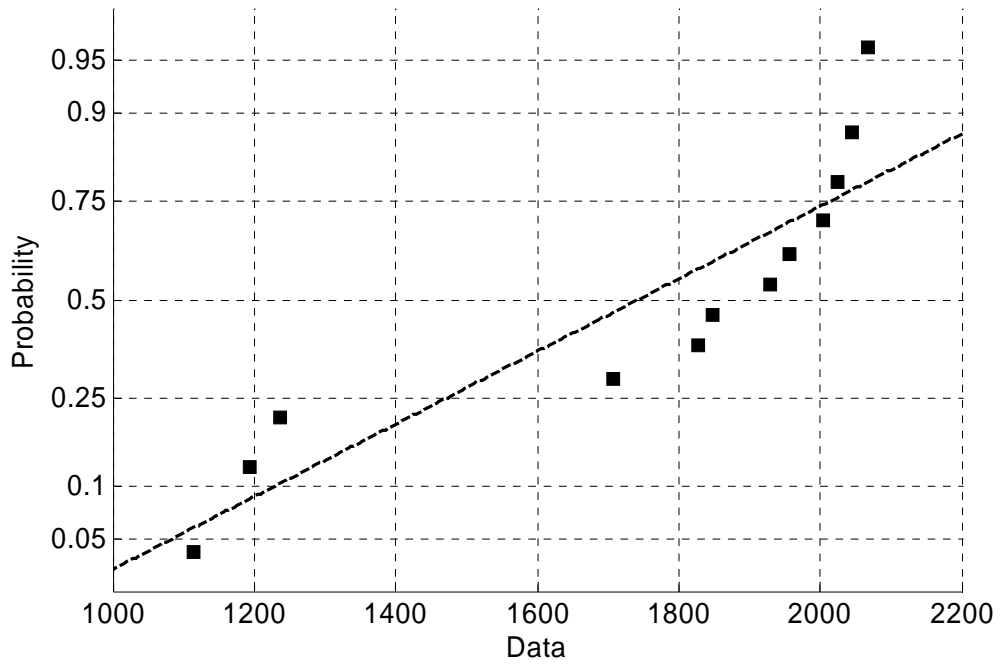


Figure C.21 Normal plot for “Standard Heli-Coil with Loctite” preloads at 1000 cycles.

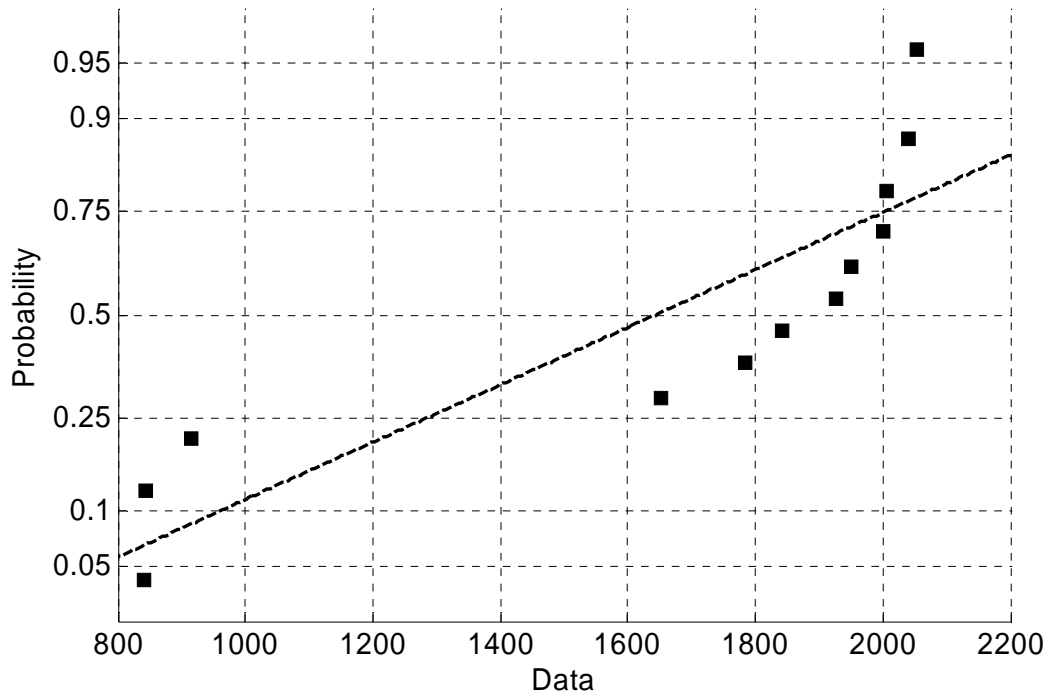


Figure C.22 Normal plot for “Standard Heli-Coil with Loctite” preloads at 1250 cycles.

Appendix C: (Continued)

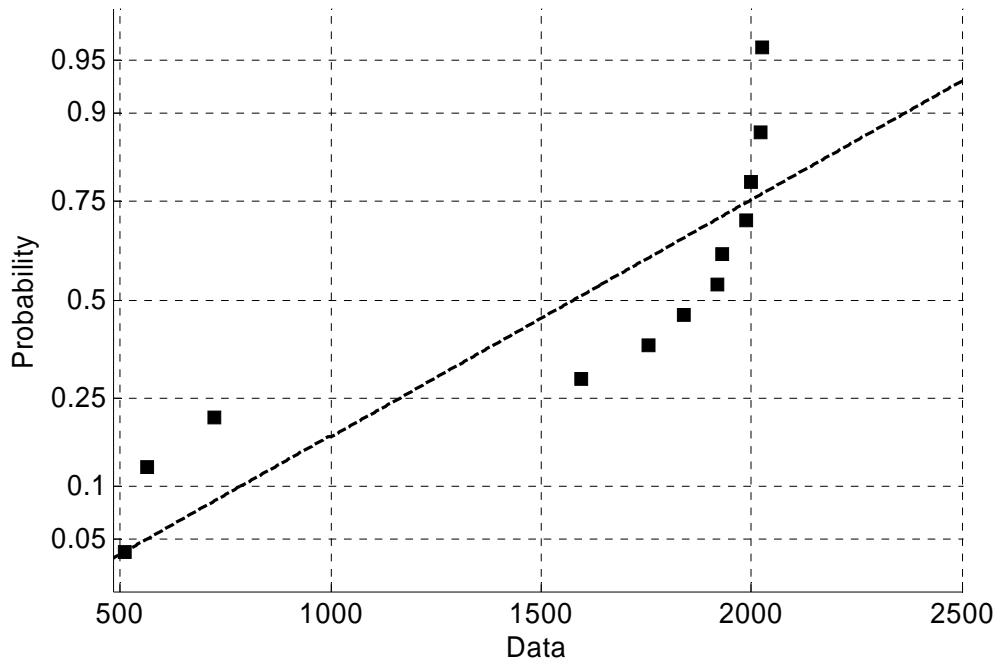


Figure C.23 Normal plot for “Standard Heli-Coil with Loctite” preloads at 1500 cycles.

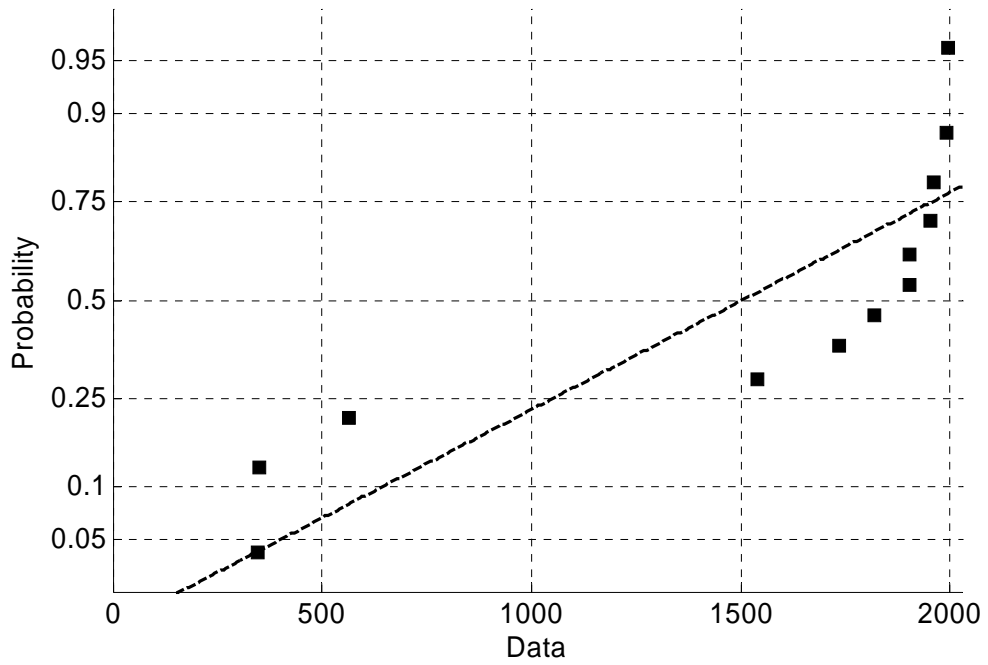


Figure C.24 Normal plot for “Standard Heli-Coil with Loctite” preloads at 1750 cycles.

Appendix C: (Continued)

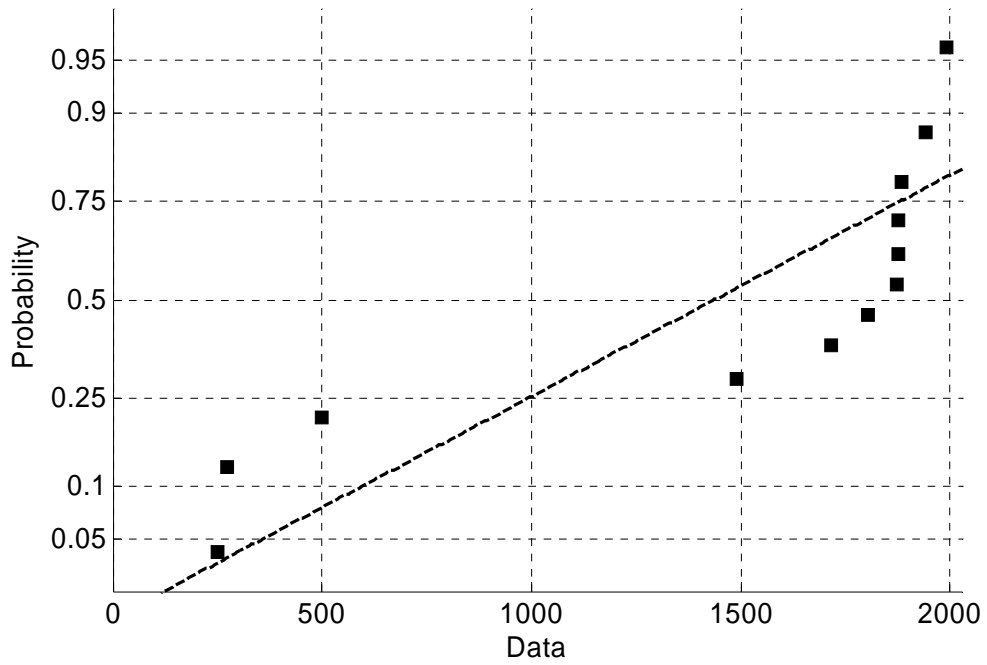


Figure C.25 Normal plot for “Standard Heli-Coil with Loctite” preloads at 2000 cycles.

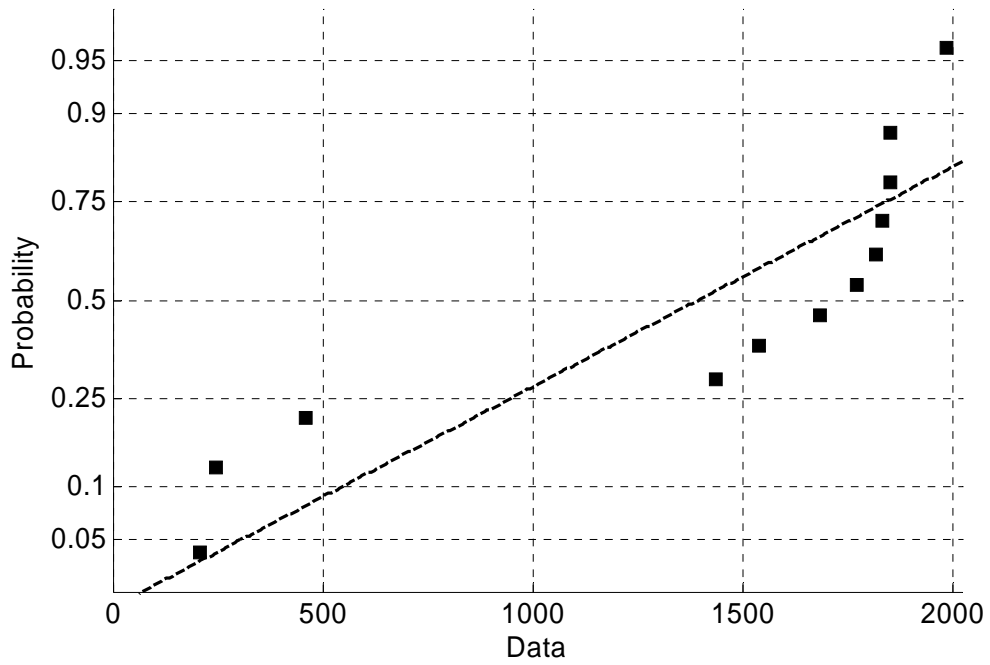


Figure C.26 Normal plot for “Standard Heli-Coil with Loctite” preloads at 2250 cycles.

Appendix D: M-files for PPCC Tests for Normality

M-files *normstd2.m*, *normlocking2.m*, and *normloctite2.m* have been written to conduct probability plot correlation coefficient tests for normality (PPCC) for minimum preloads at 10, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250 cycles. These M-files output correlation coefficients in an array for each of the three locking levels.

The data for each run is loaded, and minimum preloads for every 18.75 cycles are found using “for” loops. Every 64 data points are grouped into arrays 128 times for a total of 8192 data points for both preload and cycles. The function “min” is used to sort the preload array from least to greatest. The first element in the array is then put into another array $C_i(j)$. The cycle arrays do not need to be sorted and the median of the cycles array is assigned to the array $D_i(j)$. Then minimum preloads for all runs are put into matrix *td* in a manner in which each row is the minimum preload for all runs at a particular cycle. Rows 5, 18, 31, 44, 58, 71, 84, 98, 111, 124 of matrix *td* contain minimum preloads at 10, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250 cycles respectively. Using “for” loop these preloads are put into arrays, which are sorted by the function “sort” from least to greatest. The means of these arrays are calculated by the function “mean”. “For” loops also calculate the uniform order statistic median $p(i)$, normal order statistic median $b(i)$, and summations of the correlation coefficient. The correlation coefficient is then calculated and is put into an array.

M-file *normstd2.m* begins here.

```
% Probability Plot Correlation Coefficient Test for normality
% (PPCC) for minimum preloads of test data from Junker machine

% "mdp" is the total number of data points to be plotted. "dpc"
% is the data point count. It controls the range of data points
```

Appendix D: (Continued)

```
% to be selected to choose a specific data point. "dpcm" is the  
% data point count median. It is use to select the median cycles  
% to be plotted against the lowest data point from each group.  
% "dpcm" could be change to be the end lowest cycles as well by  
% making it equal to "dpc".
```

```
clear all
```

```
mdp=128;  
dpc=8192/mdp;  
dpcm=dpc/2;
```

```
% C arrays are the minimum preloads for each range of data  
% points.  
% D arrays are the medians of cycles for each range of data  
% points.
```

```
load test1.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C1(i)=p(1);  
d=cycles(a:b);  
D1(i)=d(dpcm);  
end;
```

```
load test3.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp
```

Appendix D: (Continued)

```
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C2(i)=p(1);
d=cycles(a:b);
D2(i)=d(dpcm);
end;

load test4.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
D3(i)=d(dpcm);
end;

load test7.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
```


Appendix D: (Continued)

```
        b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
end;

load test8.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C5(i)=p(1);
d=cycles(a:b);
D5(i)=d(dpcm);
end;

load test11.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
```

Appendix D: (Continued)

```
        a=i*dpc-dpc;  
        b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C6(i)=p(1);  
d=cycles(a:b);  
D6(i)=d(dpcm);  
end;
```

```
load test15.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C7(i)=p(1);  
d=cycles(a:b);  
D7(i)=d(dpcm);  
end;
```

```
load test21.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;
```

Appendix D: (Continued)

```
end;
p=preload(a:b);
p=min(p);
C8(i)=p(1);
d=cycles(a:b);
D8(i)=d(dpcm);
end;

load test25.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C9(i)=p(1);
d=cycles(a:b);
D9(i)=d(dpcm);
end;

load test26.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
```

Appendix D: (Continued)

```
p=min(p);
C10(i)=p(1);
d=cycles(a:b);
D10(i)=d(dpcm);
end;

load test28.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C11(i)=p(1);
d=cycles(a:b);
D11(i)=d(dpcm);
end;

load test33.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
```

Appendix D: (Continued)

```
d=cycles(a:b);
D12(i)=d(dpcm);
end;

% Minimum preloads for all runs for this locking level in a
% matrix.

td=[C1;C2;C3;C4;C5;C6;C7;C8;C9;C10;C11;C12];
td=transpose(td);

% n is the number of test runs
n=12;

% The for loop selects minimum preloads at 10, 250, 500,
% 750, 1000, 1250, 1500, 1750, 2000, 2250 cycles.

% The uniform order statistic median is calculated from the Blom
% plotting position equation p(i).

for i=1:n;
    m1(i)=td(5,i);
    m2(i)=td(18,i);
    m3(i)=td(31,i);
    m4(i)=td(44,i);
    m5(i)=td(58,i);
    m6(i)=td(71,i);
    m7(i)=td(84,i);
    m8(i)=td(98,i);
    m9(i)=td(111,i);
    m10(i)=td(124,i);
    p(i)=(i-(3/8))/(n+(1/4));
end;

% Here the minimum preloads are being sorted from least to
% greatest using "sort".

m1=sort(m1);
m2=sort(m2);
m3=sort(m3);
m4=sort(m4);
m5=sort(m5);
m6=sort(m6);
m7=sort(m7);
m8=sort(m8);
m9=sort(m9);
m10=sort(m10);
```

Appendix D: (Continued)

```
% The mean of the minimum preloads are found using "mean"

mm1=mean(m1);
mm2=mean(m2);
mm3=mean(m3);
mm4=mean(m4);
mm5=mean(m5);
mm6=mean(m6);
mm7=mean(m7);
mm8=mean(m8);
mm9=mean(m9);
mm10=mean(m10);

% The normal order statistic median b(i)

for i=1:n;

b(i)=(2.515517+.802853*(log(1/p(i)^2))^+.5+.010328*((log(1/p(i)^2)
)^.5)^2)/(1+1.432788*(log(1/p(i)^2))^+.5+.189269*((log(1/p(i)^2)
)^.5)^2+.0013088*(log(1/p(i)^2)^.5)^3)-
(log(1/p(i)^2))^0.5+0.00045;
end;

bm=mean(b);

h1=0;h2=0;h3=0;h4=0;h5=0;h6=0;h7=0;h8=0;h9=0;h10=0;
hb1=0;hb2=0;hb3=0;hb4=0;hb5=0;hb6=0;hb7=0;hb8=0;hb9=0;hb10=0;
b2=0;

%Correlation Coefficient sum's

for i=1:n;
h1=h1+(m1(i)-mm1)^2;
h2=h2+(m2(i)-mm2)^2;
h3=h3+(m3(i)-mm3)^2;
h4=h4+(m4(i)-mm4)^2;
h5=h5+(m5(i)-mm5)^2;
h6=h6+(m6(i)-mm6)^2;
h7=h7+(m7(i)-mm7)^2;
h8=h8+(m8(i)-mm8)^2;
h9=h9+(m9(i)-mm9)^2;
h10=h10+(m10(i)-mm10)^2;

hb1=hb1+(m1(i)-mm1)*(b(i)-bm);
hb2=hb2+(m2(i)-mm2)*(b(i)-bm);
hb3=hb3+(m3(i)-mm3)*(b(i)-bm);
```

Appendix D: (Continued)

```
hb4=hb4+(m4(i)-mm4)*(b(i)-bm);
hb5=hb5+(m5(i)-mm5)*(b(i)-bm);
hb6=hb6+(m6(i)-mm6)*(b(i)-bm);
hb7=hb7+(m7(i)-mm7)*(b(i)-bm);
hb8=hb8+(m8(i)-mm8)*(b(i)-bm);
hb9=hb9+(m9(i)-mm9)*(b(i)-bm);
hb10=hb10+(m10(i)-mm10)*(b(i)-bm);
b2=b2+(b(i)-bm)^2;
end;

% Correlation Coefficient is calculated.

Rp1=hb1/((h1*b2)^0.5);
Rp2=hb2/((h2*b2)^0.5);
Rp3=hb3/((h3*b2)^0.5);
Rp4=hb4/((h4*b2)^0.5);
Rp5=hb5/((h5*b2)^0.5);
Rp6=hb6/((h6*b2)^0.5);
Rp7=hb7/((h7*b2)^0.5);
Rp8=hb8/((h8*b2)^0.5);
Rp9=hb9/((h9*b2)^0.5);
Rp10=hb10/((h10*b2)^0.5);

%Correlation Coefficient the output as an array.

Rp=[Rp1,Rp2,Rp3,Rp4,Rp5,Rp6,Rp7,Rp8,Rp9,Rp10]

M-file normstd2.m ends here.

M-file normlocking2.m begins here.

% Probability Plot Correlation Coefficient Test for normality
% (PPCC) for minimum preloads of test data from Junker machine

% "mdp" is the total number of data points to be plotted. "dpc"
% is the data point count. It controls the range of data points
% to be selected to choose a specific data point. "dpcm" is the
% data point count median. It is use to select the median cycles
% to be plotted against the lowest data point from each group.
% "dpcm" could be change to be the end lowest cycles as well by
% making it equal to "dpc".

clear all

mdp=128;
dpc=8192/mdp;
dpcm=dpc/2;
```

Appendix D: (Continued)

```
.% C arrays are the minimum preloads for each range of data  
% points.  
% D arrays are the medians of cycles for each range of data  
% points.
```

```
load test2.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C1(i)=p(1);  
d=cycles(a:b);  
D1(i)=d(dpcm);  
end;
```

```
load test9.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C2(i)=p(1);  
d=cycles(a:b);  
D2(i)=d(dpcm);  
end;
```


Appendix D: (Continued)

```
load test10.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
D3(i)=d(dpcm);
end;
```

```
load test14.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;
```

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
end;
```

```
load test16.vna -mat;

time = SLM.tdxvec;
```

Appendix D: (Continued)

```
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C5(i)=p(1);  
d=cycles(a:b);  
D5(i)=d(dpcm);  
end;
```

```
load test18.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C6(i)=p(1);  
d=cycles(a:b);  
D6(i)=d(dpcm);  
end;
```

```
load test24.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

Appendix D: (Continued)

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C7(i)=p(1);
d=cycles(a:b);
D7(i)=d(dpcm);
end;
```

```
load test27.vna -mat;
```

```
time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;
```

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C8(i)=p(1);
d=cycles(a:b);
D8(i)=d(dpcm);
end;
```

```
load test30.vna -mat;
```

```
time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;
```

Appendix D: (Continued)

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C9(i)=p(1);
d=cycles(a:b);
D9(i)=d(dpcm);
end;

load test32.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C10(i)=p(1);
d=cycles(a:b);
D10(i)=d(dpcm);
end;

load test35.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;
```

Appendix D: (Continued)

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C11(i)=p(1);
d=cycles(a:b);
D11(i)=d(dpcm);
end;

load test36.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;

% Minimum preloads for all runs for this locking level in a
% matrix.

% C12 is test 36 or run 20 which is a larger outlier, here it is
% taken out an n is changed to 11

td=[C1;C2;C3;C4;C5;C6;C7;C8;C9;C10;C11];
td=transpose(td);
```

Appendix D: (Continued)

```
% n is the number of test run
n=11;

% The for loop selects minimum preloads at 10, 250, 500,
% 750, 1000, 1250, 1500, 1750, 2000, 2250 cycles.

% The uniform order statistic median is calculated from the Blom
% plotting position equation p(i).

for i=1:n;
    m1(i)=td(5,i);
    m2(i)=td(18,i);
    m3(i)=td(31,i);
    m4(i)=td(44,i);
    m5(i)=td(58,i);
    m6(i)=td(71,i);
    m7(i)=td(84,i);
    m8(i)=td(98,i);
    m9(i)=td(111,i);
    m10(i)=td(124,i);
    p(i)=(i-(3/8))/(n+(1/4));
end;

% Here the minimum preloads are being sorted from least to
% greatest using "sort".

m1=sort(m1);
m2=sort(m2);
m3=sort(m3);
m4=sort(m4);
m5=sort(m5);
m6=sort(m6);
m7=sort(m7);
m8=sort(m8);
m9=sort(m9);
m10=sort(m10);

% The mean of the minimum preloads are found using "mean"

mm1=mean(m1);
mm2=mean(m2);
mm3=mean(m3);
mm4=mean(m4);
mm5=mean(m5);
mm6=mean(m6);
mm7=mean(m7);
mm8=mean(m8);
```

Appendix D: (Continued)

```
mm9=mean(m9);
mm10=mean(m10);

% The normal order statistic median b(i)

for i=1:n;

b(i)=(2.515517+.802853*(log(1/p(i)^2))^5+.010328*((log(1/p(i)^2)
)^5)^2)/(1+1.432788*(log(1/p(i)^2))^5+0.189269*((log(1/p(i)^2)
)^5)^2+.0013088*(log(1/p(i)^2)^5)^3)-
(log(1/p(i)^2))^0.5+0.00045;
end;

bm=mean(b);

h1=0;h2=0;h3=0;h4=0;h5=0;h6=0;h7=0;h8=0;h9=0;h10=0;
hb1=0;hb2=0;hb3=0;hb4=0;hb5=0;hb6=0;hb7=0;hb8=0;hb9=0;hb10=0;
b2=0;

%Correlation Coefficient sum's

for i=1:n;
h1=h1+(m1(i)-mm1)^2;
h2=h2+(m2(i)-mm2)^2;
h3=h3+(m3(i)-mm3)^2;
h4=h4+(m4(i)-mm4)^2;
h5=h5+(m5(i)-mm5)^2;
h6=h6+(m6(i)-mm6)^2;
h7=h7+(m7(i)-mm7)^2;
h8=h8+(m8(i)-mm8)^2;
h9=h9+(m9(i)-mm9)^2;
h10=h10+(m10(i)-mm10)^2;
hb1=hb1+(m1(i)-mm1)*(b(i)-bm);
hb2=hb2+(m2(i)-mm2)*(b(i)-bm);
hb3=hb3+(m3(i)-mm3)*(b(i)-bm);
hb4=hb4+(m4(i)-mm4)*(b(i)-bm);
hb5=hb5+(m5(i)-mm5)*(b(i)-bm);
hb6=hb6+(m6(i)-mm6)*(b(i)-bm);
hb7=hb7+(m7(i)-mm7)*(b(i)-bm);
hb8=hb8+(m8(i)-mm8)*(b(i)-bm);
hb9=hb9+(m9(i)-mm9)*(b(i)-bm);
hb10=hb10+(m10(i)-mm10)*(b(i)-bm);
b2=b2+(b(i)-bm)^2;
end;
```

Appendix D: (Continued)

```
% Correlation Coefficient is calculated.
Rp1=hb1/((h1*b2)^0.5);
Rp2=hb2/((h2*b2)^0.5);
Rp3=hb3/((h3*b2)^0.5);
Rp4=hb4/((h4*b2)^0.5);
Rp5=hb5/((h5*b2)^0.5);
Rp6=hb6/((h6*b2)^0.5);
Rp7=hb7/((h7*b2)^0.5);
Rp8=hb8/((h8*b2)^0.5);
Rp9=hb9/((h9*b2)^0.5);
Rp10=hb10/((h10*b2)^0.5);

%Correlation Coefficient the output as an array.
Rp=[Rp1,Rp2,Rp3,Rp4,Rp5,Rp6,Rp7,Rp8,Rp9,Rp10]
```

M-File *normlocking2.m* ends here.

M-File *normloctite2.m* begins here.

```
% Probability Plot Correlation Coefficient Test for normality
% (PPCC) for minimum preloads of test data from Junker machine

% "mdp" is the total number of data points to be plotted. "dpc"
% is the data point count. It controls the range of data points
% to be selected to choose a specific data point. "dpcm" is the
% data point count median. It is use to select the median cycles
% to be plotted against the lowest data point from each group.
% "dpcm" could be change to be the end lowest cycles as well by
% making it equal to "dpc".

clear all

mdp=128;
dpc=8192/mdp;
dpcm=dpc/2;

% C arrays are the minimum preloads for each range of data
% points.
% D arrays are the medians of cycles for each range of data
% points.

load test5.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;
```


Appendix D: (Continued)

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C1(i)=p(1);
d=cycles(a:b);
D1(i)=d(dpcm);
end;

load test6.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C2(i)=p(1);
d=cycles(a:b);
D2(i)=d(dpcm);
end;

load test12.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
```

Appendix D: (Continued)

```
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
D3(i)=d(dpcm);
end;

load test13.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
end;

load test17.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
```

Appendix D: (Continued)

```
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C5(i)=p(1);
d=cycles(a:b);
D5(i)=d(dpcm);
end;

load test19.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C6(i)=p(1);
d=cycles(a:b);
D6(i)=d(dpcm);
end;

load test20.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
```

Appendix D: (Continued)

```
end;
p=preload(a:b);
p=min(p);
C7(i)=p(1);
d=cycles(a:b);
D7(i)=d(dpcm);
end;

load test22.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C8(i)=p(1);
d=cycles(a:b);
D8(i)=d(dpcm);
end;

load test23.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
```

Appendix D: (Continued)

```
p=min(p);
C9(i)=p(1);
d=cycles(a:b);
D9(i)=d(dpcm);
end;

load test29.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C10(i)=p(1);
d=cycles(a:b);
D10(i)=d(dpcm);
end;

load test31.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C11(i)=p(1);
d=cycles(a:b);
```

Appendix D: (Continued)

```
D11(i)=d(dpcm);
end;

load test34.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;

% Minimum preloads for all runs for this locking level in a
% matrix.

td=[C1;C2;C3;C4;C5;C6;C7;C8;C9;C10;C11;C12];
td=transpose(td);

% n is the number of test run
n=12;

% The for loop selects minimum preloads at 10, 250, 500,
% 750, 1000, 1250, 1500, 1750, 2000, 2250 cycles.

% The uniform order statistic median is calculated from the Blom
% plotting position equation p(i).

for i=1:n;
    m1(i)=td(5,i);
    m2(i)=td(18,i);
    m3(i)=td(31,i);
    m4(i)=td(44,i);
    m5(i)=td(58,i);
    m6(i)=td(71,i);
```

Appendix D: (Continued)

```
m7(i)=td(84,i);
m8(i)=td(98,i);
m9(i)=td(111,i);
m10(i)=td(124,i);
p(i)=(i-(3/8))/(n+(1/4));
end;

% Here the minimum preloads are being sorted from least to
% leatest using "sort".

m1=sort(m1);
m2=sort(m2);
m3=sort(m3);
m4=sort(m4);
m5=sort(m5);
m6=sort(m6);
m7=sort(m7);
m8=sort(m8);
m9=sort(m9);
m10=sort(m10);

% The mean of the minimum preloads are found using "mean"

mm1=mean(m1);
mm2=mean(m2);
mm3=mean(m3);
mm4=mean(m4);
mm5=mean(m5);
mm6=mean(m6);
mm7=mean(m7);
mm8=mean(m8);
mm9=mean(m9);
mm10=mean(m10);

% The normal order statistic median b(i)

for i=1:n;

b(i)=(2.515517+.802853*(log(1/p(i)^2)).^5+.010328*((log(1/p(i)^2)
)^.5)^2)/(1+1.432788*(log(1/p(i)^2)).^5+0.189269*((log(1/p(i)^2)
)^.5)^2+.0013088*(log(1/p(i)^2)).^5)^3-
(log(1/p(i)^2)).^0.5+0.00045;
end;

bm=mean(b);

h1=0;h2=0;h3=0;h4=0;h5=0;h6=0;h7=0;h8=0;h9=0;h10=0;
```

Appendix D: (Continued)

```
hb1=0;hb2=0;hb3=0;hb4=0;hb5=0;hb6=0;hb7=0;hb8=0;hb9=0;hb10=0;
b2=0;
```

```
%Correlation Coefficient sum's
```

```
for i=1:n;
    h1=h1+(m1(i)-mm1)^2;
    h2=h2+(m2(i)-mm2)^2;
    h3=h3+(m3(i)-mm3)^2;
    h4=h4+(m4(i)-mm4)^2;
    h5=h5+(m5(i)-mm5)^2;
    h6=h6+(m6(i)-mm6)^2;
    h7=h7+(m7(i)-mm7)^2;
    h8=h8+(m8(i)-mm8)^2;
    h9=h9+(m9(i)-mm9)^2;
    h10=h10+(m10(i)-mm10)^2;
    hb1=hb1+(m1(i)-mm1)*(b(i)-bm);
    hb2=hb2+(m2(i)-mm2)*(b(i)-bm);
    hb3=hb3+(m3(i)-mm3)*(b(i)-bm);
    hb4=hb4+(m4(i)-mm4)*(b(i)-bm);
    hb5=hb5+(m5(i)-mm5)*(b(i)-bm);
    hb6=hb6+(m6(i)-mm6)*(b(i)-bm);
    hb7=hb7+(m7(i)-mm7)*(b(i)-bm);
    hb8=hb8+(m8(i)-mm8)*(b(i)-bm);
    hb9=hb9+(m9(i)-mm9)*(b(i)-bm);
    hb10=hb10+(m10(i)-mm10)*(b(i)-bm);
    b2=b2+(b(i)-bm)^2;
end;
```

```
% Correlation Coefficient is calculated.
```

```
Rp1=hb1/((h1*b2)^0.5);
Rp2=hb2/((h2*b2)^0.5);
Rp3=hb3/((h3*b2)^0.5);
Rp4=hb4/((h4*b2)^0.5);
Rp5=hb5/((h5*b2)^0.5);
Rp6=hb6/((h6*b2)^0.5);
Rp7=hb7/((h7*b2)^0.5);
Rp8=hb8/((h8*b2)^0.5);
Rp9=hb9/((h9*b2)^0.5);
Rp10=hb10/((h10*b2)^0.5);
```

```
%Correlation Coefficient the output as an array.
```

```
Rp=[Rp1,Rp2,Rp3,Rp4,Rp5,Rp6,Rp7,Rp8,Rp9,Rp10]
```

M-file *normloctite2.m* ends here.

Appendix E: M-files for Confidence Intervals

M-files *CI95std2.m*, *CI95locking2.m*, and *CI95loctite2.m* create plots that show 95% confidence intervals for the population means of minimum preloads at 10, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250 cycles. M-files *CI99std2.m*, *CI99locking2.m*, and *CI99loctite2.m* create plots that show 99% confidence intervals for the population means for the same minimum preloads as the 95% confidence intervals. All M-files plot the confidence intervals with the sample mean by preload versus cycles.

The data for each run is loaded, and minimum preloads for every 18.75 cycles are found using for loops. Every 64 data points are grouped into arrays 128 times for a total of 8192 data points for both preload and cycles. The function “min” is used to sort the preload array from least to greatest. The first element in the array is then put into another array $C_i(j)$. The cycle arrays do not need to be sorted and the median of the cycle arrays are assigned to the array $D_i(j)$. Then minimum preloads for all runs are put into matrix td in a manner in which each row is the minimum preload for all runs at a particular cycle. Rows 5, 18, 31, 44, 58, 71, 84, 98, 111, 124 of matrix td contain minimum preloads at 10, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250 cycles respectively. These preloads were put into arrays for each row selected. The sample means and standard deviation were calculated using the MatLab functions “mean” and “std”. Critical values from the Student’s t-Distribution are: 2.20 for a sample size of twelve and 95% confidence level, 2.23 for a sample size of eleven and 95% confidence level, 3.11 for sample size twelve and 99% confidence level, and 3.17 for sample sizes eleven and 99% confidence level. With the sample means, standard deviations, critical values, and sample sizes the confidence interval is calculated and plotted with the sample mean.

Appendix E: (Continued)

M-file *95CIstd2.m* begins here.

```
% 95% Confidence Intervals for Minimum Preloads at cycles 10,  
% 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250.  
  
% "mdp" is the total number of data points to be plotted. "dpc"  
% is the data point count. It controls the range of data points  
% to be selected to choose a specific data point. "dpcm" is the  
% data point count median. It is use to select the median cycles  
% to be plotted against the lowest data point from each group.  
% "dpcm" could be change to be the end lowest cycles as well by  
% making it equal to "dpc".  
  
clear all  
  
mdp=128;  
dpc=8192/mdp;  
dpcm=dpc/2;  
  
% C arrays are the minimum preloads for each range of data  
% points.  
% D arrays are the medians of cycles for each range of data  
% points.  
  
load test1.vna -mat;  
  
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;  
  
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C1(i)=p(1);  
d=cycles(a:b);  
D1(i)=d(dpcm);  
end;  
  
load test3.vna -mat;
```

Appendix E: (Continued)

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C2(i)=p(1);  
d=cycles(a:b);  
D2(i)=d(dpcm);  
end;
```

```
load test4.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C3(i)=p(1);  
d=cycles(a:b);  
D3(i)=d(dpcm);  
end;
```

```
load test7.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;
```

Appendix E: (Continued)

```
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
end;

load test8.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C5(i)=p(1);
d=cycles(a:b);
D5(i)=d(dpcm);
end;

load test11.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;
```

Appendix E: (Continued)

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C6(i)=p(1);
d=cycles(a:b);
D6(i)=d(dpcm);
end;

load test15.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C7(i)=p(1);
d=cycles(a:b);
D7(i)=d(dpcm);
end;

load test21.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
```

Appendix E: (Continued)

```
        a=1;
        b=dpc;
    else
        a=i*dpc-dpc;
        b=(i+1)*dpc-dpc;
    end;
    p=preload(a:b);
    p=min(p);
    C8(i)=p(1);
    d=cycles(a:b);
    D8(i)=d(dpcm);
end;
```

```
load test25.vna -mat;
```

```
time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;
```

```
for i=1:mdp
    if i==1
        a=1;
        b=dpc;
    else
        a=i*dpc-dpc;
        b=(i+1)*dpc-dpc;
    end;
    p=preload(a:b);
    p=min(p);
    C9(i)=p(1);
    d=cycles(a:b);
    D9(i)=d(dpcm);
end;
```

```
load test26.vna -mat;
```

```
time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;
```

```
for i=1:mdp
    if i==1
        a=1;
```

Appendix E: (Continued)

```
        b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C10(i)=p(1);
d=cycles(a:b);
D10(i)=d(dpcm);
end;

load test28.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C11(i)=p(1);
d=cycles(a:b);
D11(i)=d(dpcm);
end;

load test33.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
```

Appendix E: (Continued)

```
        a=i*dpc-dpc;
        b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;

td=[C1;C2;C3;C4;C5;C6;C7;C8;C9;C10;C11;C12];

td=transpose(td);

sum=C1+C2+C3+C4+C5+C6+C7+C8+C9+C10+C11+C12;
mean1=sum/12;

% n is the number of test run
n=12;

for i=1:n;
    m1(i)=td(5,i);
    m2(i)=td(18,i);
    m3(i)=td(31,i);
    m4(i)=td(44,i);
    m5(i)=td(58,i);
    m6(i)=td(71,i);
    m7(i)=td(84,i);
    m8(i)=td(98,i);
    m9(i)=td(111,i);
    m10(i)=td(124,i);
end;

mm1=mean(m1);
mm2=mean(m2);
mm3=mean(m3);
mm4=mean(m4);
mm5=mean(m5);
mm6=mean(m6);
mm7=mean(m7);
mm8=mean(m8);
mm9=mean(m9);
mm10=mean(m10);

mm=[mm1,mm2,mm3,mm4,mm5,mm6,mm7,mm8,mm9,mm10];
mm=transpose(mm);
```


Appendix E: (Continued)

```
sdm1=std(m1);
sdm2=std(m2);
sdm3=std(m3);
sdm4=std(m4);
sdm5=std(m5);
sdm6=std(m6);
sdm7=std(m7);
sdm8=std(m8);
sdm9=std(m9);
sdm10=std(m10);
sdm=[sdm1,sdm2,sdm3,sdm4,sdm5,sdm6,sdm7,sdm8,sdm9,sdm10];
sdm=transpose(sdm);

% 2.20 is the upper critical value for the CI which comes from
% Percentage Points of the t-Distribution table for
% t(alpha/2,N-1). Thus we have t(0.025,11) for a CI of 95 percent
% and 12 samples
ta=2.20;

for i=1:10;
    ciu(i)=mm(i)+ta*sdm(i)/(12)^0.5;
    cil(i)=mm(i)-ta*sdm(i)/(12)^0.5;
end;

ci10=[ciu(1),cil(1)];
d10=[D1(5),D1(5)];
ci250=[ciu(2),cil(2)];
d250=[D1(18),D1(18)];
ci500=[ciu(3),cil(3)];
d500=[D1(31),D1(31)];
ci750=[ciu(4),cil(4)];
d750=[D1(44),D1(44)];
ci1000=[ciu(5),cil(5)];
d1000=[D1(58),D1(58)];
ci1250=[ciu(6),cil(6)];
d1250=[D1(71),D1(71)];
ci1500=[ciu(7),cil(7)];
d1500=[D1(84),D1(84)];
ci1750=[ciu(8),cil(8)];
d1750=[D1(98),D1(98)];
ci2000=[ciu(9),cil(9)];
d2000=[D1(111),D1(111)];
ci2250=[ciu(10),cil(10)];
d2250=[D1(124),D1(124)];

axes('FontSize',16);
```

Appendix E: (Continued)

```
plot(d10,ci10,'-k',d250,ci250,'-k',d500,ci500,'-k',d750,ci750,'-
k',D1,mean1,'--k','LineWidth',2);
axis([-50 2700 0 2500]);
ylabel('Preload (lb)','FontSize',18);
xlabel('Cycles','FontSize',18);
%legend();
grid;
M-File CI95std2.m ends here.
```

M-file *CI95locking2.m* begins here.

```
% 95% Confidence Intervals for Minimum Preloads at cycles 10,
% 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250.

% "mdp" is the total number of data points to be plotted. "dpc"
% is the data point count. It controls the range of data points
% to be selected to choose a specific data point. "dpcm" is the
% data point count median. It is use to select the median cycles
% to be plotted against the lowest data point from each group.
% "dpcm" could be change to be the end lowest cycles as well by
% making it equal to "dpc".

clear all

mdp=128;
dpc=8192/mdp;
dpcm=dpc/2;

% C arrays are the minimum preloads for each range of data
points.
% D arrays are the medians of cycles for each range of data
points.

load test2.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
```

Appendix E: (Continued)

```
p=min(p);
C1(i)=p(1);
d=cycles(a:b);
D1(i)=d(dpcm);
end;

load test9.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C2(i)=p(1);
d=cycles(a:b);
D2(i)=d(dpcm);
end;

load test10.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
```

Appendix E: (Continued)

```
D3(i)=d(dpcm);
end;

load test14.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
end;

load test16.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C5(i)=p(1);
d=cycles(a:b);
D5(i)=d(dpcm);
end;
```

Appendix E: (Continued)

```
load test18.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C6(i)=p(1);
d=cycles(a:b);
D6(i)=d(dpcm);
end;

load test24.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C7(i)=p(1);
d=cycles(a:b);
D7(i)=d(dpcm);
end;

load test27.vna -mat;
```

Appendix E: (Continued)

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C8(i)=p(1);  
d=cycles(a:b);  
D8(i)=d(dpcm);  
end;
```

```
load test30.vna -mat;
```

```
time = SLM.tdxvec;  
preload = 6000*SLM.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C9(i)=p(1);  
d=cycles(a:b);  
D9(i)=d(dpcm);  
end;
```

```
load test32.vna -mat;
```

```
time = SLM.tdxvec;
```

Appendix E: (Continued)

```
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C10(i)=p(1);  
d=cycles(a:b);  
D10(i)=d(dpcm);  
end;
```

```
load test35.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
    a=1;  
    b=dpc;  
else  
    a=i*dpc-dpc;  
    b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C11(i)=p(1);  
d=cycles(a:b);  
D11(i)=d(dpcm);  
end;
```

```
load test36.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

Appendix E: (Continued)

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;
```

```
% C12 is test 36 or run 20 which is a larger outlier, here it is
% taken out an n is changed to 11
```

```
td=[C1;C2;C3;C4;C5;C6;C7;C8;C9;C10;C11];
```

```
td=transpose(td);
```

```
sum=C1+C2+C3+C4+C5+C6+C7+C8+C9+C10+C11;
```

```
mean1=sum/11;
```

```
% n is the number of test run
```

```
n=11;
```

```
for i=1:n;
    m1(i)=td(5,i);
    m2(i)=td(18,i);
    m3(i)=td(31,i);
    m4(i)=td(44,i);
    m5(i)=td(58,i);
    m6(i)=td(71,i);
    m7(i)=td(84,i);
    m8(i)=td(98,i);
    m9(i)=td(111,i);
    m10(i)=td(124,i);
end;
```

```
mm1=mean(m1);
```

```
mm2=mean(m2);
```

```
mm3=mean(m3);
```

```
mm4=mean(m4);
```


Appendix E: (Continued)

```
mm5=mean(m5);
mm6=mean(m6);
mm7=mean(m7);
mm8=mean(m8);
mm9=mean(m9);
mm10=mean(m10);

mm=[mm1,mm2,mm3,mm4,mm5,mm6,mm7,mm8,mm9,mm10];
mm=transpose(mm);

sdm1=std(m1);
sdm2=std(m2);
sdm3=std(m3);
sdm4=std(m4);
sdm5=std(m5);
sdm6=std(m6);
sdm7=std(m7);
sdm8=std(m8);
sdm9=std(m9);
sdm10=std(m10);

sdm=[sdm1,sdm2,sdm3,sdm4,sdm5,sdm6,sdm7,sdm8,sdm9,sdm10];
sdm=transpose(sdm);

% 2.23 is the upper critical value for the CI which comes from
% Percentage Points of the t Distribution table for
% t(alpha/2,N-1). Thus we have t(0.025,10) for a CI of 95 percent
% and 11 samples

ta=2.23;

for i=1:10;
    ciu(i)=mm(i)+ta*sdm(i)/(11)^0.5;
    cil(i)=mm(i)-ta*sdm(i)/(11)^0.5;
end;

ci10=[ciu(1),cil(1)];
d10=[D1(5),D1(5)];
ci250=[ciu(2),cil(2)];
d250=[D1(18),D1(18)];
ci500=[ciu(3),cil(3)];
d500=[D1(31),D1(31)];
ci750=[ciu(4),cil(4)];
d750=[D1(44),D1(44)];
ci1000=[ciu(5),cil(5)];
d1000=[D1(58),D1(58)];
ci1250=[ciu(6),cil(6)];
d1250=[D1(71),D1(71)];
```

Appendix E: (Continued)

```
ci1500=[ciu(7),cil(7)];
d1500=[D1(84),D1(84)];
ci1750=[ciu(8),cil(8)];
d1750=[D1(98),D1(98)];
ci2000=[ciu(9),cil(9)];
d2000=[D1(111),D1(111)];
ci2250=[ciu(10),cil(10)];
d2250=[D1(124),D1(124)];

axes('FontSize',16);
plot(d10,ci10,'-k',d250,ci250,'-k',d500,ci500,'-k',d750,ci750,'-
k',d1000,ci1000,'-k',d1250,ci1250,'-k',d1500,ci1500,'-
k',d1750,ci1750,'-k',d2000,ci2000,'-k',d2250,ci2250,'-
k',D1,mean1,'--k','LineWidth',2);
axis([-50 2700 0 2500]);
ylabel('Preload (lb)','FontSize',18);
xlabel('Cycles','FontSize',18);
%legend();
grid;
```

M-file *CI95locking2.m* ends here.

M-file *CI95loctite2.m* begins here.

```
% 95% Confidence Intervals for Minimum Preloads at cycles 10,
% 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250.

% "mdp" is the total number of data points to be plotted. "dpc"
% is the data point count. It controls the range of data points
% to be selected to choose a specific data point. "dpcm" is the
% data point count median. It is use to select the median cycles
% to be plotted against the lowest data point from each group.
% "dpcm" could be change to be the end lowest cycles as well by
% making it equal to "dpc".

clear all

mdp=128;
dpc=8192/mdp;
dpcm=dpc/2;

% C arrays are the minimum preloads for each range of data
% points.
% D arrays are the medians of cycles for each range of data
% points.

load test5.vna -mat;
```

Appendix E: (Continued)

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C1(i)=p(1);  
d=cycles(a:b);  
D1(i)=d(dpcm);  
end;
```

```
load test6.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;  
cycles = 15*time;
```

```
for i=1:mdp  
if i==1  
a=1;  
b=dpc;  
else  
a=i*dpc-dpc;  
b=(i+1)*dpc-dpc;  
end;  
p=preload(a:b);  
p=min(p);  
C2(i)=p(1);  
d=cycles(a:b);  
D2(i)=d(dpcm);  
end;
```

```
load test12.vna -mat;
```

```
time = SLm.tdxvec;  
preload = 6000*SLm.scmeas(2).tdmeas;
```

Appendix E: (Continued)

```
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C3(i)=p(1);
d=cycles(a:b);
D3(i)=d(dpcm);
end;

load test13.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C4(i)=p(1);
d=cycles(a:b);
D4(i)=d(dpcm);
end;

load test17.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;
```

Appendix E: (Continued)

```
for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C5(i)=p(1);
d=cycles(a:b);
D5(i)=d(dpcm);
end;

load test19.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C6(i)=p(1);
d=cycles(a:b);
D6(i)=d(dpcm);
end;

load test20.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
```

Appendix E: (Continued)

```
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C7(i)=p(1);
d=cycles(a:b);
D7(i)=d(dpcm);
end;

load test22.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C8(i)=p(1);
d=cycles(a:b);
D8(i)=d(dpcm);
end;

load test23.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
```

Appendix E: (Continued)

```
end;
p=preload(a:b);
p=min(p);
C9(i)=p(1);
d=cycles(a:b);
D9(i)=d(dpcm);
end;

load test29.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C10(i)=p(1);
d=cycles(a:b);
D10(i)=d(dpcm);
end;

load test31.vna -mat;

time = SLM.tdxvec;
preload = 6000*SLM.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
```

Appendix E: (Continued)

```
C11(i)=p(1);
d=cycles(a:b);
D11(i)=d(dpcm);
end;

load test34.vna -mat;

time = SLm.tdxvec;
preload = 6000*SLm.scmeas(2).tdmeas;
cycles = 15*time;

for i=1:mdp
if i==1
    a=1;
    b=dpc;
else
    a=i*dpc-dpc;
    b=(i+1)*dpc-dpc;
end;
p=preload(a:b);
p=min(p);
C12(i)=p(1);
d=cycles(a:b);
D12(i)=d(dpcm);
end;

td=[C1;C2;C3;C4;C5;C6;C7;C8;C9;C10;C11;C12];
td=transpose(td);

sum=C1+C2+C3+C4+C5+C6+C7+C8+C9+C10+C11+C12;
mean1=sum/12;

% n is the number of test run
n=12;

for i=1:n;
    m1(i)=td(5,i);
    m2(i)=td(18,i);
    m3(i)=td(31,i);
    m4(i)=td(44,i);
    m5(i)=td(58,i);
    m6(i)=td(71,i);
    m7(i)=td(84,i);
    m8(i)=td(98,i);
    m9(i)=td(111,i);
```


Appendix E: (Continued)

```
        m10(i)=td(124,i);
end;

mm1=mean(m1);
mm2=mean(m2);
mm3=mean(m3);
mm4=mean(m4);
mm5=mean(m5);
mm6=mean(m6);
mm7=mean(m7);
mm8=mean(m8);
mm9=mean(m9);
mm10=mean(m10);

mm=[mm1,mm2,mm3,mm4,mm5,mm6,mm7,mm8,mm9,mm10];
mm=transpose(mm);

sdm1=std(m1);
sdm2=std(m2);
sdm3=std(m3);
sdm4=std(m4);
sdm5=std(m5);
sdm6=std(m6);
sdm7=std(m7);
sdm8=std(m8);
sdm9=std(m9);
sdm10=std(m10);

sdm=[sdm1,sdm2,sdm3,sdm4,sdm5,sdm6,sdm7,sdm8,sdm9,sdm10];
sdm=transpose(sdm);

% 2.20 is the upper critical value for the CI which comes from
% Percentage Points of the t Distribution table for
% t(alpha/2,N-1). Thus we have t(0.025,11) for a CI of 95 percent
% and 12 samples

ta=2.20;

for i=1:10;
    ciu(i)=mm(i)+ta*sdm(i)/(12)^0.5;
    cil(i)=mm(i)-ta*sdm(i)/(12)^0.5;
end;

ci10=[ciu(1),cil(1)];
d10=[D1(5),D1(5)];
ci250=[ciu(2),cil(2)];
```

Appendix E: (Continued)

```
d250=[D1(18),D1(18)];
ci500=[ciu(3),cil(3)];
d500=[D1(31),D1(31)];
ci750=[ciu(4),cil(4)];
d750=[D1(44),D1(44)];
ci1000=[ciu(5),cil(5)];
d1000=[D1(58),D1(58)];
ci1250=[ciu(6),cil(6)];
d1250=[D1(71),D1(71)];
ci1500=[ciu(7),cil(7)];
d1500=[D1(84),D1(84)];
ci1750=[ciu(8),cil(8)];
d1750=[D1(98),D1(98)];
ci2000=[ciu(9),cil(9)];
d2000=[D1(111),D1(111)];
ci2250=[ciu(10),cil(10)];
d2250=[D1(124),D1(124)];

axes('FontSize',16);
plot(d10,ci10,'-k',d250,ci250,'-k',d500,ci500,'-k',d750,ci750,'-
k',d1000,ci1000,'-k',d1250,ci1250,'-k',d2250,ci2250,'-
k',D1,mean1,'--k','LineWidth',2);
axis([-50 2700 0 2500]);
ylabel('Preload (lb)','FontSize',18);
xlabel('Cycles','FontSize',18);
%legend();
grid;
```

M-file *CI95loctite2.m* ends here.

M-file *CI99std2.m* is identical to *CI95std2.m* except $ta=3.11$.

M-file *CI99locking2.m* is identical to *CI95locking2.m* except $ta=3.17$.

M-file *CI99loctite2.m* is identical to *CI95loctite2.m* except $ta=3.11$.