

SUB-SCALE MACHINING OF LARGE COMPONENTS

by

Paul Edward Anderson

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Mechanical Engineering

Charlotte

2015

Approved by:

Dr. John Ziegert

Dr. Jimmie Miller

Dr. Matt Davies

Dr. Edward Morse

ABSTRACT

PAUL EDWARD ANDERSON Sub-scale machining of large components
(Under the direction of DR. JOHN ZIEGERT)

This research examines the feasibility of machining components with precise tolerances using machines smaller than the completed part. This can conceptually be achieved in two ways. 1.) move a part around a machine, machining a portion of the part at a time, resulting in a completed part; and 2.) move a machine/multiple machines around a part, machining a portion of the part at a time, resulting in a completed part. At each step, a laser tracker was used to measure the spatial relationship between the part and machine coordinate systems (CS); and custom software was developed to read the part coordinates from the NC part program and replace them with the values corresponding to the new part/machine spatial relationship.

Several series of parts were machined to allow the determination of additional error from the laser tracker to the manufacturing process, and the precision of transitional areas between machined sections. A final test included machining a set of parts larger than the machine's working area to determine tolerances the machining process is capable of and demonstrate the project objectives have been met. Initial test parts showed part feature errors significantly larger than initially expected. A search for the cause of errors led to process improvements which showed the importance of proper measurement techniques to reduce influences from machine tool geometric errors and the importance of proper fixturing. After correcting for machine tool geometric errors and using proper fixturing techniques, part feature location errors were reduced to an expected value. A variability analysis of the sub-scale machining process was performed to compare

experimental and theoretical part feature errors. The variability analysis showed expected part feature errors on the order of experimental values. Sub-scale machining is capable of producing 2D parts with feature location errors on the order of the uncertainty of the reference metrology device.

Initial test parts showed the capabilities of sub-scale machining when used to make essentially two (2) dimensional parts. An analysis of the sub-scale machining process was performed to understand the capabilities of sub-scale machine as an operation to produce large three-dimensional parts with features on multiple sides. It is expected that sub-scale machining can produce features on multisided parts with orientation errors on the order of a few tenths of a milliradian.

ACKNOWLEDGEMENTS

I thank my advisor, Dr. John Ziegert, for guiding me in this research and his endless knowledge on every subject imaginable throughout the course of this project. I would also like to thank Dr. Jimmie Miller, Dr. Edward Morse and Dr. Matthew Davies who served on my advisory committee and provided significant additional assistance. I thank Joe Dalton, Joseph Owen and Christopher Tyler for helping with various activities in the machine shop and teaching machining techniques in the machine shop. I thank Mario Valdez for his assistance with anything laser tracker related and his knowledge with Matlab and uncertainty. I thank the UNC Charlotte Center for Precision Metrology for the financial support. I would also like to thank my mother and father supporting me throughout my academic endeavors.

TABLE OF CONTENTS

CHAPTER 1: BACKGROUND AND MOTIVATION	1
Research Motivation	1
Project Scope	3
Literature Review	4
CHAPTER 2: MEASURING COORDINATE SYSTEMS	7
Realizing the Machine Coordinate System	9
Realizing the Part Coordinate System	10
Matlab Post-Processing Algorithm	13
Techniques for placing SMR Nests on a part	16
CHAPTER 3: CAPABILITY OF SUB-SCALE MACHINING	19
Part Series 1 (PS1)	20
Part Series 2 (PS2)	21
Part Series 3 (PS3)	22
Part Series 4 (PS4)	23
Search for Errors	26
CHAPTER 4: VARIABILITY ANALYSIS	32
Variability in Realization of Part and Machine CS with Laser Tracker	32
Variability in Machine Tool Positioning	39

Combined Laser Tracker and Machine Tool Variability	41
CHAPTER 5: EXTENSION TO 3D MACHINING	46
CHAPTER 6: CONCLUSION AND FUTURE WORK	51
REFERENCES	54
APPENDIX A: MATLAB POST-PROCESSING CODE	57
Transforming_g_code.m (Main Script)	57
transform.m	63
tracker_to_part.m	63
tracker_to_machine.m	64
part_to_machine.m	66
LSCircle.m	66
ij_trans.m	67
APPENDIX B: MATLAB POST-PROCESSOR DIRECTIONS FOR USE	69
Measuring machine and part coordinate systems with SA	69
Transforming Part File	75

CHAPTER 1: BACKGROUND AND MOTIVATION

Research Motivation

Industries that produce large products composed of metals show a trend of replacing complicated assemblies with fewer monolithic components [1]. Monolithic components are beneficial over assemblies due to reduced assembly time, reduced part count (reduced inventory costs), improved part strength and often reduced weight while maintaining or increasing part accuracy. Advancements in high-speed machining with significantly increased material removal rates allows for reduced part cycle times, further increasing the cost-effectiveness of monolithic components.

The need for larger monolithic components leads to a need for larger machining centers capable of handling these large components. Normal practice in machining of close tolerance components is to utilize a machine tool whose working volume is larger than the component being machined. As the size of the component increases, larger machines are required, leading to significantly higher costs to acquire, install (due to special foundations) and operate. Due to the high cost of acquisition, redundant capability is rarely available and unplanned downtime can dramatically affect production schedules. The fixture nature of these large machines leads to a reduced overall flexibility in manufacturing operations.



Figure 1: Largest five-axis gantry mill in North America built by Ingersoll Machine Tools. (28 feet wide, 240 feet long, 10 feet under the gantry) [2].

Manufacturing large components presents the challenges mentioned and several more. First, the size of the components make them difficult and hazardous to transport and work with. Second, special metrology methods are typically required as traditional metrology techniques may not be scalable to such a large dimension due to increased cost and decreased accuracy. Large, heavy components have long thermal time constants, potentially leading to large temperature variations in the part and significant dimensional errors. Thermal issues are often a significant source of both machine and part in-accuracy and are the biggest obstacle in precision manufacturing [3]. Sub-scale machining intends to address many of the cost, flexibility and accuracy issues.

Sub-scale machining refers to a process whereby large monolithic components are machined in a sequence of smaller regions by machine tools whose work volume is

smaller than the part, thus requiring that the part, or machine, to be repositioned multiple times to complete the operation. Sub-scale machining can potentially call for the simultaneous operation of multiple machines on a part, reducing cycle time. Sub-scale machining is of potential value to aircraft, automotive, turbine, shipbuilding, and large earth moving equipment industries as they stand to reap the most benefit from the advancement of this manufacturing process. These industries manufacture a wide variety of large components which require precise tolerances to ensure reliable and efficient operation of the assembled systems [4].

Project Scope

The purpose of this study is to a.) develop methods to quickly and easily determine the relative position and orientation of the part and machine coordinate systems following repositioning, b.) develop algorithms to automatically post-process the NC part program to reflect the new position and orientation, c.) determine the achievable accuracy of precision-tolerance machined components using machines having working areas smaller than the completed part, d.) conduct a variability analysis to understand the errors in the process and suggest improvements and e.) explore using sub-scale machining in 3D operations.

A series of parts were machined to determine the basic tolerance holding capability of the machine tool used for the study, and the additional error induced by adding the laser tracker to the manufacturing process. As a final test, a set of parts larger than the working area of the machine were made to determine tolerances the process is capable of achieving and demonstrate the project objectives could be met. A variability

analysis was completed following the NIST Guide to the Expression of Measurement Uncertainty (GUM) to compare and confirm experimental results.

The first phase of the project encompassed development of a post-processing algorithm. The second phase was a proof of concept with two dimensional parts coincident with an analysis of the process to determine areas where improvements can be made. A variability analysis was performed to determine whether errors seen in part features were reasonable. The third and final phase of the project was an analysis of extending the sub-scale machining process into three dimensional parts with multi-side machining.

Literature Review

Recent advancements in portable spatial metrology devices including laser trackers, structured light, fringe projection, photogrammetry, Lidar, and iGPS allow large scale metrology to be performed rapidly and accurately [5]. These recent advancements in spatial metrology coupled with fiducials attached to the part are what makes the sub-scale machining process an effective method of producing large, accurate, monolithic components.

Fiducial based machining and the use of reference features on the part for the purpose of registration and alignment has been used in the past and is used in modern day machining. The semiconductor industry uses fiducials for accurate placement of components on circuit boards and lens manufacturers use fiducials to orient lenses across machines [6]. The SMEMA Fiducial Mark Standard 3.1 is the current standard for the use and design of fiducials in the printed circuit board industry. The standard defines a fiducial as a “printed board artwork feature (or features) that is created in the same

process as the printed circuit board conductive pattern and that provides a common measureable point for the component with respect to a land pattern or land patterns” [7]. The fiducials can be used to compensate for translational and rotational offsets as well as non-linear distortions, such as temperature distortions. Fiducial markers are also used in the map making industry where fiducial marks are placed on the upper surface of the inside cone of an aerial camera [8]. The fiducials are then used for stitching the photographs together to complete the map of an area.

Woody [9] was able to use fiducial markers on monolithic components to correct for thermal distortions and part disorientation which led to more accurate part feature locations. The process Woody described included measuring the location of fiducials in a metrology environment. Then, using an on-machine measurement device, the fiducials were measured in the manufacturing environment. The two (2) fiducial measurements were compared and the NC code was compensated to achieve more accurate part features. Sub-scale Machining uses a similar approach.

In Sub-scale Machining, a laser tracker is used to rapidly measure the spatial relationship between the part and machine CSs following initial positioning and repositioning of the part. Laser trackers use distance measuring interferometry (DMI) or absolute distance measurement (ADM) plus two angle encoders to measure the spatial coordinates of a specially designed target called a Spherically Mounted Retroreflector (SMR). An SMR consists of a hollow corner-cube retroreflector mounted inside of a precision steel sphere with the apex of the retroreflector nominally coincident with the sphere center. Tolerances for concentricity of the sphere and retroreflector radius of curvature and apex is less than 0.00012” [10]. SMRs are normally attached to objects

using specially designed SMR nests that magnetically hold the sphere in a 3-point kinematic mount to ensure repeatable positioning [11]. Laser trackers are a large volume, high accuracy metrology instrument and are widely used in multiple industries. Linear measurement distances of laser trackers are on the order of several tens of meters with uncertainty on the order of tens of micrometers [12]. The laser tracker used during this project has a quoted volumetric uncertainty of $15\text{ }\mu\text{m} + 6\text{ }\mu\text{m/m}$ inside a volume of 2.5 m x 5 m x 10 m [13]. ASME standard B89.4.19 is the standard for performance evaluation tests and geometric misalignments in laser trackers [14].

Laser trackers have been used for the inspection of large mechanical components [15], used to assist in the alignment of optical components and assemblies such as turbine generators and particle accelerators [12], [16] and used for surface measurements of large lenses to assist in the early stages of manufacturing, such as the off-axis lenses for the Giant Magellan Telescope [17].

CHAPTER 2: MEASURING COORDINATE SYSTEMS

To develop the Sub-scale Machining process, a Haas Vertical Milling Machine (VMM), type XYFZ, was used to machine test parts, a Leica laser tracker was used as the stand-alone metrology system, a commercially available CAD and CAM package was used to model and develop tool paths and a Matlab script developed to use as the post-processing algorithm. The test parts were modeled to mimic components that may be used on large earth-moving equipment and were scaled to fit within the capabilities of the UNCC machine shop. A Zeiss tactile CMM was used to measure the features of test parts and determine errors in the feature locations.

The SMR nests are fixed to the part in a pseudo random fashion and remain attached to the part throughout the entire machining process. The SMR nests act as fiducials and provide common measurement points throughout the manufacturing of the component. All measurements were made before any milling began. SMRs were removed from the work piece and the laser tracker covered during milling to prevent stray chips from possibly damaging equipment and to keep coolant mist off critical components.



Figure 2: Typical machine and laser tracker setup.

Before the laser tracker was used, it was allowed to warm up and go through its initializing routine. The laser tracker's weather station instruments were placed so that they measured the conditions of the air path between the laser tracker and machine. Two-faced tests were performed to ensure the laser tracker was operating properly [18] and fiducial locations were checked to ensure a clear line of sight. The Haas VMC was put through a warm-up procedure before beginning milling operations to bring the machine close to a steady operating state.

The laser tracker was used to measure the location and orientation of the part and machine CSs relative to the laser tracker's own internal CS. These measurements were

then used to compute the HTM relating the part and machine CS's. In order to define a CS, the orientation and origin must be known. The part and machine CSs are established by collecting coordinates of a series of points on the part and machine.

Realizing the Machine Coordinate System

To establish the machine's CS, a SMR is fixed to the machine's XY table. The machine is commanded to move in the X axis only throughout the entire X axis range, pausing every two (2) inches to allow a point to be measured by the laser tracker. A point is measured by the laser tracker while the machine is static because static measurements by the laser tracker are more accurate than when measured dynamically [19]. Then, the machine is commanded to move in the Y axis only, pausing every inch to allow a point to be measured. Lines are fit to these points using a least-squares algorithm [20]. The X axis line fit is considered to be the machine's X axis. The Y axis is not perfectly perpendicular to the X line due to imperfections in the machine's construction. The Y axis is therefore used to define the machine's XY plane. The cross product of the unit vectors along the X and Y axes define the machine's Z axis, and a cross product of the Z and X axis unit vectors define the machine Y axis, creating a right handed CS. The spindle location is used as the origin of the machine CS. To establish the location of the machine spindle a fixture was developed to intentionally offset the SMR by approximately 2.5 inches from the center of rotation. The fixture is then rotated in the spindle while the laser tracker records the movement of the SMR. The movement of the SMR nominally represents a circle. A least-squares (LSQ) circle is fit to the points and the origin of the circle is defined as the location of the spindle therefore, the location of the machine CS functional point.

Realizing the Part Coordinate System

The definition of the part's CS is a similar process. At least three (3) fiducial locations are required in order to define the parts orientation and location since there are no movable axes on the part. The placement of the fiducials on the part requires pre-planned locations. The fiducial locations have the requirements of ensuring the completed part will fit on the rough stock, being out of the way of tool paths to prevent tooling collisions, being within the line of site of the laser tracker and staying attached to the part throughout the machining process in order to maintain a consistent definition of the part CS. With magnetic SMR nests, a cover for the nests is useful when machining a ferrous material. Chips build on the nests due the magnet and could cause an undesired false measurement point if a chip remains between the contact point of the SMR and nest. When ensuring the fiducials can be seen by the laser tracker, it is important to consider that the laser tracker is portable and can be moved throughout the process. The planning of the fiducial locations was performed in the CAD/CAM software and extra care was taken to ensure no collision would occur during rapid machine movements. It is important to consider that the paths for rapid movement will change from the rapid movement paths shown in the CAM software since the coordinates of the machine commands will be changed. Setting the rapid path height above any SMR nests and fixturing is recommended. A one to one scale paper template was used place the fiducials on the stock since fiducial placement accuracy of a few millimeters was sufficient. The part's X axis is defined as the vector between fiducial location 1 and 2, where fiducial 1 is defined as the part's origin. The part's Z axis is then defined as the normal vector to the best-fit plane of all the fiducial locations. If and when possible, more than three (3)

fiducial locations are used to reduce error in the definition of the part Z axis [21]. Finally, the cross product of the part Z and X unit vectors defined the part Y axis, creating a right-handed CS. A sample of fiducial layouts on a part is shown in Figure 3.

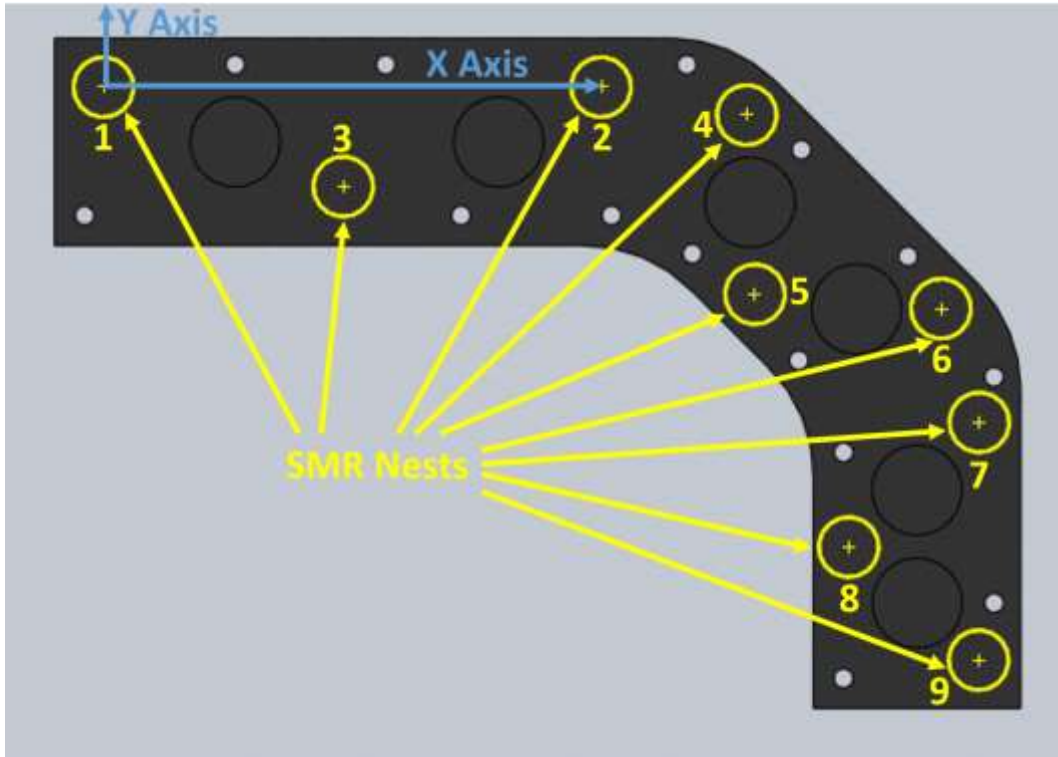


Figure 3: Fiducial locations in yellow with part axis definitions show in blue.

Unit vectors along the axes of the CSs are used to build two homogeneous transformation matrices. One is a HTM to transform from the part to the laser tracker CS seen in Equation 1; and the other to transform from the machine to the laser tracker CS, Equation 2. The two HTMs are then multiplied by to obtain the final HTM that transforms a point from the part to the machine CS, Equation 3. The post-processing algorithm used this HTM to transform points extracted from the NC code defined in the

part CS and define them in the machine CS, Equation 4. A flow chart of the HTM development process is shown in Figure 4.

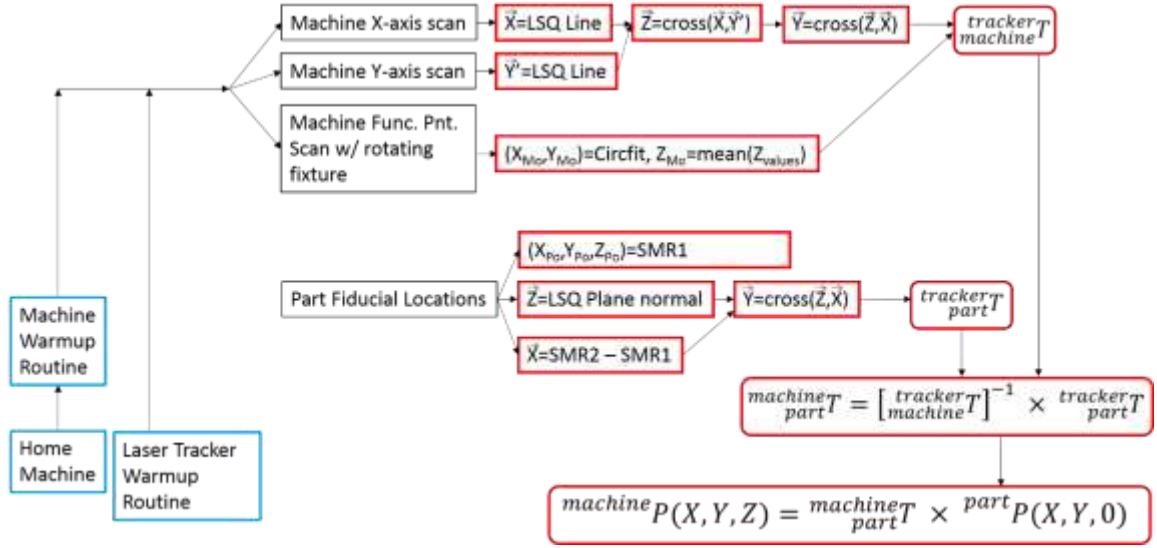


Figure 4: Flow chart of HTM development process.

$$Tracker_{T_{Machine}} = \begin{bmatrix} \widehat{X}_M \cdot \widehat{X}_T & \widehat{Y}_M \cdot \widehat{X}_T & \widehat{Z}_M \cdot \widehat{X}_T & {}^M X_{Origin} \\ \widehat{X}_M \cdot \widehat{Y}_T & \widehat{Y}_M \cdot \widehat{Y}_T & \widehat{Z}_M \cdot \widehat{Y}_T & {}^M Y_{Origin} \\ \widehat{X}_M \cdot \widehat{Z}_T & \widehat{Y}_M \cdot \widehat{Z}_T & \widehat{Z}_M \cdot \widehat{Z}_T & {}^M Z_{Origin} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$Tracker_{T_{Part}} = \begin{bmatrix} \widehat{X}_P \cdot \widehat{X}_T & \widehat{Y}_P \cdot \widehat{X}_T & \widehat{Z}_P \cdot \widehat{X}_T & {}^P X_{Origin} \\ \widehat{X}_P \cdot \widehat{Y}_T & \widehat{Y}_P \cdot \widehat{Y}_T & \widehat{Z}_P \cdot \widehat{Y}_T & {}^P Y_{Origin} \\ \widehat{X}_P \cdot \widehat{Z}_T & \widehat{Y}_P \cdot \widehat{Z}_T & \widehat{Z}_P \cdot \widehat{Z}_T & {}^P Z_{Origin} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$Machine_{T_{Part}} = Tracker_{T_{Machine}}^{-1} * Tracker_{T_{Part}} \quad (3)$$

$$Machine_P = Machine_{T_{Part}} * Part_P \quad (4)$$

Matlab Post-Processing Algorithm

In order to compensate for the random orientation of the parts, an adjusted NC code is needed to machine the nominal part. The Leica laser tracker operated through a software interface called Spatial Analyzer (SA). SA makes the points collected from the laser tracker useful by allowing them to be exported as text files in terms of the laser tracker's Cartesian CS. The Matlab post-processing algorithm or script reads the text files exported from SA and calculates the HTM to transform a point from the part CS into the machine CS. It is important that the units of the point coordinate measurements exported from SA match the units of the machine commands in the NC file.

NC code is typically written for all axis moves nominally described in the part CS assuming the part and machine CSs are aligned. The post-processing algorithm developed here imports the NC code and reads the NC code similar to a machine controller. The NC code rows are read from top to bottom and in each row is read from left to right. The post-processing algorithm looks for and identifies linear interpolation commands, both forms of circular interpolation commands, remembers modal commands such as G90 and G91 (absolute and relative distance commands), understands that commands given after modal commands are given with respect to that mode until another modal command is read. With that information, the post-processing algorithm extracts the motions commands in the axis directions and transforms them into machine coordinates using the HTM calculated dependent upon the mode of the command. Once all coordinate information is transformed into machine coordinates, a new NC file is saved in the directory. The post-processing algorithm also recognizes useful NC code syntax such a

line numbers and comments and writes them to the new NC file to maintain their usefulness.

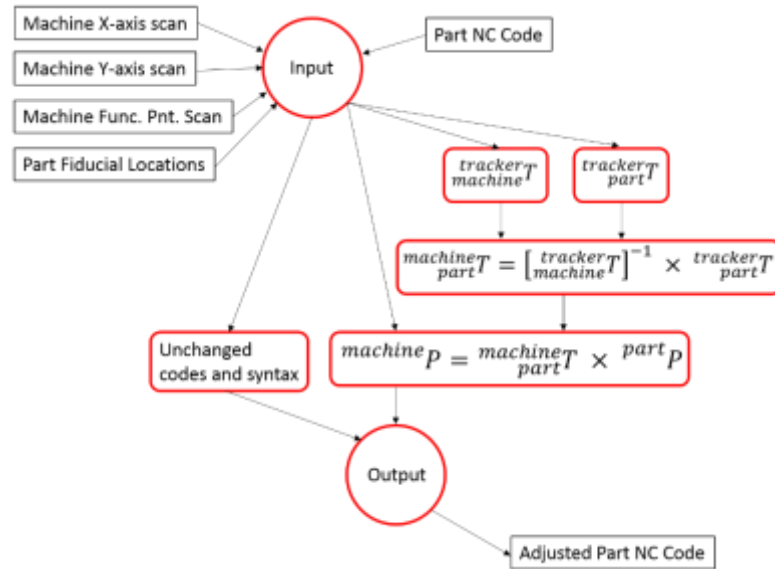


Figure 5: Matlab post-processing algorithm flow chart showing progression of calculations and transformation of part coordinate information.

<pre> % O11111 G00 G90 G54 X0.0 Y0.0 M03 S4000; G43 H01 Z1.0; G01 Z-0.1 F10.; X-8.0; G03 X-9.0 Y-1.0 I0.0 J-1.0; G01 Y-1.5; G03 X-8.0 Y-2.5 R1.0; G91 G01 X8.0; Y1.5; M05; M30; % </pre>	<pre> %; O11111; G00 G90 G54 X-7.0000 Y-5.0000 M03 S4000; G43 H01 Z1.0; G01 Z-0.1 F10.; X-12.6569 Y-10.6569; G03 X-12.6569 Y-12.0711 I0.7071 J-0.7071; G01 X-12.3033 Y-12.4246; G03 X-10.8891 Y-12.4246 R1.0; G01 G01 X5.6569 Y5.6569; X-1.0607 Y1.0607; M05; M30; %; </pre>
*NC code in part coordinates	*NC code in machine coordinates

Figure 6: Original and altered NC code sample showing the part origin (-7,-5) in machine CS and the orientation rotated at 45 degrees CCW.

HTMs allow for point compensation due to all six (6) degrees of freedom (DOF). This causes an issue when working with the Haas VMC because the Haas VMC is a 3-axis machine allowing for compensation in only three (3) DOF; two (2) linear DOF and one (1) rotational DOF. A machine with 5-axis control would be needed to compensate for all six (6) DOF.

In the case of the Haas VMC used, compensation could only be made in the X and Y linear axes and for rotations about the Z axis leaving rotations about the X and Y axes and offsets in the Z axis unaccounted for by the post-processing algorithm. Offsets in the Z axis were managed by manually setting the tool length offsets in the machine controller. Rotations unaccounted for about the X and Y would cause a bias in the length of the parts, causing the parts to always be machined shorter than nominal. Table 1 shows the error that was caused by the rotations about the X and Y axes of the machines. The maximum error shows up when machining Section 3. The magnitude of the errors are considering insignificant compared to the other expected in the part feature locations.

Table 1: Errors caused by rotations about them machine's X and Y axes that cannot be accounted for by the Haas VMC and post-processing alorithm.

		Zpart to Xmach [rad]	Zpart to Ymach [rad]	Zpart to Z mach [rad]	ZX Mag [rad]	Resultant Error over 750 mm [μm]
PS4-1	Sec 1	-0.0001	0.0010	1.0000	0.0010	0.4
	Sec 2	-0.0006	0.0003	1.0000	0.0006	0.2
	Sec 3	0.0023	-0.0034	1.0000	0.0041	6.3
PS4-2	Sec 1	0.0003	0.0008	1.0000	0.0008	0.3
	Sec 2	-0.0004	0.0009	1.0000	0.0009	0.3
	Sec 3	0.0021	-0.0030	1.0000	0.0037	5.0
PS4-3	Sec 1	0.0000	0.0001	1.0000	0.0001	0.0
	Sec 2	0.0002	0.0002	1.0000	0.0002	0.0
	Sec 3	0.0019	-0.0027	1.0000	0.0033	4.1
PS4-4	Sec 1	0.0002	0.0007	1.0000	0.0007	0.2
	Sec 2	0.0005	-0.0001	1.0000	0.0005	0.1
	Sec 3	0.0025	-0.0027	1.0000	0.0037	5.1

Techniques for placing SMR Nests on a part

For this project, SMR nests are placed on the part stock in a pseudo-random fashion and act as the fiducials for laser tracker measurements. There are many factors when choosing the number of fiducials to use and the location of the fiducials. Many fiducials are desirable as the measurement of fiducials allows the part surface form to be preserved and reduce the chance of a measurement outlier causing undesirable error in part feature location through an averaging affect. Although, many fiducials can be time and monetarily prohibitive. Fewer fiducials are cheaper and require less time to measure but increase the chances of a measurement outlier causing undesirable errors. Experienced judgment is used to determine the number of fiducials required.

In sub-scale machining all fiducials have an effect on the realization of the part CS but fiducial [SMR] 1 & 2 have the largest effect. Fiducial 1 defines the part CS origin and the vector between Fiducial 1 & 2 determines the orientation of the part CS about it's X axis and the others define the part Z vector. Misalignment of the part's X vector and

origin can cause the nominal part form to be outside the rough stock allotted for the machining operation. When considering how to place the fiducials 1 & 2 on the part, the technique should be matched to the amount of rough stock that has been allotted around the nominal profile of the part. For Part Series (PS) 1 – 3, a paper template constructed from a 1-to-1 scale drawing plot and was used to place the all SMR nests on the part stock. The nominal part shape for PSs 1 - 3 was approximately 18” long and approximately 1” of stock was left all around the nominal finished part form. This required the SMR nests to be placed within a few millimeters of their planned location. For PS4, the part was significantly longer and less rough stock was allotted around the perimeter of the finished form. The placement accuracy of SMR nest 1 & 2 for PS4 were much more critical to ensure the nominal part fit on the rough stock. A fixture to place SMR nests 1 & 2 was machined to place the SMR nests within a millimeter or less of their nominal position, as shown in Figure 7. The remaining nests were placed with a paper template because their main requirement was to be out of the way of tool paths. A fixture of similar purpose could be constructed for any machining setup.



Figure 7: Fixture used for placing SMR nest 1 & 2 on PS4.

Aspect ratio of fiducial locations is important in realizing the part CS. Ideally fiducials should be placed as far apart as possible. Placing all fiducials in a line should be avoided, as the aspect ratio for a line approaches zero. Fiducials placed in a line could result in error in the realization of the part's CS. For PS4, the fiducials were arranged on the part with three (3) fiducials in each section and so that at least six (6) fiducials were directly above the machine tool's table and the aspect ratio was approximately one-to-one.

CHAPTER 3: CAPABILITY OF SUB-SCALE MACHINING

A series of parts, PS1 - PS4, were machined in order to quantify the tolerance capability of the sub-scale machining process. The series of parts began with a traditional machining setup where the part and machine axes are aligned manually. Each step incorporated a small step towards the completed sub-scale machining process. The final part series was a set of parts machined using the post processing algorithm, and that were larger than the machine's working area to show proof of concept. Figure 8 shows the design of the parts. One corner of each part was used to create the part datum, and all features were measured relative to those datum surfaces. CMM errors are considered insignificant as the errors being analyzed are larger than the expected measurement uncertainty of the CMM.

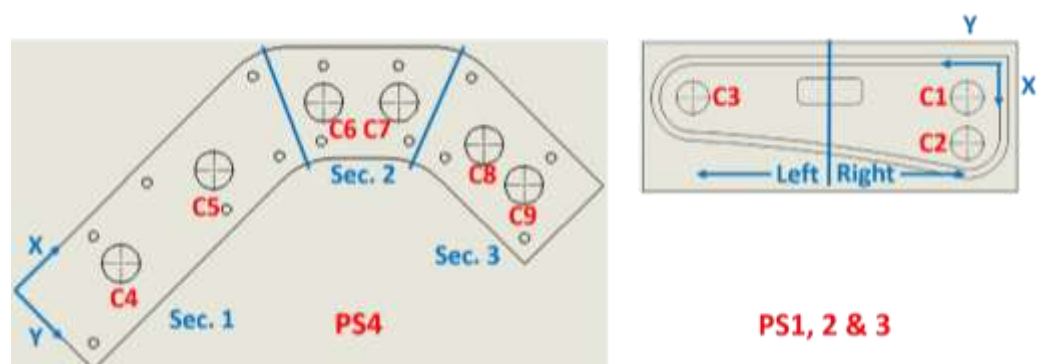


Figure 8: Schematic showing parts used to determine feature errors. The left diagram was the part used for PS4 and the right diagram was the part used for PS1, PS2, & PS3.

Part Series 1 (PS1)

PS1 consisted of five (5) parts and was machined traditionally where the part and machine axes were mechanically aligned and the part was machined in one fixturing each. The errors of features in PS 1 describe the capabilities of the Haas VMM and was used as a baseline for determining whether or not adding the laser tracker to the machining process increased the magnitude of errors in the part feature locations. A maximum true position error of approximately 100 micrometers for PS1 describes the approximate capability of the Haas VMM, as shown in Figure 9. This value is likely an over-estimate as the proper tooling and machining parameters were still being altered during this series to determine the best combination.

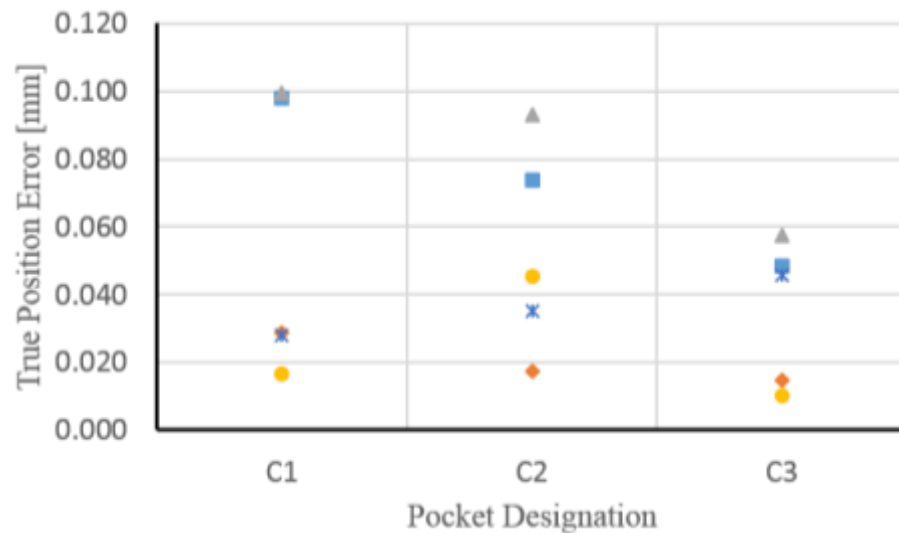


Figure 9: PS1 pocket true position error.

Part Series 2 (PS2)

PS2 consisted of ten (10) parts and was the first set of parts to incorporate the laser tracker into the machining process. PS2 was machined in one fixturing but was arbitrarily placed within the machine's working area. Fiducials placed on the part allowed the laser tracker to measure its orientation and the post-processing algorithm was used to alter the NC code to account for the misalignment of the part and machine axes. No increase of errors of part features is expected in PS2 due to the entire part being machined in one fixturing. PS2's errors are on the same order of magnitude as PS1 with the exception of one part which is considered an outlier, Figure 10. This shows that the laser tracker does not introduce additional error into the process if the part is machined in one fixturing.

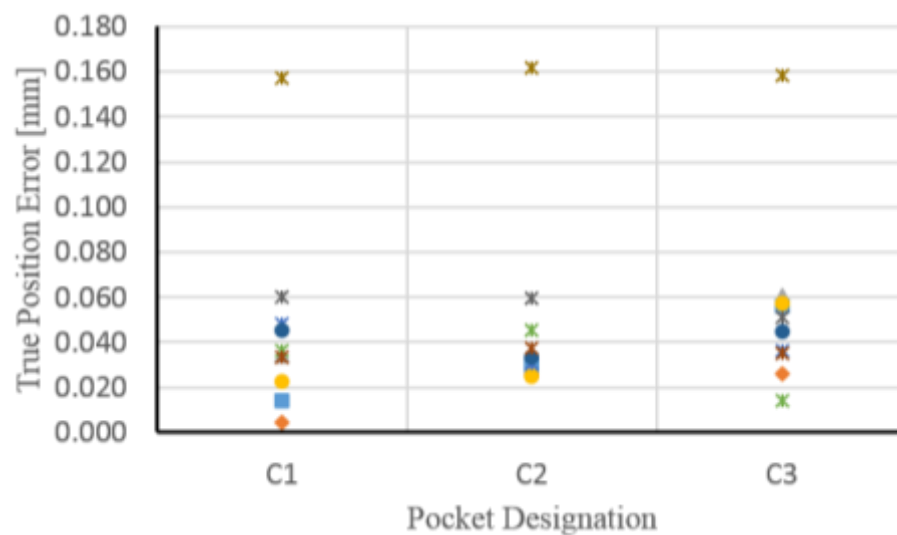


Figure 10: PS2 pocket true position error.

Part Series 3 (PS3)

PS3 consisted of five (5) parts and was the first set of parts to be machined in multiple fixturings. The same geometry and features were used as PS1 & PS2 but the part was split approximately in half and each half machined in separate fixturings. The NC code was split into two programs, one for the right and one for the left portion of the part as defined by the blue line in Figure 5. The right portion of the part was milled first and also creates the datum surfaces used to define the part's CS when measured on the CMM. Since pockets C2 & C3 were machined in the same fixturing as the features used to define the part's CS on the CMM, only the HAAS machine's errors should contribute to their positioning errors. The errors from "re-finding" the part's orientation after the second fixturing will show up in pocket C3. Pockets C1 & C2 show a true position error approximately equal to the pockets on PS1 & PS2, as expected. Pocket C3 shows a significant increase in true position error that is much larger than expected based on the manufacturer's estimates of the laser tracker accuracy, Figure 11.

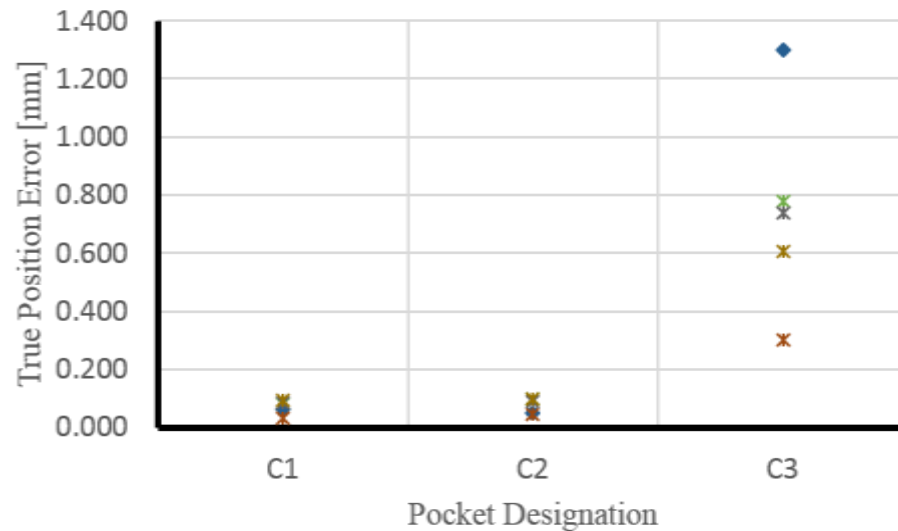


Figure 11: PS3 pocket true position error.

Part Series 4 (PS4)

PS4 consisted of four (4) parts and was the first set of parts to be machined that was not capable of fitting within the machine's working area. Similar to PS3, PS4 was machined in multiple sections but this time the part was split into three (3) sections instead of two (2). PS4 was the proof of concept that sub-scale machining is possible. An increase in errors of part features is expected in PS4 since the part is machined in three separate fixturings. Figure 12, Figure 13 and Figure 14 show the approximate orientation and amount of part hangover when mounted to the Haas VMC. It was a concern that overhanging portion of the part when machining section three (3) would affect the realization of the part CS. A Solidworks FEA simulation shows the maximum deflection of the largest overhanging portion to be approximately 100 μm . It was determined that this over hanging portion should have little effect on the realization of the part CS. The

intent of the multiple fiducial locations was to minimize any errors caused by overhanging portions or not flat parts.

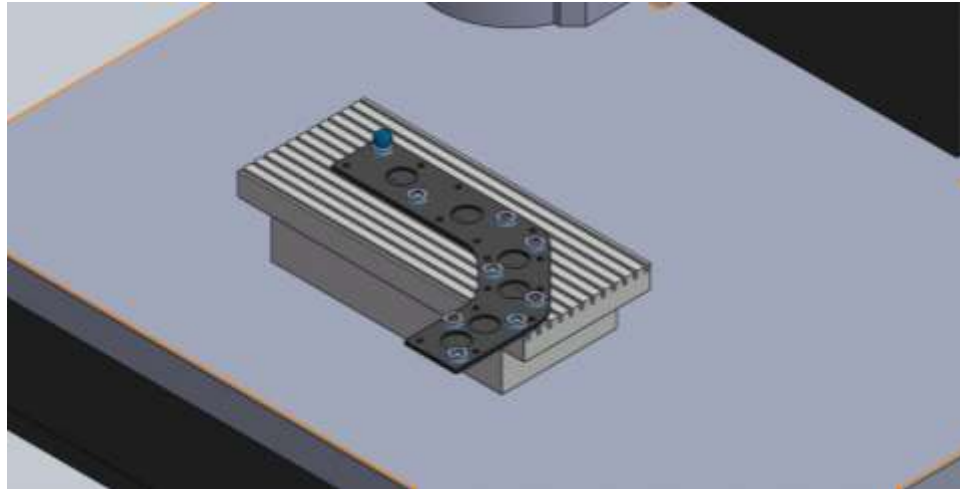


Figure 12: Approximate orientation and location of PS4 when milling section 1. The amount of overhang in the position is approximately 8.5 inches.

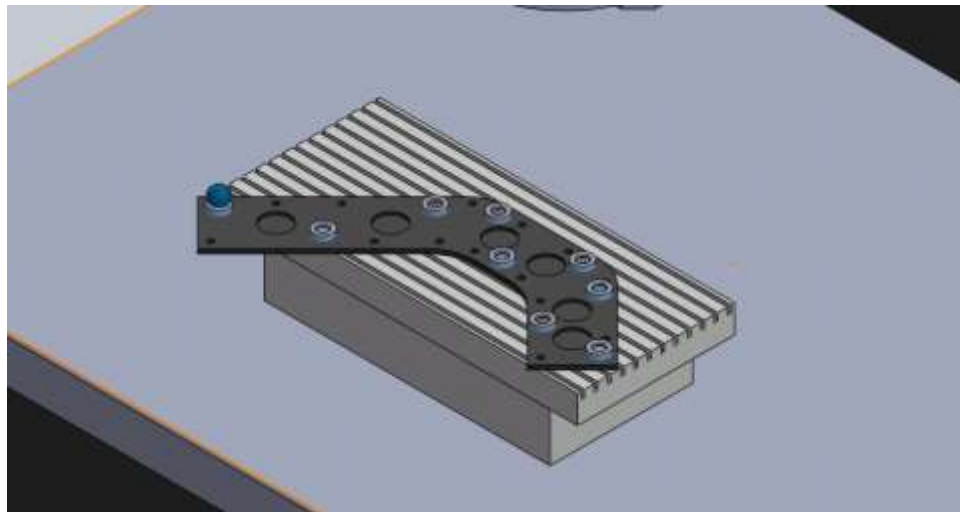


Figure 13: Approximate orientation and location of PS4 when milling section 2. The amount of overhang in the position is approximately 11.5 inches.

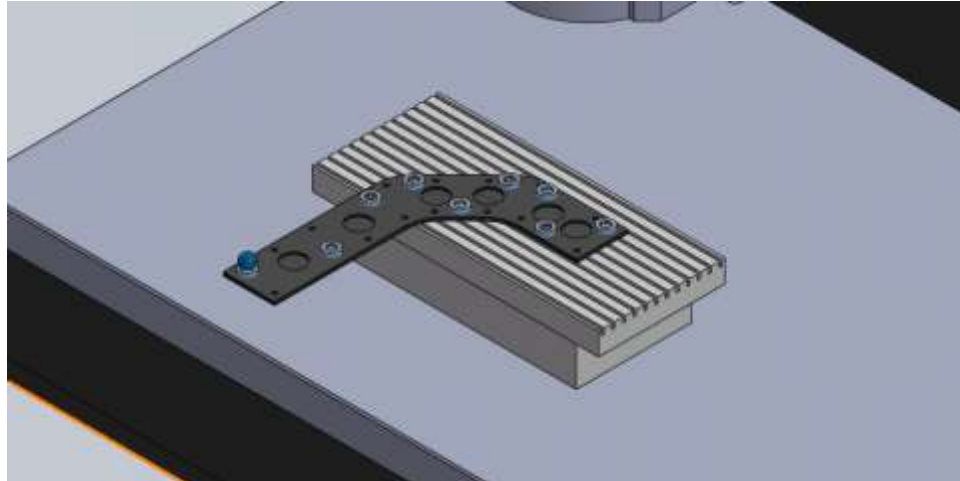


Figure 14: Approximate orientation and location of PS4 when milling section 3. The amount of overhang in the position is approximately 16.5 inches.

Figure 15 shows the true position errors for the pockets for PS4. Pockets C4 & C5 were milled in the same portion of the part as the features that were used to define the origin on the CMM, showing a true position error approximately equal those in PS1. Pockets C6, C7, C8, & C9 show an increase of 4 times in true position error from “re-finding” the part in the second and third fixturings.

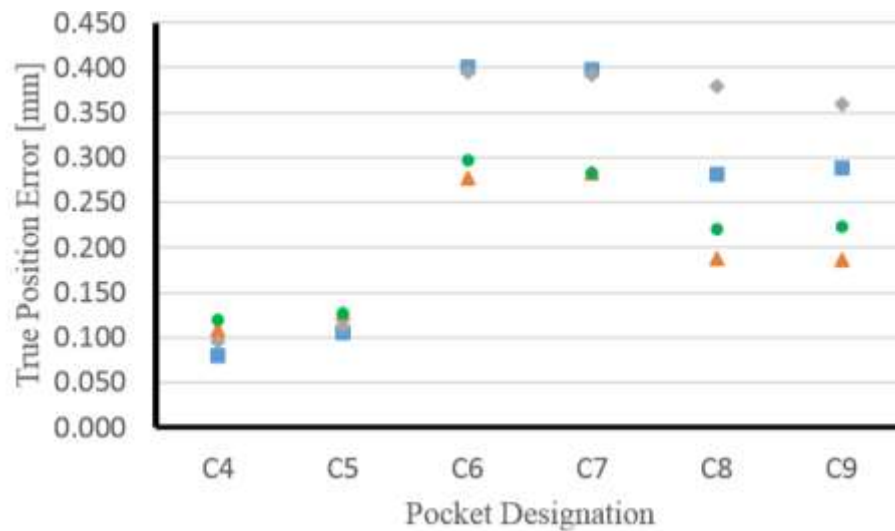


Figure 15: PS4 pocket true position error.

Search for Errors

The large errors seen in the pockets machined after re-fixturing prompted a search for their source. In order to determine other possible sources of error, several checks were made. Repeatability tests were performed to check the performance of the Haas VMM's homing capability, a repeatability test was performed on the fixture used to find the machine's spindle location and machine axis squareness measured. It was determined that the Haas VMM used to produce the test parts had an X-Z axis squareness error of approximately 50 arc seconds causing the angle between the X and Z axes to be acute. Previous measurements of the machine spindle to find the machine origin were taken with the machine's Z axis in the upper limits of the range for convenience, approximately 430 millimeters above the plane which the milling of the part took place. When the spindle location is measured at the upper extreme of the Z axis and the spindle lowered for milling, the spindle shifts approximately -100 μm in the machines X direction, Equation 5. It was determined that all measurements of the machine and part CS should

be taken as close to the same plane as possible to avoid any effects from machine squareness errors.

$$430 \text{ mm} \times 50 \text{ arcsec} = 104 \text{ } \mu\text{m} \quad (5)$$

Since the spindle location was measured with the Haas VMC's Z axis in the same position when the sections of PS4 were machined, the vector in which the error occurred can be subtracted out of the pocket locations as measured by the Zeiss CMM. When the error vectors are subtracted from the measured values, the position error of the pockets on PS4 more closely align with the expected magnitude of part feature errors, Figure 16.

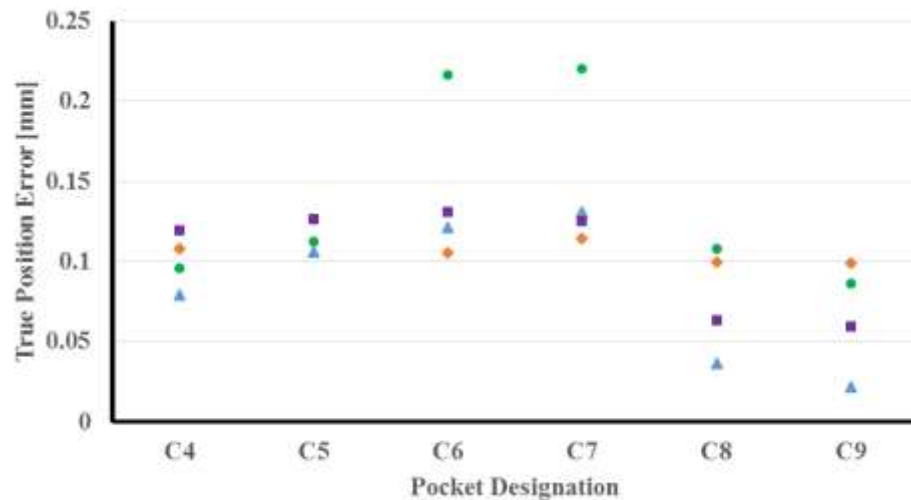


Figure 16: PS4 true position errors after subtraction of 100 micrometer error caused by X-Z squareness error.

To avoid the timely process of re-machining PS4 to determine the error in the process after realizing and finding a solution to the significant geometry errors in the Haas VMC, SMR nests were attached to one of the parts from PS4 and the location of the part features in the part CS were measured on the Zeiss tactile CMM, creating an artifact. Precision spheres simulated the SMRs while being measured on the Zeiss to allow proper

measurement of the fiducial and part feature locations. The CS the part features were related to are the same as the part CS as described in Chapter 3. This allowed the pockets to be found in the machine CS using the post-processing algorithm and probed using a dial indicator mounted in the machine spindle. Circular run-out (CR) was used as the metric to determine positioning accuracy.

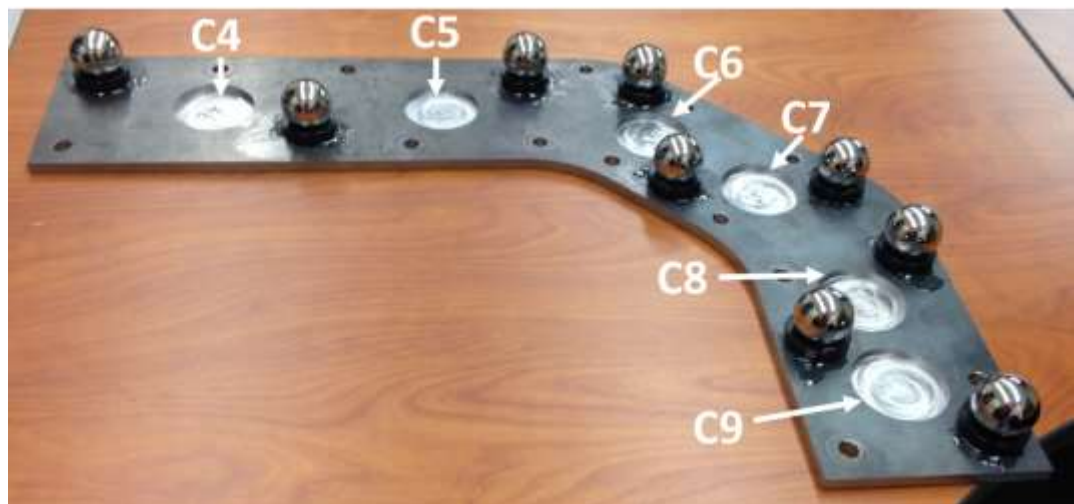


Figure 17: PS4 part artifact with 1.5" precision spheres to replace the SMR while being measured on Zeiss CMM.

With all the CS measurements being taken in approximately the same plane, CR values when tramming pockets on the Haas VMC were still a factor of 4 higher than expected. Table 2 shows the values for three (3) trials when attempting to locate the part artifact pockets in the machine CS with a dial indicator on the Haas VMC.

Table 2: CR values for three (3) trials when measuring part artifact pocket locations on Haas VMC.

	Trial 1 CR [mm]	Trial 2 CR [mm]	Trial 3 CR [mm]
C4	0.152	0.152	0.152
C5	0.127	0.127	0.127
C6	0.229	0.203	0.229
C7	0.203	0.178	0.203
C8	0.406	0.406	0.406
C9	0.356	0.381	0.381

The material used for PS4 released residual stresses when machined causing an undesired flatness deviation of approximately 400 μm . The flatness of the material affects the definition of the part CS. The incorporation of more than three (3) fiducial locations was intended to increase accuracy in situations like this. Although in the case of the machining of PS4, only portions of the part were fixtured and held flat during the machining of each section. This caused the part shape and the location of the fiducials to change with the machining of each section. Therefore, the definition of the part CS was different each time. Experiments have shown that if Section 1 isn't properly fixtured, the definition of the part CS becomes flawed causing a significant error in the definition of the feature locations when transformed in to the machine CS. Section 1 contains the two most critical fiducial locations. The fiducial that defines the part origin, SMR1, and the fiducials that define the part CS X axis, SMR1 & SMR2. To confirm, the part artifact was fixtured to a machine tool where the entire part fit within the machine's working area and the part pulled "flat" to the machine's table. When finding the part features in the machine CS, a maximum of 125 μm of CR was measured. On the same machine, without section 1 properly fixtured, CR values of up to 400 μm were measured as seen in Table 3.

Table 3: Pocket CR with and without having Section 1 of PS4 properly fixtured.

	C4	C5	C6	C7	C8	C9
Sec 1 not fixtured CR [mm]	0.178	0.152	0.305	0.279	0.381	0.356
Sec 1 fixtured CR [mm]	0.051	0.025	0.051	0.076	0.127	0.127

To understand the difference between the part when it is clamped “flat” and when it is in its free state, a repeatability test was performed where the Section 3 remained fixtured to the table and Section 1 was fixtured flat and then left in a free state. The difference in the two measurements arise when the HTM from the part to machine is examined. Table 4 shows the part to machine HTM for when the part is held “flat” and when only Section 3 is fixtured. There is an 86 μm difference in the origin of the part CS. This leads to an approximately 125 μm difference in pocket location in the machine CS as shown in Table 5.

Table 4: Comparison of properly clamped part to machine HTM and not properly clamped part to machine HTM.

Trial 1 Partially Fixtured				Partially - Fully Fixtured		Trial 2 Fully Fixtured			
					MAG [mm]				
0.017	-1.000	0.002	-31.332	0.076	0.086	0.017	-1.000	0.001	-31.408
1.000	0.017	-0.001	-213.740	-0.041		1.000	0.017	-0.001	-213.699
0.002	0.002	1.000	-100.251			0.001	0.001	1.000	-111.504
0	0	0	1			0	0	0	1
Trial 3 Partially Fixtured					MAG [mm]				
0.017	-1.000	0.002	-31.331	0.077	0.087				
1.000	0.017	-0.001	-213.740	-0.041					
0.002	0.002	1.000	-108.472						
0	0	0	1						
Trial 4 Partially Fixtured					MAG [mm]				
0.017	-1.000	0.002	-31.333	0.075	0.088				
1.000	0.017	-0.001	-213.746	-0.046					
0.002	0.002	1.000	-103.313						
0	0	0	1						

Table 5: Comparison of Pocket 5 & 6 transformed coordinates.

C8							
Fully Fixtured		Partially Fixtured		Partially - Fully Fixtured			
X [mm]	Y [mm]	X [mm]	Y [mm]	X [mm]	Y [mm]	MAG [mm]	
247.636	328.594	247.769	328.523	0.133	-0.071	0.151	
		247.747	328.535	0.111	-0.059	0.126	
		247.736	328.535	0.100	-0.059	0.116	
C9							
Fully Fixtured		Partially Fixtured		Partially - Fully Fixtured			
X [mm]	Y [mm]	X [mm]	Y [mm]	X [mm]	Y [mm]	MAG [mm]	
323.827	327.478	323.960	327.399	0.133	-0.079	0.155	
		323.938	327.415	0.111	-0.063	0.128	
		323.927	327.416	0.100	-0.063	0.118	

CHAPTER 4: VARIABILITY ANALYSIS

Sub-scale machining works by transforming points from one CS to another through a series of measurements of each CS with a stand-alone metrology system. To understand the errors involved in the sub-scale machining process a variability analysis was performed. The variability analysis used techniques for quantifying expected variations similar to an uncertainty analysis following techniques outlined in the NIST JCGM 100:2008 Guide to the Expression of Uncertainty in Measurement (GUM) [22]. There are two (2) main variation contributors analyzed. The variation contributors include measurements made by the laser tracker which leads to errors in the realization of the part and machine CSs and the machine's ability to position its spindle.

Variability in Realization of Part and Machine CS with Laser Tracker

As described in previous chapters, a Leica laser tracker was used to measure points on the machine and part that are used to describe the machine and part CSs. These points were fit to substitute geometry such as planes, lines, and circles in order to describe the location and orientation of certain features of the part and machine. The calculated location and orientation of these features were used to calculate the transformation matrix that would allow points to be transformed from the part CS to the machine CS despite random orientation and location of the part.

Due to the complicated nature of the process used to realize the part to machine HTM and the large possibility of missing correlations, a Monte Carlo (MC) simulation

was developed to study the variation of the transformed part feature locations. Inputs into the MC simulation were a real measurement scenario with points taken in the machine shop with random variation added to each point collected by the laser tracker [23]. The MC simulation was iterated one (1) million times and followed the same mathematical procedures of the port-processing algorithm. The MC output described the standard variation of the transformation of part features to machine CS [23]. A flow chart of the Monte Carlo process is shown in Figure 18.

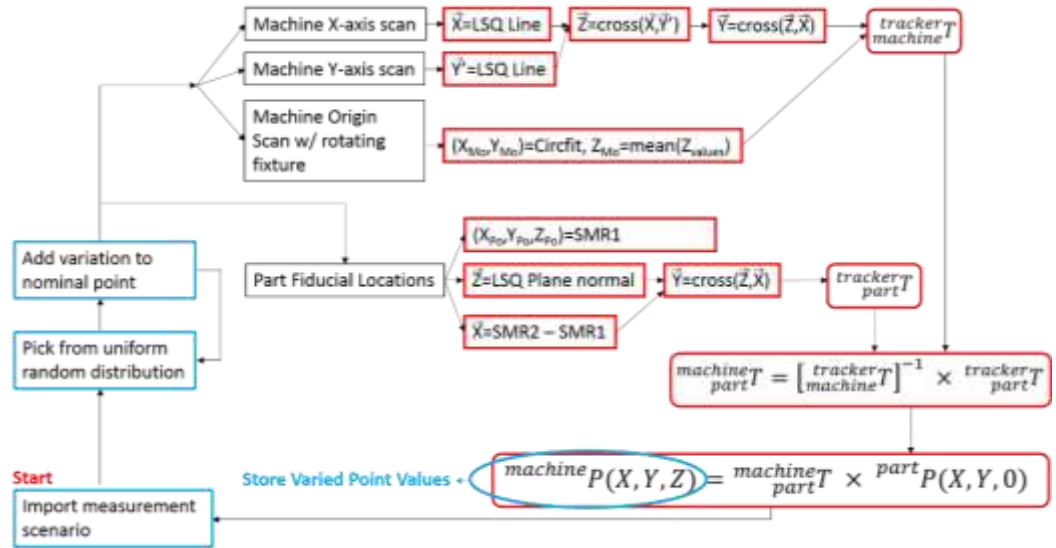


Figure 18: Monte Carlo simulation flow diagram.

A random uniform distribution $50/\text{SQRT}(3)$ micrometers wide was applied to the X, Y & Z values of each point collected throughout the measurement process. The output of the MC simulation yielded an approximate Gaussian distribution for the X and Y error vectors where the standard deviation is equal to the 68% of the expected variation, see Figure 19. The covariance and correlation coefficient was calculated to determine if the X

and Y variations were correlated. The definitions of standard deviation, covariance and correlation coefficient are described in Equations 6, 7 & 8 respectively. The MC simulation was repeated for each pocket in PS4 as each part feature was expected to see different variation. Variance (σ_x^2) for the X and Y coordinates in terms of the part CS were calculated for each pocket along with covariance (σ_{xy}) and correlation coefficient (r). A correlation coefficient close to one (1) or negative one (-1) shows a positive or negative linear correlation and a correlation coefficient close to zero shows little to no correlation. The results from the Monte Carlo simulation are shown in Table 6, where “Part Origin” is the machined feature on the part measured by the tactile CMM.

$$\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (6)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad (7)$$

$$r = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \quad (8)$$

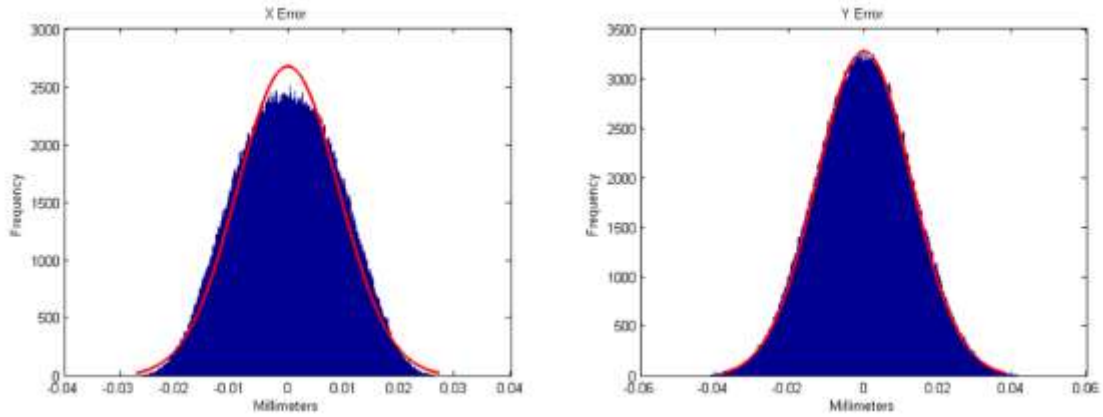


Figure 19: Histogram of X (left) and Y (right) values of the part CS origin in the machine CS as determined by a Monte Carlo simulation. The lines are the calculated best fit histograms as determined by Matlab.

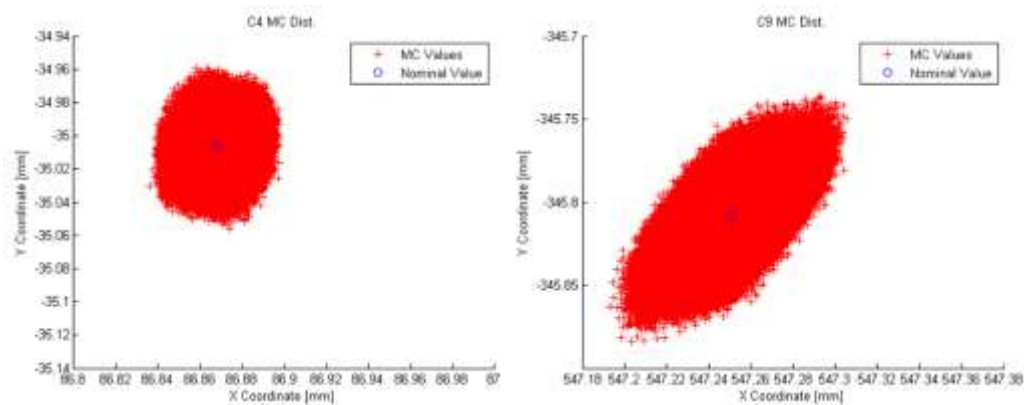
Table 6: Results from the Monte Carlo simulation showing X axis and Y axis variance for each pocket with XY covariance and correlation coefficient.

	Laser Tracker Var.	LT XY CoVar.	LT XY Cor. Coef.
	$\sigma^2 [\mu\text{m}^2]$	$\sigma_{XY} [\mu\text{m}^2]$	r
Part Origin X	84	13	0.10
Part Origin Y	178		
C4 X	83	-1	-0.01
C4 Y	132		
C5 X	83	7	0.06
C5 Y	140		
C6 X	89	31	0.21
C6 Y	233		
C7 X	111	74	0.41
C7 Y	297		
C8 X	183	151	0.60
C8 Y	340		
C9 X	250	193	0.66
C9 Y	341		

The results from the Monte Carlo simulation show that the X and Y components of the error vector are correlated. The correlation of the X and Y vectors change from pocket to pocket and become more correlated as the pocket becomes further from the part

CS origin. The correlation between the X and Y components become obvious when the errors are plotted. The increase in correlation can be seen when comparing the plot of pocket C4 that has a correlation coefficient near zero to pocket C9 which has a correlation coefficient of 0.66, Figure 20.

Figure 20: Pocket C4 Monte Carlo results shown on left with minimal correlation and Pocket C9 Monte Carlo results on right showing correlation.



To confirm the MC simulation, ten measurements of the part CS origin in the machine's CS were performed and compared to the output from the MC simulation. The expanded variation in X was determined to be ± 17 micrometers and the variation in Y was determined to be ± 25 micrometers at a 95% confidence interval ($k = 2$), as determined by the MC simulation. The results of the repeated tests are shown in Figure 21.

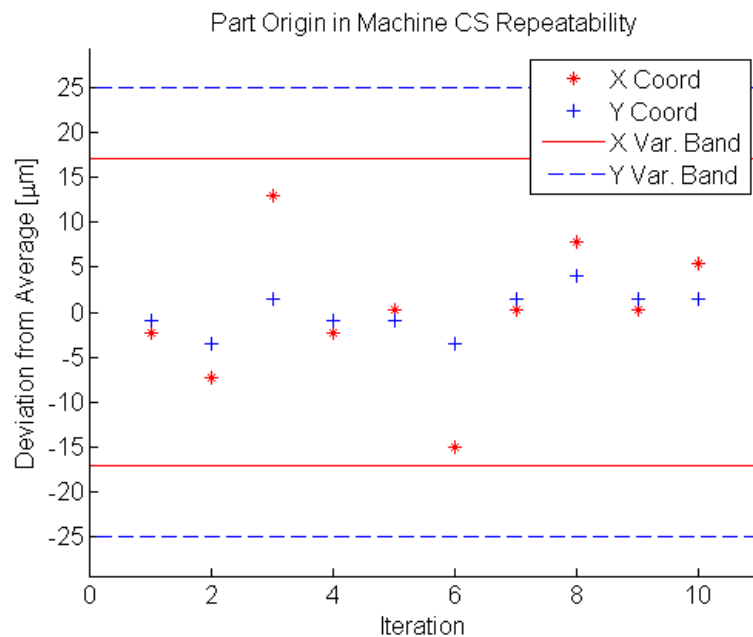


Figure 21: Part in machine CS repeatability results.

The error vectors were analyzed in terms of magnitude and direction as well. It was predicted that the correlation between the X and Y error vectors would arise in the histogram of the error vector orientation. Figure 22 & Figure 23 show the histograms of the error magnitude and direction for pocket C9 determined from the Monte Carlo simulation. Two (2) peaks that are π radians apart in Figure 23 show that error vectors have a tendency to occur most often in those directions.

Since error magnitudes of vectors cannot be zero, the histogram of the error vector magnitudes resembles a Rayleigh distribution [24]. A Rayleigh distribution is a continuous probability distribution function for positive, random values. The typical methods for calculating the variance of a Rayleigh distribution require the assumption

that X & Y values be normal, of equal variance and uncorrelated. The X & Y values for the Monte Carlo simulation have been shown to be approximately normal but not of equal variance and uncorrelated. Therefore, the variance of the error magnitude is calculated from a best fit Rayleigh distribution using the Matlab function `std(fitdist(R,'Rayleigh'))` where R is the row vector of error magnitudes.

Figure 22: Histogram of Pocket C9 error magnitudes determined from Monte Carlo simulation with best fit Rayleigh profile.

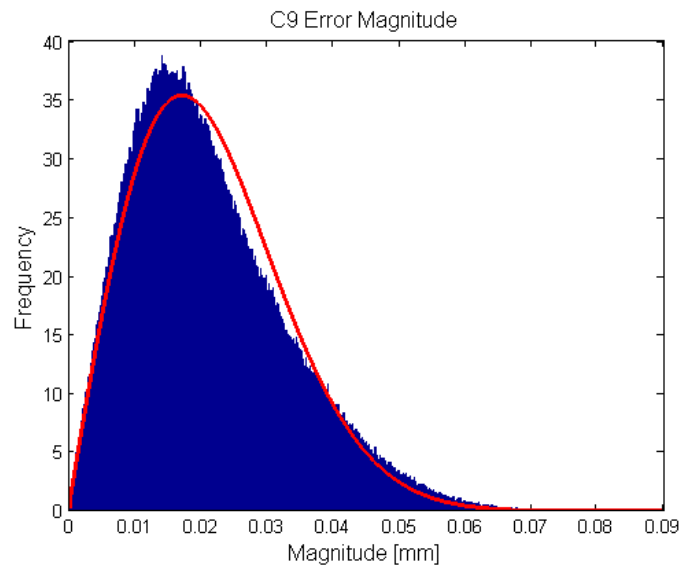
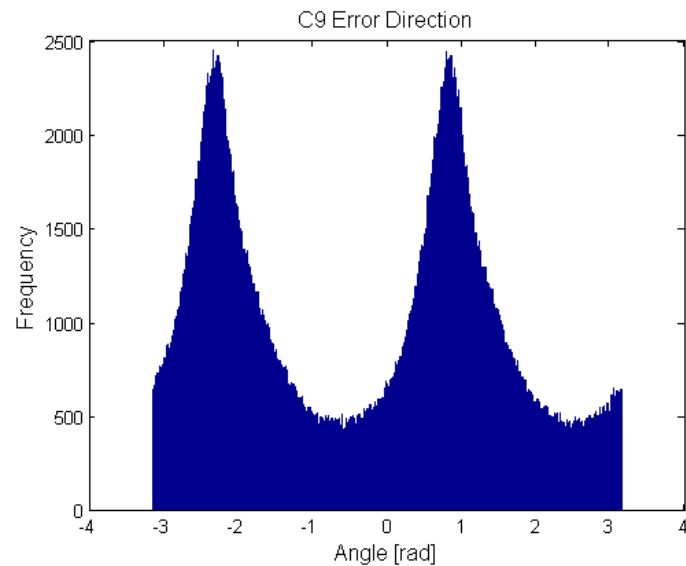


Figure 23: Histogram of Pocket C9 error directions determined from Monte Carlo simulation.



Variability in Machine Tool Positioning

The second major variability contributor to sub-scale machining is the ability of the machine tool to position its spindle and create features. A “Type A” evaluation of the Haas VMM used in this project was performed by machining a set of parts using conventional machining methods with machine and part axes aligned, PS1. The parts were then measured on a Zeiss bridge-type CMM [22]. The uncertainties in the measurement of the part features by the Zeiss CMM were considered insignificant, thus ignored. Figure 24 shows the X errors in the part features and Figure 25 shows the Y errors in the part features.

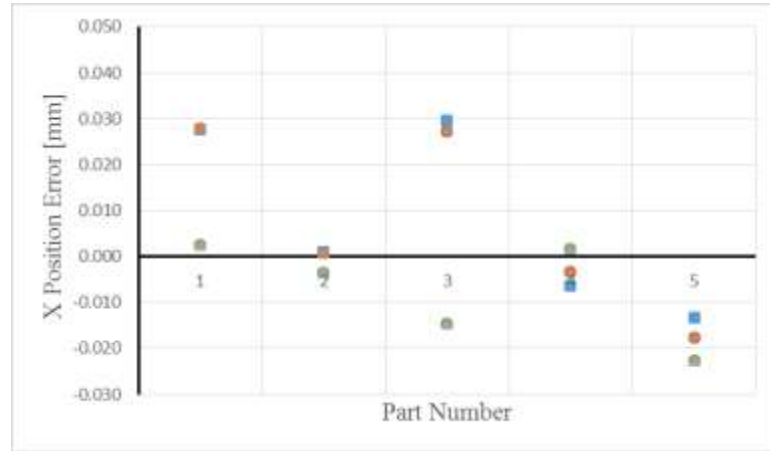


Figure 24: X position errors of parts machined for “Type A” variability analysis of machine positioning.

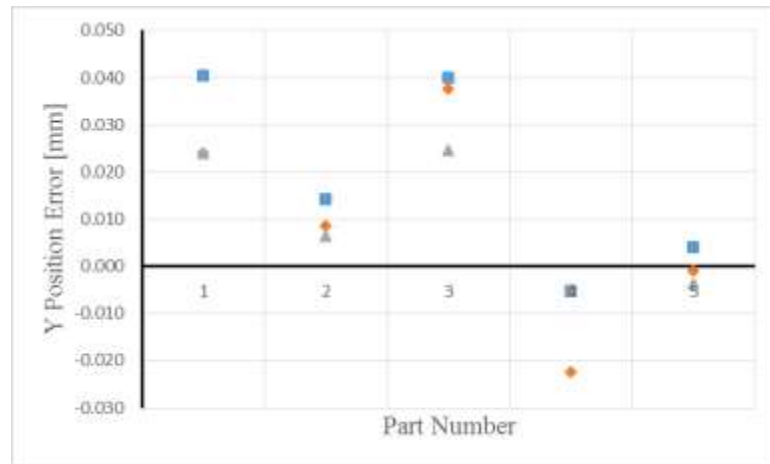


Figure 25: Y position errors of parts machined for “Type A” variability analysis of machine positioning.

The estimate of the population variance of the errors for the X axis is 361 micrometers² and 324 micrometers² for the Y axis calculated using Equation 6. The variance of the error equals the variability in the data set [25]. The covariance of the

machine X and Y errors is determined to be 169 micrometers² resulting in a correlation coefficient of 0.70. The variability in the machine tool is assumed to be constant throughout the working area of the machine. The error vectors were converted to magnitude and orientation and the statistical methods used to calculate the mean and variance of the MC simulation were used to calculate the mean and variance of the machine tool errors. The machine tool error magnitude mean was determined to be 24 micrometers with a variance of 169 micrometers².

Combined Laser Tracker and Machine Tool Variability

When a part feature is machined or located using the sub-scale machining process, two major variability contributors are the variability from the laser tracker realizing the part and machine CSs and the machine tool positioning variability. To calculate error values in which to compare with experimental values obtained in the machine shop, the mean error magnitudes of the laser tracker (q_{LT}) and machine tool (q_M) were combined to achieve the mean error magnitude (q), Equation 9. The variance of the laser tracker and machine tool add to achieve the variance of the total mean error magnitude (σ_q^2) assuming the machine tool and laser tracker error are uncorrelated, Equation 10 [26]. The total mean error magnitude is then multiplied by two (2) to obtain the error's equivalent in terms of CR (q_{CR}), Equation 11. The variance of the equivalent CR ($\sigma_{q_{CR}}^2$) is equal to twice the variance of the total mean error magnitude variance.

$$q = q_{LT} + q_M \quad (9)$$

$$\sigma_q^2 = \sigma_{q_{LT}}^2 + \sigma_{q_M}^2 \quad (10)$$

$$q_{CR} = 2q \quad (11)$$

$$\sigma_{q_{CR}}^2 = 2\sigma_q^2 \quad (9)$$

Table 7: Laser tracker and machine tool error vector magnitude mean and Rayleigh variance.

	LT Error Mag. Mean	LT Error Mag. Var.	Mach. Error Mag. Mean	Mach. Error Mag. Var.
	$q_{LT} [\mu m]$	$\sigma_{q_{LT}}^2 [\mu m^2]$	$q_M [\mu m]$	$\sigma_{q_M}^2 [\mu m^2]$
Part Origin	15	56	24	169
C4	13	46	24	169
C5	13	48	24	169
C6	16	69	24	169
C7	18	88	24	169
C8	20	112	24	169
C9	21	127	24	169

Table 8: Total error magnitude mean, total error magnitude variance, equivalent CR mean and equivalent CR variance.

	Sum LT & Mach. Mean Error Mag.	Sum LT & Mach. Mean Error Equiv. CR	Sum LT & Mach. Mean Error Var.	Sum LT & Mach. Mean Error Equiv. CR Var.
	$q [\mu m]$	$q_{CR} [\mu m]$	$\sigma_q^2 [\mu m^2]$	$\sigma_{q_{CR}}^2 [\mu m^2]$
Part Origin	38	77	225	450
C4	37	74	215	430
C5	37	74	217	434
C6	40	79	238	476
C7	42	83	257	513
C8	44	88	281	563
C9	45	90	296	591

Table 9: Mean equivalent CR and measured CR with respective 95% confidence intervals.

	Sum LT & Mach. Mean Error Equiv. CR	Error Mag. Equiv. CR 2*STD.	Measured CR Av.	Measured CR 2*Std.
	$q_{CR} [\mu m]$	$2\sigma_{qCR} [\mu m]$	$CR_{meas} [\mu m]$	$2\sigma_{CRmeas} [\mu m]$
Part Origin	77	42	N/A	N/A
C4	74	41	102	72
C5	74	42	70	77
C6	79	44	76	41
C7	83	45	63	29
C8	88	47	89	66
C9	90	49	108	49

To make the variability values comparable to a metric that was measured in the machine shop, the variability values were converted into an equivalent CR assuming the form error of the circular pocket is much less than the position error of its center. CR measures cumulative values of coaxiality, and form error [27]. In this experiment, the coaxiality of the machine spindle and circular pocket axes are measured with the circularity of the pocket unseparated, within the same measurement. In an ideal situation with zero form error/circularity, CR will only measure twice the position error. The PS4 artifact has circularity values of approximately 25 micrometers. Therefore, the circularity could add or subtract 25 micrometers to the measured CR values.

With proper measurement technique and part fixturing, all pockets on the PS4 artifact were measured four 4 times to achieve spread of CR values to compare to the variability analysis results for equivalent CR. The four 4 measured CR values were averaged and the sample variance calculated, Table 8. The standard deviation of the equivalent CR and measured CR values were calculated where the standard deviation is equal to the square root of the variance. The 95% confidence interval of each value was calculated. The 95% confidence interval is equal to plus/minus two (2) times the standard deviation.

Table 8 shows the possible values for CR determined from the variability analysis and CR values measured experimentally. The 95% confidence intervals of the equivalent and measured CR values have overlapping quantities confirming the variability analysis and experimental results, Figure 26. For example, the variability analysis shows that 95% of CR measurements for pocket C9 will fall between 41 micrometers and 139 micrometers of CR. The average measured CR for pocket C9 is 108 micrometers which is within the 95% confidence interval of the variability analysis.

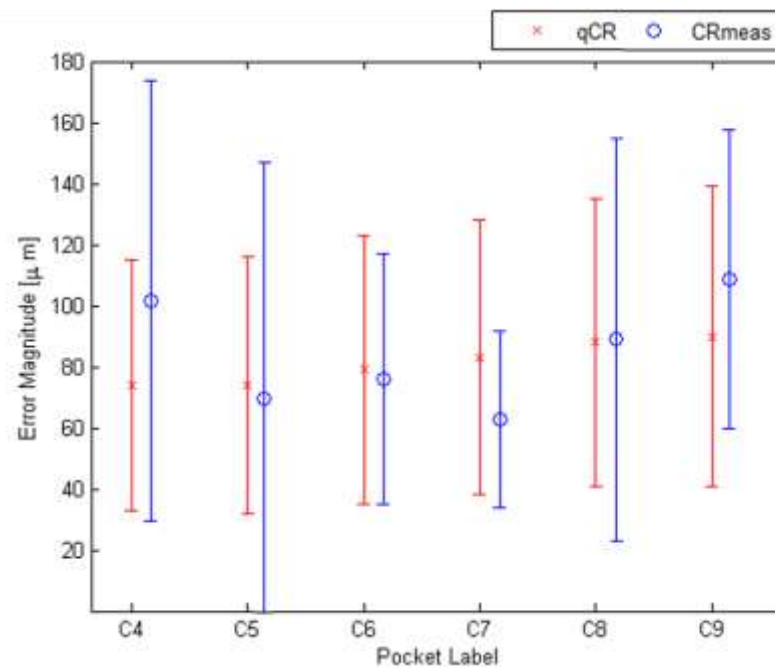


Figure 26: Plot showing the mean error and 95% confidence intervals of the variability analysis and measured CR values.

CHAPTER 5: EXTENSION TO 3D MACHINING

The previous chapters discuss using sub-scale machining for creating essentially two-dimensional parts. Chapter 2 discusses how the 3 axis Haas VMM cannot compensate for all six (6) DOF of misalignment between the part and machine. To account for part and machine misalignment in all six (6) degrees of freedom, a machine with five (5) controlled axes is required. Previous 2D experiments have dealt with distances between holes, a 3D extension of Sub-scale machining would allow machines to work on multiple sides of a part and aim to improve the accuracy of features in terms of concentricity, angularity, parallelism and perpendicularity. Although no parts were machined and measured, an analysis has been performed to determine what sort of errors are expected when using a laser tracker as the stand alone metrology system in a 3D sub-scale machining operation.

To evaluate the ability of the laser tracker to measure differences in orientation, a plate with six (6) fiducial locations was used to define a plane. The plane was rotated about one (1) axis of the plate with the angle of tilt measured using an angle measuring interferometer and laser tracker. The plate was measured at three (3) different angles and measured five (5) times with the laser tracker at each angle. Initially the plate was lying flat on the table, the angle interferometer was set to zero (0) and the plate was measured five (5) times with the laser tracker. The plate was fixtured about the pivot point and at the end propped up using toe clamps. The toe clamps were lightly tightened to hold the

plate in place when removing the SMR from the magnetic SMR nests, Figure 27. With this setup, the angle interferometer is not significantly influenced by rotations and translations in other DOF that may occur when changing the plate's angle, reducing undesired rotations and translations that the interferometer would not measure but would arise in the laser tracker measurements.

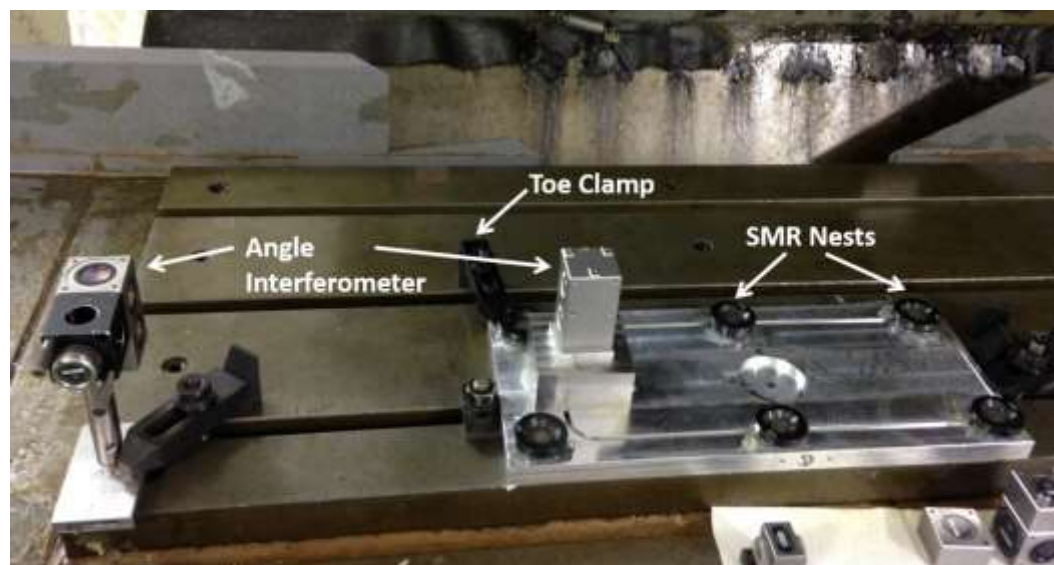


Figure 27: Angle measurement setup. The dimensions of the plate are approximately 14 by 7 inches.

The plate was tilted to approximately 0.4, 0.8 and 1.2 degrees. The angle of the plate determined by the angle interferometer was recorded before and after the measurements with the laser tracker were made. The before and after measurements of the angle interferometer were averaged to determine best estimate angle interferometer measurements. The before and after angle interferometer measurements varied by a max of four (4) arc seconds, Table 10.

Table 10: Before and after and average (best estimate) angle interferometer measurements.

	Interferometer Meas. Start	Interferometer Meas. End	Interferometer Av.	Interferometer Av.
	[arcsec]	[arcsec]	[arcsec]	[deg]
Pos1	0	0	0	0.00
Pos2	1445	1443	1444	0.40
Pos3	2833	2829	2831	0.79
Pos4	4176	4175	4176	1.16

The angle of the plate as measured by the laser tracker was determined by the angle between the normal vector of the best fit plane formed by the SMR targets with the plate at an angled positions and the average of the plate normal vectors of the 5 plate measurements at the initial position, Table 11.

Table 11: Angle of plate as determined by the laser tracker.

	LT Pos 2	LT Pos 3	LT Pos 4
	[mrad]	[mrad]	[mrad]
Trial 1	7.24	13.95	20.49
Trial 2	7.23	13.97	20.52
Trial 3	7.23	13.95	20.51
Trial 4	7.22	13.96	20.49
Trial 5	7.23	13.94	20.50

The difference between the laser tracker and angle interferometer angle was calculated, Table 12. Table 12, shows the difference between the interferometer and laser tracker measurements to be a few tenths of a milliradian. Similar results are seen when examining lines machined in two separate setups in the initial part series. Figure 28 is a measurement of the top line on one of the parts in PS3 where the top line was machine in

separate setups. The line was measured on a tactile CMM and linear LSQ line fit to the left side and the right side of the straight line. The angle between the straight lines is determined to be the angle between the two setups. The other 4 parts in PS3 show the angle between the setups to be 0.56, 0.09, 0.37 and 0.41 milliradians.

Table 12: Difference between angle interferometer and laser tracker measurements.

	LT Pos 2 - Interferometer Pos 2	LT Pos 3 - Interferometer Pos 3	LT Pos 4 - Interferometer Pos 4
	[mrad]	[mrad]	[mrad]
Trial 1	0.24	0.22	0.24
Trial 2	0.23	0.25	0.28
Trial 3	0.23	0.23	0.26
Trial 4	0.22	0.24	0.25
Trial 5	0.23	0.22	0.26

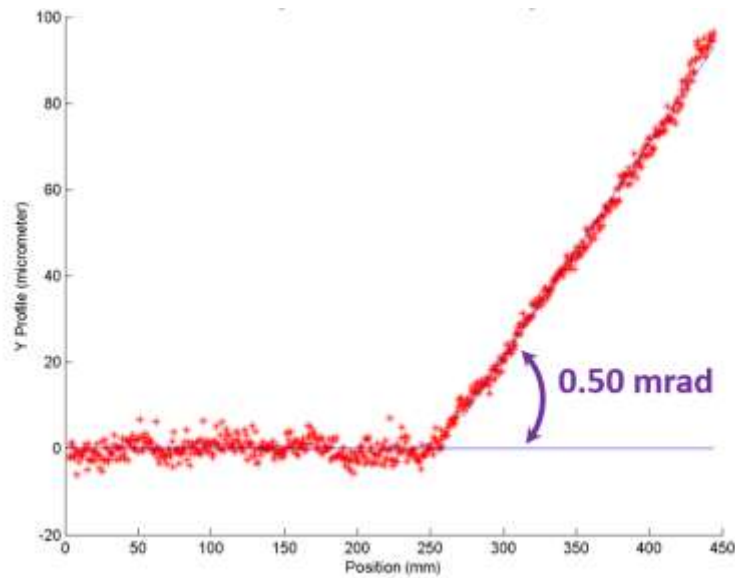


Figure 28: Straight line machined in two separate setups with best fit linear line subtracted from all points and angle between first and second section showing disorientation of CS between both setups.

With these particular setups, the errors in orientation realization are on the order of a few tenths of a milliradian. Orientation error will contribute to position error as well, where position error will increase as the distance from what is defined as the origin increases. For a multi-sided machining operation, angularity, parallelism and perpendicularity tolerance on the order of a few tenths of a milliradian is expected to be met.

CHAPTER 6: CONCLUSION AND FUTURE WORK

The experiments and analysis of sub-scale machining have shown that sub-scale machining is possible and produces 2D parts with feature location errors on the order of the uncertainty of the metrology device used to align and register the part and machine CSs. This study has shown that it is important to measure all machine and part features in approximately the same plane as milling will take place to reduce influences from machine geometry errors. It is also important to ensure that the part maintains the same shape throughout the machining process because changes in the parts shape lead to changes in the realization of the part CS causing large part feature location errors. Sub-scale machining can also be used for the machining of 3D components using the same measurement methods. Through a series of experiments, it is determined that part features can be held to orientation tolerances of a few tenths of a milliradian.

To continue the exploration of Sub-scale Machining, experimental results from a multisided 3D machining operation would validate the theoretical evaluation. When part dimensions are on the order of several meters or more, compensation for part temperature can be incorporated in the post-processing algorithm using methods described in the FCS to achieve more accurate feature locations assuming the time between part temperature measurements, post-processing and machining is relatively short. The limitations of sub-scale machining are the measurement capability of the large scale metrology system, machine tool accuracy and thermal environment.

To make a process like sub-scale machining possible, there are many challenges that need to be addressed. Foreseen challenges include a.) logistics of managing locating and fixturing multiple machine tools on the shop floor, b.) providing the machine tool with necessary resources such as pressurized air, coolant and electricity in a flexible and rapid manner, c.) managing of the chips and coolant as to not interfere with working machine tool and the placement of future machine tools, d.) tool changes and tool management, e.) operation of relatively sensitive metrology equipment in a volatile manufacturing environment, f.) planning of fiducial locations to ensure no fiducials are in tool paths, damaged or removed and remain within the visibility of the metrology instrument(s), g.) selection of fiducial type because magnet fiducials collect ferrous chips requiring cleaning after each machining operation, h.) the decomposition of the large component into sections to allow tool paths to be programmed and i.) development of the fixturing process to minimize effects from released residual stresses.

For the experiments in this project, the large components were broken down into sections that were known to fit within the Haas VMCs working area and each section placed in a pseudo random location within the working area when machined. There was still a small possibility that the section that was planned to be machined has portions outside of the Haas VMC's working area causing a machine axis over-travel error. This would cause the operator to have to reposition the part on the machine's table, wasting the time spent in the initial measurements and setup. To make the sub-scale machining process more efficient, an ideal situation would be to randomly place the part within the machine tool's working area and have software recognize what portion of the part the machine tool can reach. Then, write the nominal NC code and alter the code for axis

misalignment to avoid the possibility of misplacing the machine on the work piece. The software could keep track of what portions of the part were not complete and direct the placement of machine tools to achieve minimal part cycle times.

REFERENCES

- [1] J. E. Halley, K. S. Smith, A. M. Helvey and W. R. Winfough, "The Impact of High Speed Machining Technology on the Design and Fabrication of Aerospace Components," in *Proceedings of the: 1999 ASME Design Engineering Technical Conferences*, Las Vegas, Nevada, 1999.
- [2] M. Albert, "Going to Great Lengths," *Modern Machine Shop*, 11 March 2010. [Online]. Available: <http://www.mmsonline.com/articles/going-to-great-lengths>. [Accessed 26 January 2015].
- [3] E. G. Loewen and Bausch & Lomb Inc., "Metrology Problem in General Engineering: A Comparison with Precision Engineering".
- [4] S. Smith, "The Need Research and Development in Manufacturing and Metrology".
- [5] W. Cuypers, N. Van Gestel, A. Voet, J.-P. Kruth, J. Mingneau and P. Bleys, "Optical measurement techniques for mobile and large-scale metrology," in *Optics and Lasers in Engineering*, 2009.
- [6] X. Fernandez and J. Amat, "Research on Small Fiducial Mark Use for Robotic Manipulation and Alignment of Ophthalmic Lenses," in *7th IEEE International Conference on Emerging Technologies and Factory Automation*, 1999.
- [7] [Online]. Available: http://www.sepptronic.de/NEUES/IPC_SMEMA_3.1.pdf.
- [8] H. Karabork, F. Yildiz and E. Coskun, "Object Recognition for Interior Orientation in Digital Photogrammetry," in *XXth ISPRS Congress, Geo-Imagery Bridging Continents*, 2004.
- [9] A. B. Woody, *Fiducial Calibration System: A Technique for improving the accuracy of Large Scale Monolithic Components*, 2005.
- [10] *Leica SMR Calibration Certificate Serial No: 13204*, 2015.
- [11] A. H. Slocum, "Kinematic coupling for precision fixturing - Part 1: Formulation of design parameters".

- [12] B. Bridges, "How Laser Trackers Work".
- [13] H. Metrology, "PCMM System Specifications Leica Absolute Tracker AT901 and Leica T-Products".
- [14] "ASME B89.4.19 Performance Evaluation Tests and Geometric Misalignments in Laser Trackers," *Journal of Research of the National Institute of Standards and Technology*, vol. 114, no. 1, pp. 21-35, 2009.
- [15] "FARO Laser Tracker Vantage Integrates Manufacturing and Quality Assurance," [Online]. Available: <http://www.faro.com/en-in/products/metrology/faro-laser-tracker/case-studies/2014/04/11/faro-laser-tracker-vantage-integrates-manufacturing-and-quality-assurance>. [Accessed 13 January 2013].
- [16] J. H. Burge, P. Su, C. Zhao and T. Zobrist, "Use of a commercial laser tracker for optical alignment," in *Proc. of SPIE Vol. 6676*, 2007.
- [17] T. L. Zobrist, H. T. Burge, W. B. Davison and H. M. Martin, "Measurements of large optical surfaces with a laser tracker," in *Proc. of SPIE*, 2008.
- [18] "Performance Evaluation of Laser-Based Spherical Measurement Systems," ASME B89.4.19, 2006.
- [19] T. Welty, *Evaluation of Laser Trackers*, 2014.
- [20] J. Raja and B. Muralikrishnan, *Computational Surface and Roundness Metrology*, 2009.
- [21] T. Yoshizawa, "Laser Tracker Systems," in *Handbook of Optical Dimensional Metrology*, CRC Press, 2009, pp. 138 -145.
- [22] NIST, "JCGM 100:2008 Evaluation of measurement data - Guide to the expression of uncertainty in measurement," 2008.
- [23] NIST, "JCGM 101:2008 Evaluation of measurement data - Supplement 1 to the "Guide to the expression of uncertainty in measurement" - Propagation of distributions using a Monte Carlo Method," 2008.
- [24] E. Morse and H. Voelcker, "A Tale of Two Tales," in *The Brown & Sharpe Publication of Precision Manufacturing*.

- [25] L. Kirkup and B. Frenkel, An Introduction to Uncertainty in Measurement, Cambridge University Press, 2006.
- [26] S. R. Patterson, "Treatments of Errors and Uncertainty an ASPE Tutorial".
- [27] T. C. Inc, Geometric Dimensioning and Tolerancing Pocket Guide, Long Boat Key, 1995.
- [28] J. J. Craig, Introduction to Robotics Mechanics & Control, Adison-Wesley Publishing Company, 1986.
- [29] S. A. Kyle, S. Robson, D. P. Chapman, P. A. Cross, J. C. Iliffe and S. Oldfield, "Understanding Large Scale Metrology".
- [30] J. R. Taylor, An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements, Sausalito, CA: University Science Books, 1982.
- [31] S. Smith, B. Woody and D. Chopra, "Including the Uncertainty of CMM Measurements and Stitching Algorithms in the Uncertainty Analysis of Fiducial Based Machining".

APPENDIX A: MATLAB POST-PROCESSING CODE

Transforming_g_code.m (Main Script)

```
%The purpose of this script is to alter the NC code with point
coordinates
%in the part coordinate sytem and tranform them into coordinates in the
%machine coordinate system.

clc;
clearvars -except partcode

%Calculate transofrmation matrices
pTt = tracker_to_part;
mTt = tracker_to_machine;
[mTp, mRp] = part_to_machine(pTt, mTt);

%Determining the size of the NC code array.
[m,n]=size(partcode);

%Opens a files for the new part code to be written to. Erases all
previous
%text if the file already exists.
fid = fopen('newpartcode.nc','wt');

%Creating an array with an extra empty row.
pc=[partcode cell(m,1)];

%Initializes the array of same size for the new part code.
pcn=cell(m,n+1);

prg_typ=1; %Initialize program type
for i=1:m %For loop through rows

    p_done=(i/length(partcode)*100); %Print program type on main screen
    fprintf('Row %d of %d or %3.0f%% done \n',i,length(partcode),
p_done);

    j=1; %Initialize j
    %This for loop looks through each row for a G90, G91 or G28 command
    %which signifies whether the programming coordinates are
    incremental,
    %absolute or relative it the machine home.
    for k=1:n
        aa=strfind(pc{i,k},'G90'); %Searching for G90, absolute
        coordinate programming
        bb=strfind(pc{i,k},'G91'); %Searching for G91, incremental
        coordinate programming
        cc=strfind(pc{i,k},'G28'); %Searching for for G28 commands
```



```

    if aa==1
        prg_typ='abs';
    elseif bb==1
        prg_typ='inc';
    elseif cc==1
        prg_typ='mach';
    else
        if prg_typ==1 %Rename program type
            prg_typ=prg_typ;
        else
            prg_typ=prg_typ;
        end
    end
end

end

%Search for X,Y,I, & J coordinates and transform based on
coordinate
%mode.
while j<=n %For loop through columns

    if strcmp(prg_typ,'abs')==1 %If the current coordinates are in
absolute mode

        str=pc{i,j}; %Naming the string of interest.
        str2=pc{i,j+1}; %Name the forward string of interest.

        %strfind returns the starting character indice of the
desired string
        a=strfind(str,'X'); %Search for a X character in the
string.
        b=strfind(str,'Y'); %Search for a Y character in the
string.
        c=strfind(str2,'Y'); %Search for a Y character in the next
column. Used when an X and Y charater is used in a row.
        d=strfind(str, 'I'); %Search for a I character for G2 and
G3 commands.
        e=strfind(str2, 'J'); %Search for a J character for G2 and
G3 commands.
        f=strfind(str2, 'F'); %Search for an F character in the
current string.

        if a==1 & c==1 %If an X and Y character are detected.
Isolate respective values, apply transformation, write new values.

            x_char=numel(str); %Count the number of characters in
the X string.
            y_char=numel(str2); %Count the characters in the Y
string.
            x=str2num(str(2:x_char)); %Converting the X string
number to a value.
            y=str2num(str2(2:y_char)); %Converting the Y string
number to a value.

```

```

        [xnew,ynew]=transform(x,y,mTp); %Applies transformation
matrix.

        pcn{i,j}=strcat('X',num2str(xnew,'%10.4f')); %Creates X
pre-fixed string with new X values rounded to 4 decimal places.
        pcn{i,j+1}=strcat('Y',num2str(ynew,'%10.4f')); %Creates
Y pre-fixed string with new Y values rounded to 4 decimal places.

        j=j+2;

        elseif a==1 & isempty(b)==1 %If an X coordinate is found
but no Y coordinate, the previous Y coordinate needs to be used. The
previous Y values should still be stored as variable "y"
        y=y; %The y value is still equal to the previous y
value.
        x_char=numel(str); %Count the number of characters in
the X string.
        x=str2num(str(2:x_char)); %Converting the X string
number to a value.

        [xnew,ynew]=transform(x,y,mTp); %Applies transformation
matrix.

        if isempty(str2)==0
                pcn{i,j}=strcat('X',num2str(xnew,'%10.4f'));
%Creates X pre-fixed string with new X values rounded to 4 decimal
places.
                pcn{i,j+1}=strcat('Y',num2str(ynew,'%10.4f'));
%Creates Y pre-fixed string with new Y values rounded to 4 decimal
places.
                pcn{i,j+2}=str2;
                j=j+3;
        else
                pcn{i,j}=strcat('X',num2str(xnew,'%10.4f'));
%Creates X pre-fixed string with new X values rounded to 4 decimal
places.
                pcn{i,j+1}=strcat('Y',num2str(ynew,'%10.4f'));
%Creates Y pre-fixed string with new Y values rounded to 4 decimal
places.
                j=j+2;
        end

        elseif b==1 & isempty(a)==1 %If a Y coordinate is found but
no X coordinate, the previous X coordinate needs to be used. The
previous X value should still be the variable "x"
        x=x; %The x value is still equal to the previous x
value.
        y_char=numel(str); %Count the characters in the Y
string.
        y=str2num(str(2:y_char)); %Converting the Y string
number to a value.

        [xnew,ynew]=transform(x,y,mTp); %Applies transformation
matrix.

```

```

        if isempty(str2)==0
            pcn{i,j}=strcat('X',num2str(xnew,'%10.4f'));
%Creates X pre-fixed string with new X values rounded to 4 decimal
places.
            pcn{i,j+1}=strcat('Y',num2str(ynew,'%10.4f'));
%Creates Y pre-fixed string with new Y values rounded to 4 decimal
places.
            pcn{i,j+2}=str2;
            j=j+3;
        else
            pcn{i,j}=strcat('X',num2str(xnew,'%10.4f'));
%Creates X pre-fixed string with new X values rounded to 4 decimal
places.
            pcn{i,j+1}=strcat('Y',num2str(ynew,'%10.4f'));
%Creates Y pre-fixed string with new Y values rounded to 4 decimal
places.
            j=j+2;
        end

        elseif d==1 & e==1 %If an I an J command is found for a G2
or G3 command
            i_char=numel(str); %Count the number of characters in
the X string.
            j_char=numel(str2); %Count the characters in the Y
string.
            ic=str2num(str(2:i_char)); %Converting the X string
number to a value.
            jc=str2num(str2(2:j_char)); %Converting the Y string
number to a value.
            [Inew,Jnew]=ij_trans(ic,jc, mRp); %Call the function to
transform I and J
            pcn{i,j}=strcat('I',num2str(Inew,'%10.4f')); %Creates I
pre-fixed string with new I values rounded to 4 decimal places.
            pcn{i,j+1}=strcat('J',num2str(Jnew,'%10.4f')); %Creates
J pre-fixed string with new J values rounded to 4 decimal places.
            j=j+2;
        else
            pcn{i,j}=partcode{i,j};
            j=j+1;
        end

        elseif strcmp(prg_typ,'inc')==1 %If the current coordinates are
in incremental mode
            str=pc{i,j}; %Naming the string of interest.
            str2=pc{i,j+1}; %Name the forward string of interest.

            %strfind returns the starting character indice of the
desired string
            a=strfind(str,'X'); %Search for a X character in the
string.
            b=strfind(str,'Y'); %Search for a Y character in the
string.
            c=strfind(str2,'Y'); %Search for a Y character in the next
column. Used when an X and Y charater is used in a row.
            d=strfind(str, 'I'); %Search for a I character for G2 and
G3 commands.

```

```

        e=strfind(str2, 'J'); %Search for a J character for G2 and
G3 commands.
        f=strfind(str, 'G54'); %Search for a G54 command. If a G90
command is found before G54, which it typically is, the X and Y
coordinates need to be transformed instead of only rotated.

        if a==1 & c==1 %If an X and Y character are detected.
Isolate respective values, apply transformation, write new values.

                x_char=numel(str); %Count the number of characters in
the X string.
                y_char=numel(str2); %Count the characters in the Y
string.
                x=str2num(str(2:x_char)); %Converting the X string
number to a value.
                y=str2num(str2(2:y_char)); %Converting the Y string
number to a value.

                [xnew,ynew]=ij_trans(x,y,mRp); %Applies transformation
matrix.

                pcn{i,j}=strcat('X',num2str(xnew,'%10.4f')); %Creates X
pre-fixed string with new X values rounded to 4 decimal places.
                pcn{i,j+1}=strcat('Y',num2str(ynew,'%10.4f')); %Creates
Y pre-fixed string with new Y values rounded to 4 decimal places.

                j=j+2;

        elseif a==1 & isempty(b)==1 %If an X coordinate is found
but no Y coordinate, the previous Y coordinate needs to be used. The
previous Y values should still be stored a variable "y"
                y=0; %The y value is still equal to the previous y
value.
                x_char=numel(str); %Count the number of characters in
the X string.
                x=str2num(str(2:x_char)); %Converting the X string
number to a value.

                [xnew,ynew]=ij_trans(x,y,mRp); %Applies transformation
matrix.

                pcn{i,j}=strcat('X',num2str(xnew,'%10.4f')); %Creates X
pre-fixed string with new X values rounded to 4 decimal places.
                pcn{i,j+1}=strcat('Y',num2str(ynew,'%10.4f')); %Creates
Y pre-fixed string with new Y values rounded to 4 decimal places.

                j=j+2;

        elseif b==1 & isempty(a)==1 %If a Y coordinate is found but
no X coordinate, the previous X coordinate needs to be used. The
previous X value should still be the variable "x"
                x=0; %The x value is still equal to the previous x
value.
                y_char=numel(str); %Count the characters in the Y
string.

```

```

        y=str2num(str(2:y_char)); %Converting the Y string
number to a value.

        [xnew,ynew]=ij_trans(x,y,mRp); %Applies transformation
matrix.

        pcn{i,j}=strcat('X',num2str(xnew,'%10.4f')); %Creates X
pre-fixed string with new X values rounded to 4 decimal places.
        pcn{i,j+1}=strcat('Y',num2str(ynew,'%10.4f')); %Creates
Y pre-fixed string with new Y values rounded to 4 decimal places.

        j=j+2;

    elseif d==1 & e==1 %If an I an J command is found for a G2
or G3 command
        i_char=numel(str); %Count the number of characters in
the X string.
        j_char=numel(str2); %Count the characters in the Y
string.
        ic=str2num(str(2:i_char)); %Converting the X string
number to a value.
        jc=str2num(str2(2:j_char)); %Converting the Y string
number to a value.

        [Inew,Jnew]=ij_trans(ic,jc, mRp); %Call the function to
transform I and J

        pcn{i,j}=strcat('I',num2str(Inew,'%10.4f')); %Creates I
pre-fixed string with new I values rounded to 4 decimal places.
        pcn{i,j+1}=strcat('J',num2str(Jnew,'%10.4f')); %Creates
J pre-fixed string with new J values rounded to 4 decimal places.
        j=j+2;

    else
        pcn{i,j}=partcode{i,j};
        j=j+1;
    end

    elseif strcmp(prg_typ,'mach')==1 %If the current coordinates
are in machine coordinates
        pcn{i,j}=partcode{i,j};
        j=j+1;
    else
        pcn{i,j}=partcode{i,j};
        j=j+1;
    end

end

end %Column for loop end

%Printing to new .NC file
for ii=1:m

```

```

    for jj=1:n
        if isempty(pcn{ii,jj})==1
            break
        else
            fprintf(fid, '%s ',pcn{ii,jj});
        end
    end
    fprintf(fid, ';\n'); %Add semicolon on end of each row
end

fclose(fid); %Closing new part code file

%clearvars -except partcode

```

transform.m

```

function [xnew,ynew] = transform(x,y,mTp)
%The purpose of this function is to transform the coordinates in the
%part coordinate system into the machine coordinate system.

cP=[x; y; 0; 1]; %Point in part CS with Z location removed due to 2d
nature

dP=mTp*cP; %Transform point into machine CS.

xnew=dP(1,1);

ynew=dP(2,1);

end

```

tracker_to_part.m

```

function [pTt] = tracker_to_part
%The purpose of this function is to compute the transformation matrix
from
%the tracker to part coordinate system. Taken from Computational
Surface
%and Roundness Metrology by Jay Raja.

%% Determine part plane normal vector from plane fit of SMR coordinates
load('points.txt');
M=points;
n=size(M,1);
% find centroid
xc = sum(M(:,1))/n;
yc = sum(M(:,2))/n;
zc = sum(M(:,3))/n;
% build matrix A
for i=1:n

```

```

A(i,1) = M(i,1)-xc;
A(i,2) = M(i,2)-yc;
A(i,3) = M(i,3)-zc;
end
% find eigen values and eigen vector corresponding
% to largest eigen value
[v,d] = eig(A'*A);
a1=[d(1,1);d(2,2);d(3,3)];
[y,i]= min(a1);
a = v(1,i); %Direction cosine of Z onto X
b = v(2,i); %Direction cosine of Z onto Y
c = v(3,i); %Direction cosine of Z onto Z

z_unit=[a b c];

%% Determine part rotation
SMR1=points(1,:); %Coordinates of the SMR in tracker frame.
SMR2=points(2,:);

x_vec=SMR2-SMR1; %Calculate X axis vector

%Determine the unit vector each axis.
xa=x_vec(1,1)/norm(x_vec);
xb=x_vec(1,2)/norm(x_vec);
xc=x_vec(1,3)/norm(x_vec);
x_unit=[xa xb xc];

y_unit=cross(z_unit,x_unit); %Calculate Y vector

%% Build transformation matrix
pTt=[x_unit(1,1) y_unit(1,1) z_unit(1,1) SMR1(1,1); x_unit(1,2)
y_unit(1,2)
z_unit(1,2) SMR1(1,2); x_unit(1,3) y_unit(1,3) z_unit(1,3)
SMR1(1,3);
0 0 0 1];

end

```

tracker_to_machine.m

```

function [mTt] = tracker_to_machine
%The purpose of this function is to build the transformation matrix
from
%the tracker to the machine coordinates.

%Load the text files exported data from Spacial Analyzer
load('Xrun.txt'); %Load txt file with point data obtained by moving the
machine table in the pos X direction.
load('Yrun.txt'); %Load txt file with point data obtained by moving the
machine table in the pos Y direction.
load('Origin.txt'); %Load text file with location of the machine
spindle relative to the tracker.

```

```

%Determine the origin or the spindle using a circle scan and
substituting a
%LSQ Circle. (LSCircle function from Mario Valdez)
c_x=Origin(:,1); %Circle scan X coordinates
c_y=Origin(:,2); %Circle scan Y coordinates
x_guess=(max(c_x)+min(c_x))/2;
y_guess=(max(c_y)+min(c_y))/2;

%Estimate spindle location based on circle scan
[xc,yc,R] = LSCircle(c_x,c_y,[x_guess y_guess]);
zc=mean(Origin(:,3));

%Assign imported values arrays
x_x=flipud(Xrun(:,1)); %X coordinates of machine X run
x_y=flipud(Xrun(:,2)); %Y coordinates of machine X run
x_z=flipud(Xrun(:,3)); %Z coordinates of machine X run
y_x=flipud(Yrun(:,1)); %X coordinates of machine Y run
y_y=flipud(Yrun(:,2)); %Y coordinates of machine Y run
y_z=flipud(Yrun(:,3)); %Z coordinates of machine Y run

%Step1: Find x_bar, y_bar, and z_bar for the x & y movements
x_xsum=0;
for i=1:length(x_x)
    x_xsum=x_xsum+x_x(i);
end
x_xbar=x_xsum/length(x_x);

x_ysum=0;
for i=1:length(x_y)
    x_ysum=x_ysum+x_y(i);
end
x_ybar=x_ysum/length(x_y);

x_zsum=0;
for i=1:length(x_z)
    x_zsum=x_zsum+x_z(i);
end
x_zbar=x_zsum/length(x_z);

y_xsum=0;
for i=1:length(y_x)
    y_xsum=y_xsum+y_x(i);
end
y_xbar=y_xsum/length(y_x);

y_ysum=0;
for i=1:length(y_y)
    y_ysum=y_ysum+y_y(i);
end
y_ybar=y_ysum/length(y_y);

y_zsum=0;
for i=1:length(y_z)
    y_zsum=y_zsum+y_z(i);
end

```



```

y_zbar=y_zsum/length(y_z);

%Formulating the matrix A
x_A=zeros(length(x_x),3);
y_A=zeros(length(y_x),3);

for i=1:length(x_x)
    x_A(i,:)=[x_x(i)-x_xbar, x_y(i)-x_ybar, x_z(i)-x_zbar];
end

for i=1:length(y_x)
    y_A(i,:)=[y_x(i)-y_xbar, y_y(i)-y_ybar, y_z(i)-y_zbar];
end

%Using the Singular Value Decomposition to find the direction cosines
[Ux Sx Vx]=svd(x_A);
[Uy Sy Vy]=svd(y_A);

x_vec=[Vx(1,1),Vx(2,1),Vx(3,1)]; %Build X vector
y_vec=[Vy(1,1),Vy(2,1),Vy(3,1)]; %Build Y vector to define plane
z_vec=cross(x_vec,y_vec); %Calculate Z vector
y_vec2=cross(z_vec,x_vec); %Calculate Y vector

%Creating the Transformation Matrix from tracker to machine.
mTt=[x_vec(1,1) y_vec2(1,1) z_vec(1,1) xc; x_vec(1,2) y_vec2(1,2)
     z_vec(1,2) yc; x_vec(1,3) y_vec2(1,3) z_vec(1,3) zc; 0 0 0 1];

end

```

part_to_machine.m

```

function [mTp, mRp] = part_to_machine (pTt, mTt)
%The purpose of this function is to create the rotation and translative
% matrices dTc from the part to the machine coordinate system.
% These matrices are for the rotation and I and J coordinates used in
G2
% G3 circular interpolation commands.

mTp=mTt^-1*pTt;

mRp=[mTp(1,1) mTp(1,2) mTp(1,3); mTp(2,1) mTp(2,2) mTp(2,3);
     mTp(3,1) mTp(3,2) mTp(3,3)];

end

```

LSCircle.m

```

function [xc,yc,R] = LSCircle(x,y,guess)

```

```

% This function calculates the Least-Squares best-fit for a circle
using
% the general solution.
% Input: x, and y are each n-by-1 vectors and guess is a 1-by-2 vector.
%         Usually you want guess = [0,0] but it depends on the circle.
% Output: xc, yc and R are the Least-Squares estimates.
% Initialize the error as a "large value" and for the Least-Squares
% estimation, setting a tolerance on the iteration serves a purpose
for
% the convergence of a solution.

%Created by: Mario Valdez

pts = [x y];
n = length(pts);
err = 1;
tol = 1e-8;
while err > tol
    Int_pts = pts - ones(n,1)*guess;
    % Polar Conversion.
    [TH,R] = cart2pol(Int_pts(:,1), Int_pts(:,2));
    % Least-Squares algorithm.
    A = [cos(TH).^2, sin(TH).*cos(TH), cos(TH);...
        sin(TH).*cos(TH), sin(TH).^2, sin(TH);...
        cos(TH), sin(TH), ones(n,1)];
    b = [R.*cos(TH);R.*sin(TH);R];
    ATA = A'*A;
    ATb = A'*b;
    X = inv(ATA)*ATb;
    % This shows how far off the initial guess of the circle was from
the
    % estimations.
    err = sqrt(X(1)^2 + X(2)^2);
    % Update our guess by adding the new (x,y) from the solution.
    guess = guess + X(1:2)';
end
% Least-Squares Parameters.
LS_C = [guess' ; X(3)];
xc = LS_C(1,1);
yc = LS_C(2,1);
R = LS_C(3,1);

```

ij_trans.m

```

function [Inew, Jnew] = ij_trans(ic,jc, mRp)
%This function will transform the I and J coordinates from a G2 or G3
%circular interpolation command.
%Most G2 and G3 commands are written with I and J coordinates. I
specifies
%the X coordinate of the center of the radius relative to the start
point
%of the radius. J specifies the Y coordinate of the center of the
radius

```

```
%relative to the start point of the radius. The center of the radius  
does not get  
%translated relative to the start point of the radius. Therefore, point  
only gets  
%rotated.  
  
%Original location of the center of the radius.  
cP=[ic; jc; 0];  
  
%Transformed location of the center of the radius.  
dP=mRp*cP;  
  
%Assigning the new I coordinate to a variable.  
Inew=dP(1,1);  
  
%Assigning the new J coordinate to a variable.  
Jnew=dP(2,1);
```

APPENDIX B: MATLAB POST-PROCESSOR DIRECTIONS FOR USE

The Matlab post-processor for altering NC was built with the intention of making the process of altering the NC code as quick and easy as possible. The post-processor is designed to handle all scenarios of code syntax although some special cases may have been missed. It is important to take precautions to when first using the post-processor with a machine to ensure that no alter code causes a machine spindle crash or controller crash before using the machine to manufacture a part. The post processor is setup so that it would be easy to tweak to a specific machine.

If the laser tracker is on a portable stand, ensure the floor beneath the laser tracker is free of chips or debris or anything that would make the stand settle or shift during use. Place the laser tracker in a location where chips and coolant won't be thrown onto it during machining, where people passing by won't disturb the stand or walk in front of the beam and where the laser tracker can see all points of measurement.

Measuring machine and part coordinate systems with SA

To make the points easier to export later, the points measured for the X axis, Y axis, machine spindle location and part should be measured using different point groups in SA. The point groups should be named "Xrun", "Yrun", "Origin" and "Points", respectively.

The first CS to be measured is the machine CS. To measure the X and Y axes of the machine. In order to ensure the positive direction of the X and Y axes of the machine CS are oriented correctly, the axes are measured while moving in their positive direction. For example, the Haas VMC X axis has a range from negative thirty (-30) inches to zero

(0) inches. Therefore, the machine coordinates measured with the laser tracker start at X = -30 inches and incrementally stepped, pausing every two (2) inches until X = 0 inches is reached. See Figure 29 for example NC code used to measure the Haas VMC X axis.

```
%
011111 (Measure machine axes with LT w/ 6 sec pauses)
G00 G90 G53 X-30. Y-6.; (Bring back to center of working area)
G4 P6.;
G53 X-28.;
G4 P6.;
G53 X-26.;
G4 P6.;
G53 X-24.;
G4 P6.;
G53 X-22.;
G4 P6.;
G53 X-20.;
G4 P6.;
G53 X-18.;
G4 P6.;
G53 X-16.;
G4 P6.;
G53 X-14.;
G4 P6.;
G53 X-12.;
G4 P6.;
G53 X-10.;
G4 P6.;
G53 X-8.;
```

Figure 29: Example NC code for measuring Haas VMC X axis in proper order.

The same applies to the Y axis. See Figure 30 for example NC code used to measure the Haas VMC Y axis. Return the machine table back to the middle of it's working area. Set the part CS offset corresponding to the part CS offset used in the NC code in the machine controller to this location. Now it is important to not move them machine's table until the rest of the measurements are complete.

```

G53 X-15. Y-12.;
G4 P12.;
G53 Y-11.;
G4 P6.;
G53 Y-10.;
G4 P6.;
G53 Y-9.;
G4 P6.;
G53 Y-8.;
G4 P6.;
G53 Y-7.;
G4 P6.;
G53 Y-6.;

```

Figure 30: Example NC code for measuring Haas VMC Y axis in proper order.

To measure the spindle location of the machine, the rotating fixture is held in a tool holder with a ½” collet. The tool holder is inserted into the spindle and the Z axis is lowered to approximately the same position in which most of the milling of the part will occur. Ensure the fixture can rotate 360 degrees freely and that the laser tracker can see the SMR while the fixture is rotated except for the portion that the shaft blocks the laser beam. Set the laser tracker to take a spatial scan with increments of 0.001”. Place the SMR in the SMR nest and begin taking points. Turn the machine spindle on with a spindle speed of 5 RPMs. It is required that the operator hand rotate the SMR in the nest to ensure the SMR doesn’t go outside its reflective range. Stop taking points, remove the SMR and return the spindle back up to tool change position.

Set the laser tracker to take single points. Place the SMR in SMR location 1 on the part. Figure 3 shows the SMR location definitions used for PS4. Measure the location of the SMR with the laser tracker. Place the SMR in SMT location 2 and measure the location of the SMR with the laser tracker. It is important the SMR location 1 and 2 are measured in the proper order because the part CS X axis is defined using these two SMR locations. Continue measure the remaining SMR locations until complete. To make the

points collected by the laser tracker useful, the points are exported from SA into a .txt file.

In SA, right click on the point group containing the points measured to define the machine X axis. Click “Export as ASCII file”, Figure 31. Change the export settings to not include target information in the exported file, Figure 32. Export the points for machine Y axis and name as “Yrun.txt”. Export the points for machine spindle location and name as “Origin.txt”. Export the points for part SMR locations and name as “Points.txt”, Figure 33. A sample .txt file with proper formatting is shown in Figure 34.

In order for post-processor to recognize the .txt files with coordinate data from the laser tracker, all files must be located in the same directory. Cut and paste the four (4) text files into the desired director with the Matlab files.

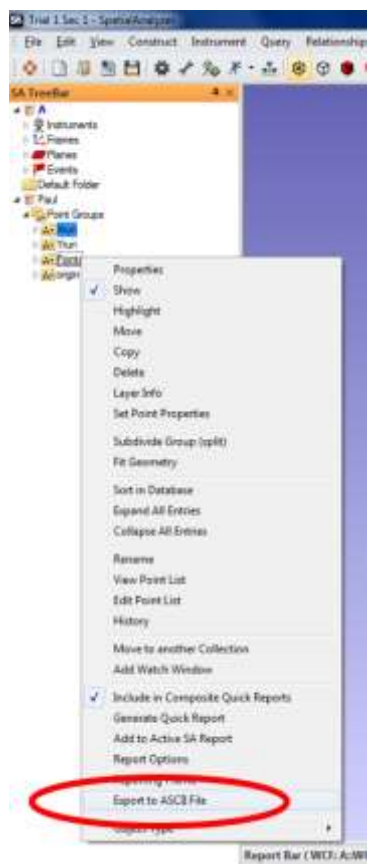


Figure 31: "Export as ASCII File" option in Spatial Analyzer.

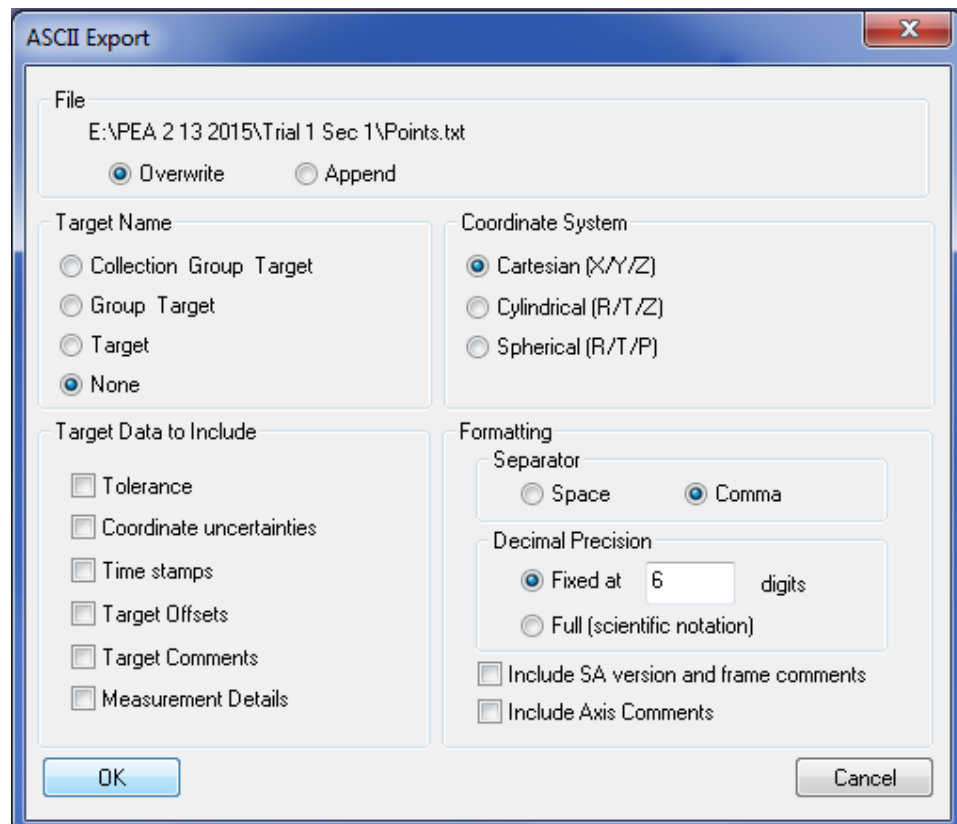


Figure 32: Export file settings in SA.

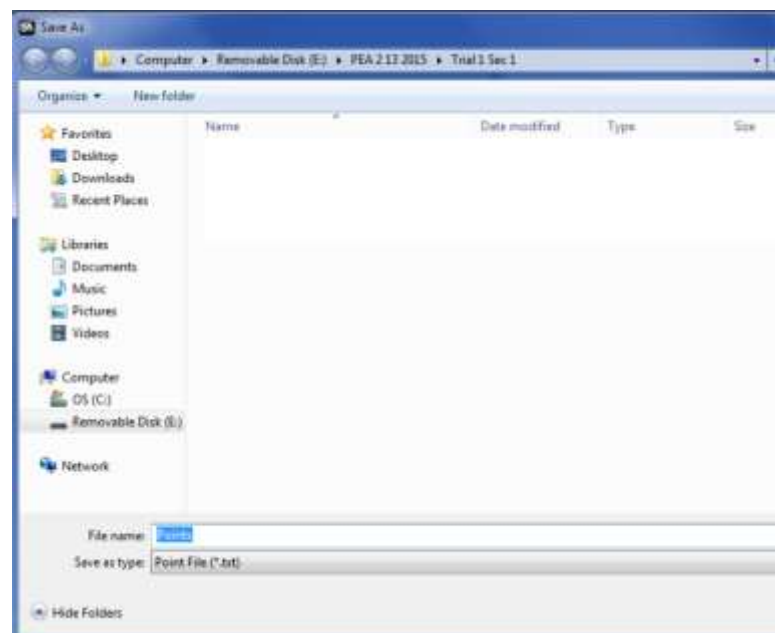
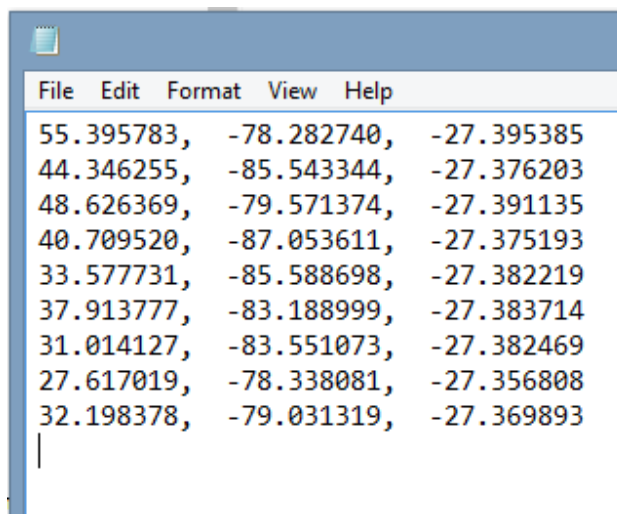


Figure 33: Saving exported text file as "points.txt".



55.395783,	-78.282740,	-27.395385
44.346255,	-85.543344,	-27.376203
48.626369,	-79.571374,	-27.391135
40.709520,	-87.053611,	-27.375193
33.577731,	-85.588698,	-27.382219
37.913777,	-83.188999,	-27.383714
31.014127,	-83.551073,	-27.382469
27.617019,	-78.338081,	-27.356808
32.198378,	-79.031319,	-27.369893

Figure 34: Sample exported text file containing point measurement data with comma delimiter in laser tracker CS.

Transforming Part File

To transform the NC code with the Matlab post-processor, put the desired NC code in the same directory as the Matlab functions and .txt files. Begin importing the original NC file by right-clicking on the original NC file in the Matlab home screen, Figure 35. The Matlab variable import menu will open. Ensure all columns and rows are selected, change the column delimiters to “Space”, change variable type to “Cell Array”, change all column types to “Text”, change variable name to “partcode” (variable name is case sensitive) and click “Import Selection”, Figure 36. Matlab will confirm that the NC code was imported. Exit Matlab variable import menu. The cell array variable created can be seen in the Matlab workspace, Figure 37. Right-click on the main Matlab script, “Transforming_g_code.m”, and click run, Figure 38. In Matlab’s command window, the percent complete will be displayed. Once the script is finished running, a new NC file

name “newpartcode.nc” will be saved in the same directory as the Matlab script, Figure 39. “newpartcode.nc” can be renamed as desired and is ready to be use in the machine tool.

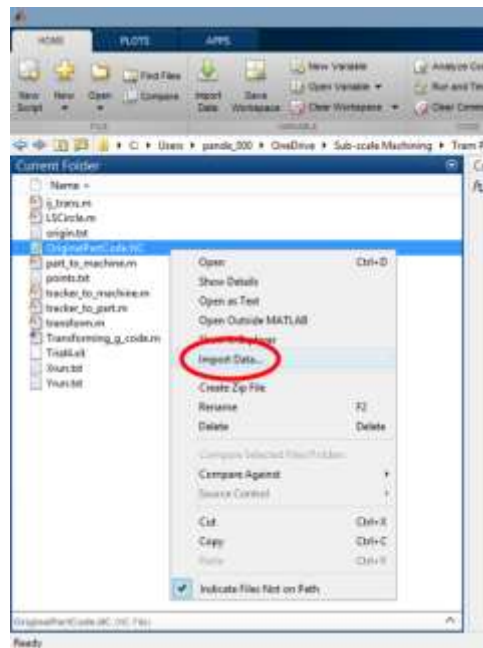


Figure 35: Import data function on Matlab home screen.

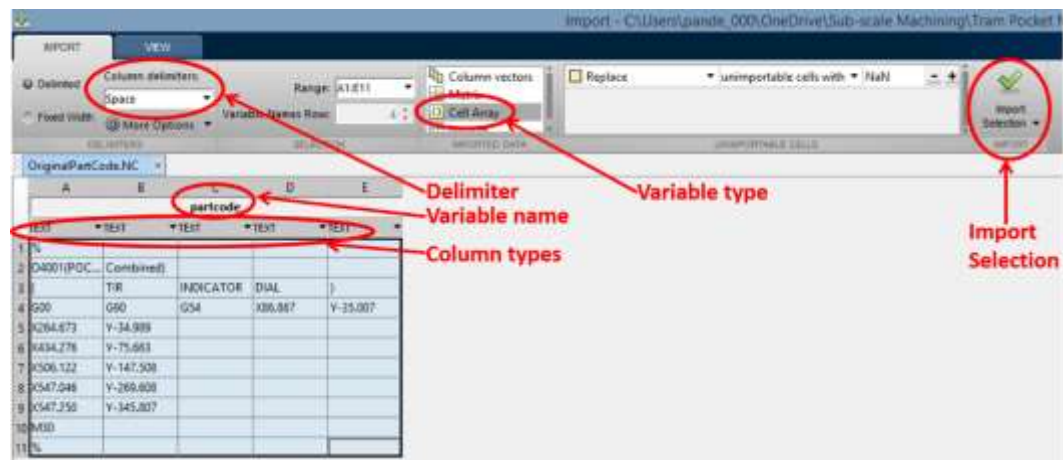


Figure 36: Matlab import data menu with points of interest.

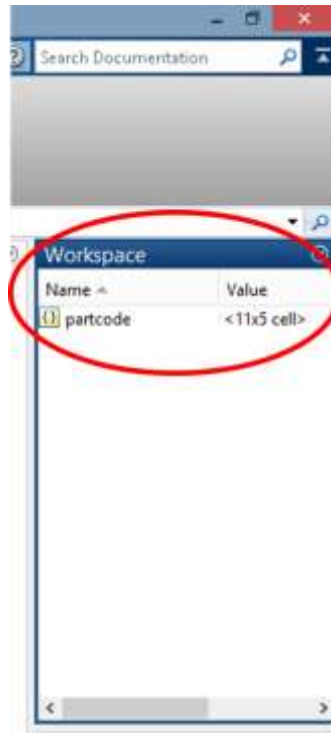


Figure 37: Variable “partcode” cell array created with Matlab import data menu from original NC code.

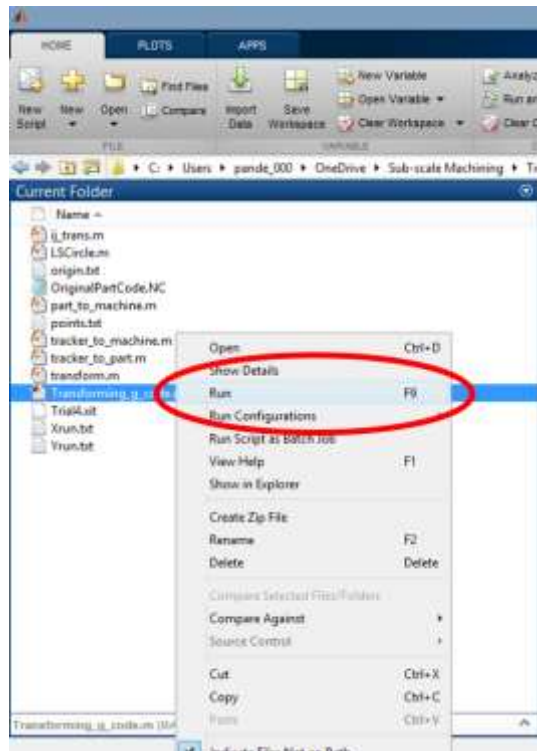


Figure 38: Right-click and select run on the main Matlab script "Transforming_g_code.m".

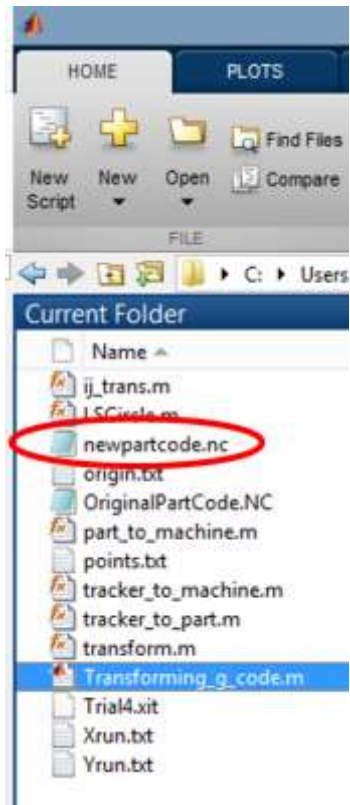


Figure 39: Transformed NC file ready to be renamed if desired and used in a machine tool.