



## **Design of a Modular Cloud Storage Gaging System for Automotive Manufacturing**

**Mr. Mark Jeffrey ZurSchmiede, Federal Screw Works**

I am a Computer Engineering Graduate Student at Grand Valley State University and a practicing software engineer at Federal Screw Works. I spent the first four years of my career designing embedded software for the Aerospace and Medical industries and took a new job at an Automotive manufacturing facility. This latest project seeks to merge the automotive manufacturing world with embedded systems world by designing custom gaging solutions for the companies automotive parts.

**Dr. Jeffrey Ward, Grand Valley State University**

# Design of a Modular Cloud Storage Gaging System for Automotive Fastener Manufacturing

Mark ZurSchmiede, Dave Bartell, Dr. Jeffrey Ward

April 6, 2015

## Abstract

The proposed research project will involve the electrical and software design of an automated gaging system for automotive parts. Using an optical micrometer, the proposed gage will construct a virtual 3D image of a cylindrical part and then extract dimensional information from the 3D image. The design of the gaging system has two primary objectives. First, to record critical dimensions on every part that comes through a manufacturing line and provide alerts if the system has fallen outside of the maximum tolerances specified for any critical dimension. Second, to report the dimensions of all parts to a cloudbased database allowing managers to check process statistics remotely. All parts that are produced on manufacturing lines with this gaging system integrated must be marked with a serial number allowing any part run through the system to be matched to its dimensional data in the database as well any other critical information about the part.

In the past, common problems have arisen when trying to implement automated gaging. These include contaminants on the part corrupting the gaging system, uneven surfaces on the part causing probe tips to move in unpredictable ways and problems with the steel blanks such as warped thread blanks which cause a part to sit in a gaging fixture improperly. The gage design will account for all of these failure modes by using optics to eliminate mechanical problems and software filters to eliminate outliers in datasets. In addition, the gage design will provide a shallow learning curve in the manufacturing realm by using a Programmable Logic Controller (PLC) to control the state machine of the system with an embedded controller processing the data from the gage and responding to analysis requests from the PLC.

# 1 Background

The primary focus of this project is the design and testing of two systems. A precision gaging system for measuring dimensions of cylindrical automotive parts and the design of a cloud storage database for storing information about the parts. Cell style manufacturing has yielded a need for gaging systems that are in line to the processing of the parts, meaning occurring in sequential operations. Having the gaging in line with the processing allows the system to detect much more quickly if the process metrics have gone outside of specified boundaries and reduces the amount of material that needs to be scrapped as a result of a bad manufacturing process. Further, modern technology has made cloud data storage not only accessible but reasonably inexpensive. Storing information about every part that is manufactured is becoming more common place because it can greatly reduce the overall cost of a recall if the exact problem in the process can be targeted and parts that are bad to be filtered. For example, now instead of targeting parts that were manufactured between two dates, the system can find barcodes that were cut on a CNC with a misaligned spindle and the system can produce a list of all the barcodes that customers should scrap from their supply.

The design of the two systems each has their own objectives. The primary objective of the gaging system is to record critical dimensions on every part that comes through the manufacturing line and provide alerts if the system has fallen outside of the maximum tolerances specified for any critical dimension. The objective of the database storage system is to report the dimensions of all parts to a cloud-based database allowing managers to check process statistics remotely. In order to store the data, all parts that are manufactured on lines with this gaging system integrated will be marked with a serial number allowing any part run through the system to be matched to its dimensional data in the database as well as the day, time, and shipment number of the part. This portion of the project is critically important as it will allow customers to request dimensional information about a certain part, and the system will allow us to find parts that were in the same shipment and even which parts in that shipment were machined on the same chuck to be able to narrow down the defective parts in the event of a recall.

The measurement that needs to be taken in the initial revision of this gage is inner diameter (ID) to outer diameter (OD) coaxiality. Constrained by the particular process in which some of the features are being cold formed, as well as constraints of using existing machinery to cut the new part, a manufacturing process has been engineered by which the OD of the part will be machined on a different machine than the ID. For this reason, special care needs to be taken to ensure that the part wasn't misloaded into the first machine as it would greatly distort some of the measured dimensions of the features produced during the cold forming process. This is the primary reason for the gage on this first iteration. The gage will hold the ID of the part in place with a mandrel and then spin the part while measuring the edges of the OD. In this way the center axis of the OD can be interpolated through software, and the gage can ensure that the part was not misloaded into the machine which machines the OD of the part. This gage concept is more than is necessary to take this particular measurement, but the intention is to refine and expand this idea to include all manner of dimensional measurements. The gage system constructs a virtual 3D image of the entire part and then extracts dimensional information from that 3D image.

The second measurement to be taken was added during the design of the gage. This involved taking the diameter of a location on the part. After the part goes through the gage it enters another machine that drills and reams the inside of the part. When the part is sitting in a chuck, the center axis of the part can be affected by how large or small the diameter is on the part as it is being gripped from the outside and only the top gripper clamps down on the part while the bottom gripper never moves. An increase in diameter will cause the center axis of the part to move upward. However as some of the operations on this machines are controlled by servos it is possible to send the diametrical measurement to this machine and have it offset the servos thus bringing everything back to the same center axis.

## 2 Overview of system

### 2.1 Mechanical

The mechanical concept of the gage is being driven largely by the capabilities of the 2D scanner being used. Figure 1 displays the type of optical comparator which was chosen to detect the OD edges of the part. This 2D scanner is capable of taking measurements on an input pulse and saving the data to its own internal memory to be read back

later. For this reason, it is desired to set up the mechanical actuators to drive the system using only one motor. To accomplish this, an encoder was connected to the motor and the output pulse of the encoder was connected to the gage to drive the sampling of the data. The mechanics of this design provide a great advantage as far as tracking the sampling of the part. During the initial conceiving phase, one of the primary challenges was how to overcome the latencies between positional sensors monitoring the position of the part and the sampling of gage. This system overcomes those challenges in the same way that a distributor in an engine times the firing of the cylinders albeit at a much faster rate. It allows the mechanics of the system to drive the sampling of the gage. In this system, there is no need to account for informational latencies because the scanner is told by the encoder pulse to take a sample when it is in the correct position.

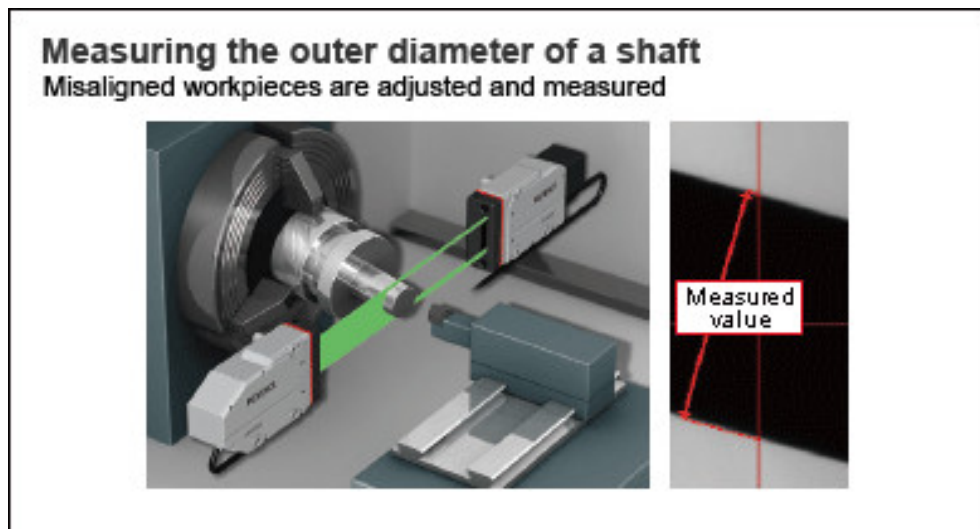


Figure 1: Optical gage for measuring diameter of a part

The mechanical setup of the system will drive a linear and rotary actuator with a servo using a ball screw actuator. The center of the ball screw travels through a bearing that is fixed in position. As the ball screw turns, the pitch of the screw will also cause it to move forward like a screw moves into a threaded hole. The bearing is fixed in position with the rest of the assembly mounted on a plate connected to a pair of precision guide rails which allows the entire assembly to move along one axis. By fixturing the part at the end of this setup, the part can traverse in a helical pattern while the optical micrometer can scans the OD. The mandrel will fixture the part by its internal diameter (ID). The encoder used to drive the sampling of the optical micrometer is attached in line with the ball screw. Figure 2 displays the mechanical setup of the gage with references to each of these parts. The optical micrometer can take sample readings on an encoder pulse so this setup allows the physical rotation of the part to trigger samples to be taken on the optical micrometer. The optical micrometer can store up to 400,000 samples which are downloaded to another CPU for analysis.

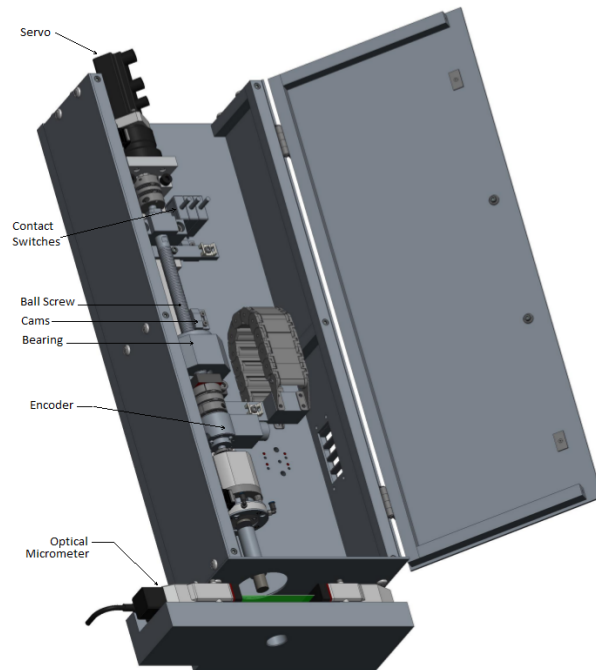


Figure 2: Mechanical layout of the Gage

## 2.2 Electrical

The electrical system of the first iteration of the gage is fairly simple. The design brought on by the concept of this system is a hybrid PLC and Embedded CPU design. The PLC controls the vast amount of IO in the system and is run on ladder logic to allow the system design to have a layout more easily understandable to an industrial electrician. The more complex needs of the project are integrated into the Embedded CPU. This helps provide a shallower learning curve for engineers to gain a fundamental understanding of how the manufacturing cell functions, but still allows room for more advanced operations inside the cell which would otherwise be far too cumbersome to do with ladder logic alone.

The system consists of three different controllers. The first is a sequencing PLC which is used to control various aspects of the automated machinery surrounding the part as well as process the state machine for the gage operations. The second CPU is the servo positioning controller. The servo controller gives the PLC a simple interface in which to move the servo to specific positions, travel at specific velocities, and apply specific amounts of torque through a motor. Finally, the embedded CPU controls all serial communication with external peripherals and systems and contains the software libraries for analyzing the point data fed back from the sensor as well as creating and analyzing the virtual 3D image. Also, the embedded device manages all the information about each part that is measured by the gage until that information is needed to be stored in the cloud database, as well as actually does the process of uploading the data when the time comes.

The 2D scanner communicates with the embedded CPU over Ethernet. Keyence, the manufacturer of the 2D scanner, has provided a library for communication with the 2D scanner over Ethernet or USB connection from a Windows PC. For this reason, the Embedded PC selected was a Q10WCPU, which is an embedded CPU running Windows Embedded Standard 2009. The embedded CPU sits on a communication bus with a PLC and motion controller CPU which drives the servo. The Embedded CPU communicates over the backplane bus with a QD75MH1 positional controller for controlling the servo. This is a simple positional control. The servo is programmed with positional data that instructs the servo where to travel, how fast to traverse, and how quickly to accelerate and decelerate. The controller card is able to detect the load attached and in many instances automatically adjust its feedback parameters for optimal movement. Additionally, a PLC was integrated to control the state machine and general control of the

gage. A diagram of the communications of the various parts of the system can be seen in fig. 3.

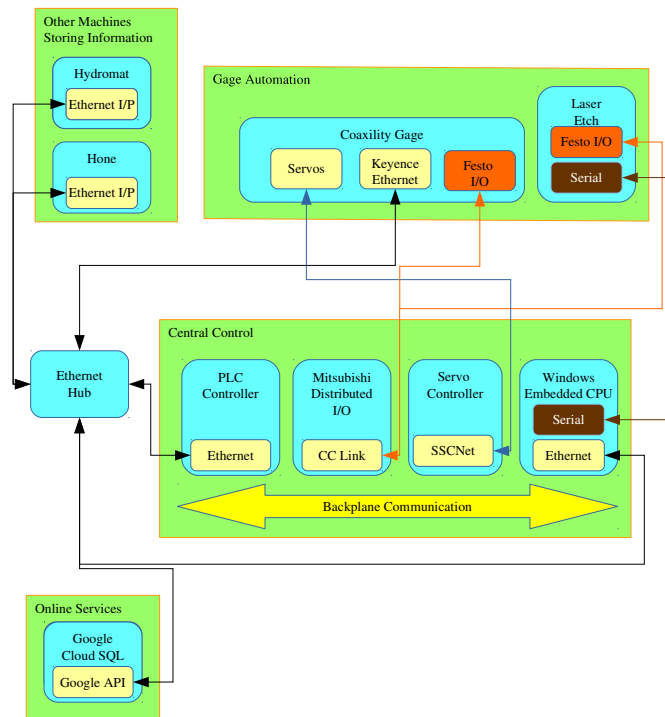


Figure 3: Communication Diagram for the various electrical components in the gage

## 2.3 Software

There are two software design concepts which make up the lion's share of the work for this project. There is the gage concept, which details the components of the gage software that will be analyzing the part, and the data storage concept which details how the information about the part is tracked and stored. Tying these two systems together is the serialization of each part.

### 2.3.1 Gage Concept

The software for the gage includes the libraries necessary to download and analyze the data. The gage samples are taken by allowing the mechanical setup to trigger samples to be taken at specific intervals while traversing the part in a helical movement. Using the position of the linear/rotary actuator and the diameter of the part at the specific position, a 3D rendering of the image is constructed, and the system uses this simple design to decipher very complex measurements of a part. The strength of this design is that it allows very complex measurements to be made on a part aside from simple diametrical measurements such as coaxiality, runout, roundness, taper, and hopefully even radius measurements. It could be applied to any part which is mostly spherical or cylindrical in nature.

### 2.3.2 Gage Mastering

The Gage is primarily used for finding the coaxiality of a particular part. This means that the gage is intending to find the angular difference axis of the ID of a part and the OD of a part. The gage itself is capable of measuring

the OD of the part but not the ID. The initial concept for the gage was that the mechanical design of the gage would be tailored to center the ID of the part in perfect alignment of the center point of the gage. This was a grand idea but not very practical. There is always some degree of misalignment even in the most precise gage. Adding to this problem is that coaxiality is being measured by computing the maximum ID to OD offset over the length of the part. This measurement is amplified the longer it is stretched out along a cylindrical surface so any misalignment of the mechanism holding the part causes tremendous accuracy problems in the gage.

Software mastering of the gage was done to counteract these problem. The concept was to rotate the mandrel, which is the mechanism that holds the part, 180 degrees while taking measurements along the surface and recording the ID to OD offset measurement. The mandrel has relief cuts in line with the spinning axis to allow the material to move slightly while finding the ID of the part so it was not practical to try to measure the center of the mandrel itself. Instead, a precision ground mastering ring was placed over the Mandrel and used to find the ID of the mandrel. The idea was that if the part used to master the gage was ground as precisely as possible then the OD offset measurements from this part could be used as the axis offset of the mandrel. These OD offset measurements were recorded and the offsets at each angular position were to be used to offset the measurements taken on a part at each position.

The only way that this idea works is if the system knows the angular position of the mandrel so that the offsets taken during mastering can be applied to the offsets taken during measurement of the part. The angular position of the mandrel is found by searching the data for the z Pulse. In a typical encoder there are two signals that are 90 degrees out of phase with each other. A controller monitoring these encoder signals can uses the order of the rising and falling edges of each signal to determine if the encoder is rotating forward or backward. A third signal known as the z pulses once per revolution and is used for the controller to find the encoders angular rotation. An image of the timing of these pulses can be found in fig. 4. The figure shows that as the encoder rotates it finds the z pulse twice. Once as it is rotating forward, and again after it stops and begins rotating backwards.

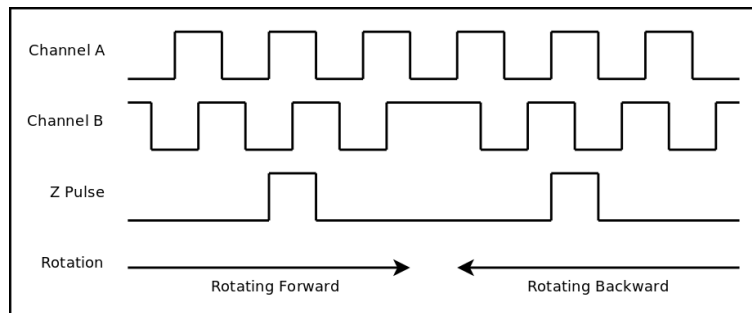


Figure 4: Encoder Pulses

In addition to using an encoder to drive the sampling of the LS9000 gage, the LS9000 also allows the programmer to offset the pulse count in the gage data when it sees this z pulse. In the data, this appears as a shift of the pulse count from -1800 to -1 because the resolution of the encoder is 1800 pulses per revolution. In this application, the z pulse is used to find the angular position of the gage. When the data is read back from the encoder it searches for the first -1 in the pulse count and throws away all data found before this. After finding this first -1 the gage looks for the next -1 in the data, if the next -1 is preceded by a value of -1800 the z pulse has been found and all data is collected after that point. If the -1 is not preceded by a -1800 this data is thrown away again, and the algorithm continues searching for the next -1. This allows the master offsets and the measurements to sync up their angular positions. This is intended to remove any problem of a concentricity offset between the ID and OD.

### 2.3.3 Database Storage Concept

The database storage is the subsystem that allows the manufacturing line to store dimensional data for every part that runs through the system. Once the part has been run through the entire system, the data is stored in an online database called Google Cloud SQL. Google Cloud SQL is a standard MySQL database that runs on Google's Servers. They provide an IP address with which to connect to the database and encryption keys to use with connecting.



The database is capable of storing more data than simply the information saved from the gage and can recall this information via serial number. A variety of verification checks occur throughout the manufacturing line of these parts in addition to the information that is captured from this gage. Furthermore, the company can store information about which particular chucks that the parts were fixtured in when they were being machined in case there is a problem and they want to sort which parts were machined in a bad station.

The Embedded CPU communicates directly with the online database using the freely distributed open source MySQL Connector C++ library provided by Oracle. Using this library, the system can programmatically store data to allow customers who wish to retrieve this information access to the data in the event of a quality problem.

### **3 Educational Purposes**

The educational focus of this project involves designing the gage with as shallow of a learning curve as possible to allow new programmers to become familiar with the software as quickly as possible. As this is an industrial application most of the engineers working in the facility in which it is used have only concerned themselves with ladder logic programming. However, ladder-logic, while providing a rich visual interface to an unexperienced programmer leaves a lot to be desired when it comes to more advanced programming applications. For this reason, the system was developed as a hybrid system, with an embedded controller handling the processing of the gage data and a PLC controlling the industrial automation. The idea is to allow ladder logic programmers to have an easy jump into C++ programming.

The educational purposes of this tool are not for classroom-style education but instead a very hands-on real world application example. The gage was built for an industrial manufacturing cell but designed for the fulfillment of a Master's level thesis project at Grand Valley State University. The educational design of the tool came from several discussions between the Master's level student designing the gage and one of the engineering managers at the industrial location. The student's desire was to design the entire codebase for the gage in C++ as this programming language is far more advanced than ladder logic and does not provide as many barriers when trying to design algorithms. The manager's desire was that the design of the gage be programmed in ladder logic to ensure that industrial technicians would be able to read and redesign the code if necessary. In the end, a compromise was reached in which the system would be partially designed in ladder logic to allow industrial electricians to interface with the code and partially designed C++ to allow more advanced operations to be integrated.

The two CPUs communicate back and forth over a backplane and are each provided an Application Programming Interface (API) for sharing an area of with the other device. From this area, a mechanism has been set up on the Embedded CPU to allow the PLC to make function calls to the Embedded CPU. This hybrid setup allows a technician to more easily gain a grasp on the more advanced programming aspects of the gage by examining only the details of the function that is called by the PLC without having to understand how the entire program functions. This mechanism provides a convenient learning bridge to help the new programmer learn how C++ programs are designed.

### **4 Testing**

All automated testing is done through the Google Test Framework which provides a system for doing test assertions. The Google Test Framework was chosen for its ability to mock out classes. Mocking classes means providing fake classes for the class under test to use. Mocking is necessary to control a class which needs to communicate with external devices or use external libraries. Very often, external peripherals such as databases, or USB hardware are not available during the unit tests and the preferred way is to use Mock classes in place of these classes so that the inputs can be checked and the return values can be controlled.

## 4.1 Testing the Cloud Database

The first test was written during the initial design of the database and was written to ensure that it could communicate with a local MySQL database properly. A MySQL database was set up on the local machine. The first part of the test ensures that it can establish a connection to this local database. The second part of the test populates a dataset and uploads it to the database. The third part of the test ensures that if the database tries to rewrite this information to the database, that the database throws an exception as there cannot be two pieces of information with the same key which in this case is the unique serial number of each part. The fourth part of this test ensures that the read command can read back each element and verifies that the data stored in the database is correct.

In the next two tests the system must be connected to a certain network with a MySQL database at a specific IP address to work. The only way to test the system in the event of an unreliable network was to manipulate the network while the system was connected to an external database. The first test checks the general usage of the functions used to upload and download data from the gage. The second test identifies that the system responds appropriately in the event of a network connection loss. The actual database is in the cloud so the system needs to be able to reconnect appropriately after a loss of internet connection.

The tests connect to an actual MySQL database set up on an external computer using the external computers IP address. Using bash scripts, the test creates a MySQL database and adds in the necessary tables that the function is expecting to be present in the real model. At the end of the test, the database is deleted to ensure that each test does not incur any side effects on other test runs. The test generates different data for 10 different part numbers and assigns these part number serial strings. All entries in the database are read back and then the data is verified to be the same as the data that was put into the cloud database.

The second script tests that the system will respond appropriately when the network connecting the system to the cloud database is down. Bash scripts have been written to drop and reconnect the network while the test is running. The test starts an asynchronous upload, waits for a short period of time and calls the bash script to drop the network and tests that the system appropriately throws an exception.

## 4.2 Testing the Gage

The first test of system was to establish how long it would take to download the data from the gage. For the initial application of this gage, the gage needed to be able to accomplish all of its operations in under 15 seconds as this was the targeted processing time of the manufacturing line. The Keyence library provided the ability to connect over either Ethernet or USB. The initial size to download was 50000 samples as this was believed to be the target number of samples. The system was designed to actively download the data while the scan was taking place. This design left only a minimal amount of data to be downloaded after the servo had completed its movement. Currently, the scan takes about 7 seconds with processing of the gage happening in about 3 seconds. Typically the analysis of the data has completed by the time the part is in position to be picked up. Ethernet was chosen as the active download mode as the active download style removed the need for high speed data transfer and a concern with the USB connection was that the USB driver could make the embedded device more susceptible to system crashes of the embedded OS.

### 4.2.1 Testing Coaxiality

Trying to find the coaxiality was the most challenging aspect of the entire project. The coaxiality problem can be seen in the data that follows as the increase in the amplitude of the waveform as it measures from top to bottom. A good way to think about the movement of the system with coaxiality problems is like a spinning top that is falling out of balance. If the ID axis of the part or the mandrel has a different angular position than OD then the OD will wobble like a top with the wobble increasing the further the gage is measuring from the fixture position.

In the first design of the system it was believed that there would be some mechanical misalignment in the spindle controlling the part. The results of this can be seen in fig. 5. In these results, the part being recorded is a straight cut cylinder which was turned in a lathe on the outside and precision ground on the inside to keep the first findings as simple as possible. The blue line marks the offset data taken from the gage when mastering the points for 360

degrees around the part. The first 1800 measurements are repeated for 40000 samples to show the comparison to the actual data. The red line indicates the measured results of the part from the gage. The green line indicates the subtraction of the measured results from the offset. From left to right displays the measurements from top to bottom on the part.

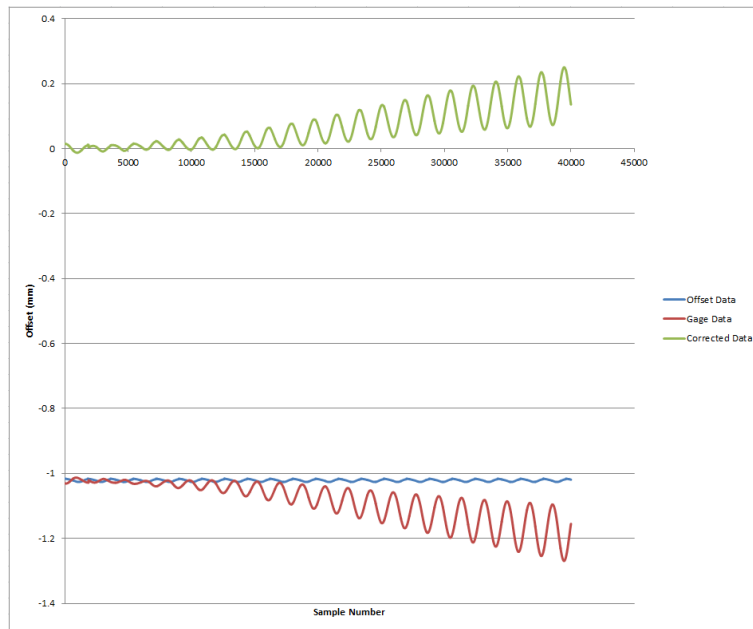


Figure 5: Data found during the single rotation mastering.

It is fairly easy to see why this initial idea of mastering was abandoned. If, as initially believed, the only problem that existed in the gage was concentricity, meaning that the center axis of the mandrel was not concentric to the center axis of the spindle that was rotated, then this solution would be acceptable. However, from this data it is clear that there were many more problems. First, there is an angularity problem, meaning that the linear movement of the gage carriage is not perpendicular to the gage itself. This can be seen because the center axis of the red line shows a steady descent on the graph. Second, there is a coaxiality problem. There is not enough data in this graph to show if this is a coaxiality problem in the part being measured, or the way it is being fixtured, or in the gage itself. The first coaxiality problem can be ruled out as this measurement was taken on a part that was precision ground and measured for coaxiality by measuring the ID to OD concentricity on an external gage. The external gage measures ID to OD concentricity by rolling the OD of the part on a precision ground tool steel fixture and measuring the change of the ID as the part rolls using a needle. The measurement found on this part was about 1 thousandth of an inch which is about 25 micron. The fallout of this problem shows concentricity problems of greater than 200 micron.

The solution to the problem of the gage itself having coaxiality problem (meaning that the mandrel OD is not aligned to the spindle that is spinning it) was to build a mastering ring that was as long as the length of the part. This mastering ring allowed the system to store offsets not just for 1800 measurements around the part but offsets for the part at every single position in which it would be measured, all 40000 samples. Using this method allows all mechanical ills of the system to be correct with offsets. A scan of the offsets of the part as well as the part itself can be found in fig. 6.

Using this method provided results that made much more sense. Looking at the new graph the green line shows perfectly the coaxiality problems in the part itself. This information is further verified by looking at the position where the sine wave amplitude seems to come to zero. This is the position where the mandrel touches the part. If the offset positions were not correcting the mechanical inaccuracies of the gage, then this center position would likely show up somewhere else or be projected off of the measurable data.

The next test of the coaxiality measurement was to make sure that the measurements were repeatable. This was done in two ways. First, by running multiple scans on one part to make sure that the micrometer and all associated software were repeating the measured value. Second, by removing the part from the gage and then refixturing the part to make

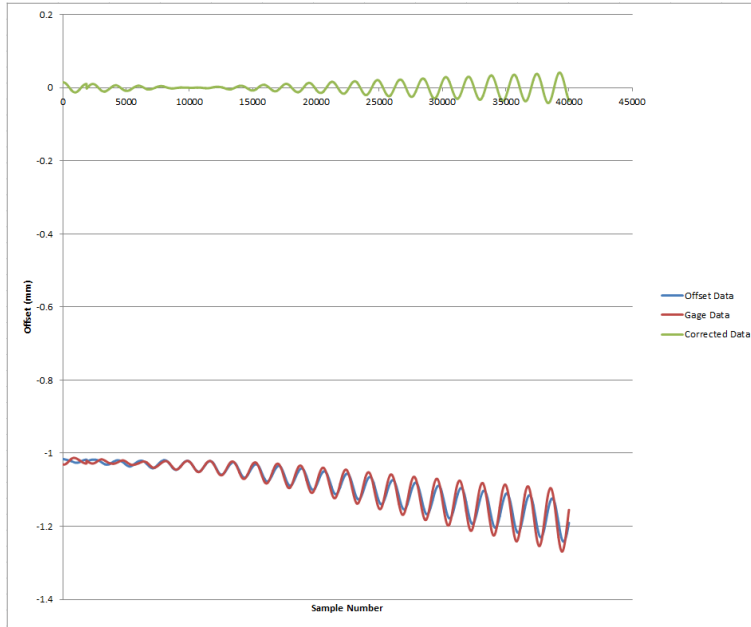


Figure 6: Data found after mastering the entire part.

sure that refixturing the part was not having a significant influence on the coaxiality measurement. For the company developing the gage, a gage is considered to be repeating successfully if the gage can consistently measure a dimension which is accurate to within 10% of the total tolerance of that dimension. In this case, the coaxiality measurement has a 200 micron total tolerance so the gage must be able to repeat the measurement to within 20 micron. Results of the first test can be seen in table 1. From the sample set the gage appears to be capable for coaxiality to within 12 micron.

Table 1: Coaxial results of sequential scanning of the same part

Sample	Coaxiality
1	0.008
2	0.005
3	0.0051
4	0.0052
5	0.004
6	0.0058
7	0.0063
8	0.0112
9	0.0091
10	0.0062

The next test was to see how much variation was found in the mechanical fixture which holds the part in place. This is the mandrel which at first had a slightly rounded shape but was ground flat. This test was done by unclamping and clamping the mandrel in between scans to see how this changed the test results compared to the previous test. The results of this test can be seen in table 2. From this data, the refixturing of the mandrel does seem to still be having an impact on the scans however the variability is still within the 20 micron threshold. This is acceptable criteria for the first iteration of the gage. Future iterations will need to be redesigned to improve this fixturing problem.

Table 2: Coaxial results of refixturing between scans

Sample	Coaxiality
1	0.0216
2	0.0151
3	0.0205
4	0.0181
5	0.0273
6	0.0165
7	0.0231
8	0.0127
9	0.0245
10	0.0229

#### 4.2.2 Testing Diameter

The first test of the diameter examines the capabilities of the gage. These initial tests proved much better than initially expected. The gage boasts 2 micron repeatability, meaning that on average sequential scans should be within 2 micron of the initial measurements. The diametrical tests validated this claim. The following figures show the data scanned from the part as can be seen by the gage interface program. The diametrical results of the scan can be seen in fig. 7.

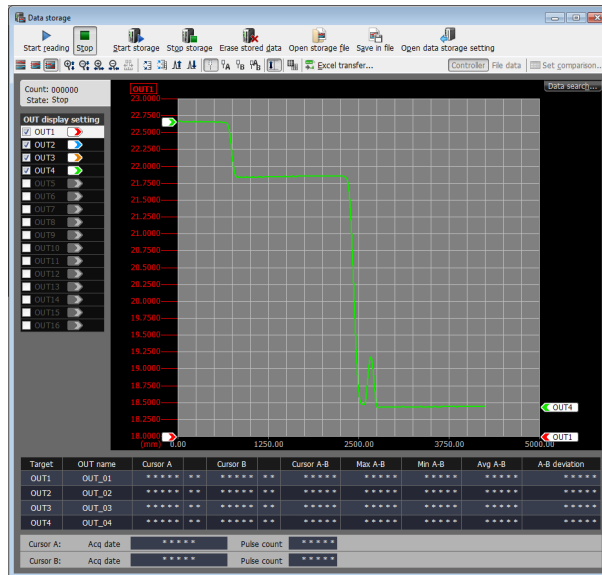


Figure 7: Diametrical Data from Gage

The same scan as above but zoomed in can be seen in fig. 8. This shows the data at the far left of fig. 7. At first, it was believed that humps in the data were simply noise from the rotation of the part until markers were placed on the graphs. Looking at the bottom of the screen the pulse count for the two markers can be seen. The markers lay a little over 900 pulses apart which is half of the pulses in one revolution of the gage. This part was taken out and measured in a micrometer and these results were confirmed. If the part is rotated 90 degrees the diameter will increase or decrease in accordance with the scan. This is called part roundness and is a common problem when cutting on lathes. The results of this analysis proved that the gage was not only capable enough to show diameter but could also be applied to a roundness test as well.

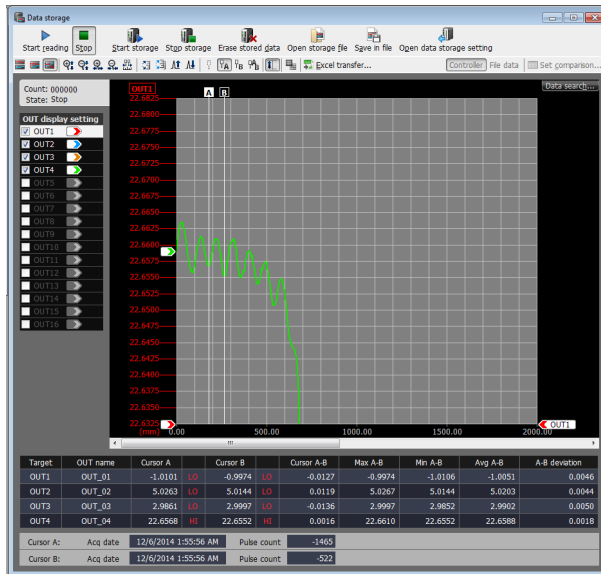


Figure 8: Diametrical Data, Zoomed in

The next test was testing the repeatability of the diametrical measurements the same as was done with the coaxiality in fig. 7. The diametrical measurements are found by taking the average of a range of measurements on the part. The following tests were conducted in the same manner as the coaxiality test done in section 4.2.1. The first test measures the data of sequential scans without refixturing the part to test the repeatability of the gage and software. From the results in table 3 the gage shows to be highly repeatable. However the diametrical data does show an upward trend in the measurement of parts. This is believed to be from the internal elements of the gage heating up though no tests were done to confirm this. It does show that the gage does have some variation in measurements.

Table 3: Diametrical results of sequential scanning of the same part

Sample	Diameter
1	21.8457
2	21.8457
3	21.8456
4	21.8456
5	21.8459
6	21.8458
7	21.8459
8	21.8459
9	21.846
10	21.8462

The following test shows the fixturing and refixturing of parts between scans to test if refixturing the parts has an impact on the diametrical measurements. The results in table 4 show similar data to what can be seen in table 3. From this it can be concluded that refixturing does not have any impact on the diametrical measurements.

Table 4: Results of refixturing between scans

Sample	Diameter
1	21.8464
2	21.8465
3	21.8465
4	21.8466
5	21.8467
6	21.8467
7	21.8469
8	21.847
9	21.8473
10	21.8471

## 5 Conclusion

The requirement set forth at the beginning of the project was that the gage was able to repeatably obtain measurement values within 10% of the total tolerance of the measurement. The allowable coaxiality for the part being measured was 200 micron giving the gage a range of 20 micron variation. The allowable diametrical tolerance was 100 micron giving the gage a range of 10 micron variation. The diametrical results are well within the allowable variation however the coaxiality readings while still within the allowable variation demonstrate that there are still issues that need to be resolved with the mechanical fixturing. The mandrel on which the part is resting needs to be improved as the reclamping of the mandrel does seem to have a significant influence on the measured coaxiality of the part.

## References

- [1] Nicolai M. Josuttis. *The C++ Standard Library: A Tutorial and Reference*, 2nd edition edition, April 2011.
- [2] Keyence Inc., Osaka, Japan. *High-speed, High-precision Micrometer Measuring Instrument LS9000 User's Manual*, January 2014.
- [3] Mitsubishi Electric Corporation, Japan. *Melservo-J3 Series Servo Amplifier Instruction Manual*, sh(na)030084-d edition, August 2012.
- [4] Mitsubishi Electric Corporation, Japan. *Type QD75MH Positioning Module - User's Manual*, sh(na)030084-d edition, January 2012.
- [5] Mitsubishi Electric Corporation, Japan. *Melsec Q/L/F Structured Programming Manual*, sh(na)-080782eng-m edition, July 2013.
- [6] Oracle. *MySQL Connector/C++ Developer Guide*, 42146 edition, 2015.