

Faculty of Engineering

**3D Printer Automatic Calibration**

Prepared for

University of Waterloo

Waterloo, Ontario

Malcolm Williams

3A Mechatronics Engineering

May 9th, 2015

High St.

Waterloo, ON

N2L 3X8

May 9th, 2016

Professor William Melek,

Director of Mechatronics Engineering

Department of Mechanical and Mechatronics Engineering

University of Waterloo

Waterloo, Ontario

N2L 3G1

Dear Professor Melek,

This report is entitled “Automatic Calibration Development for a Delta Style 3D Printer”. It is an analysis of the design requirements, implementation and troubleshooting undertaken while designing an auto-calibration tool for delta style 3D printers. It is my second work term report (WKRPT300). It was prepared for the University of Waterloo.

Delta style 3D printers are a new motion platform for 3D printing and are swiftly growing in popularity. This style of printer offers numerous advantages over different styles, but unfortunately can be plagued by accuracy issues due to imperfect build quality. It is my opinion that software can be developed to correct this issue. If successful, it would increase reliability, accuracy and reduce manufacturing cost.

My primary focus was to increase the accuracy of these printers. To achieve this goal, a software model for the kinematics of the printer was derived and an algorithm was developed to determine what misalignments are present and how to correct them.

This report was written entirely by me and has not received any previous academic credit at this or any other institution. This project was undertaken on my own time and not as a part of any organisation. I did not receive any assistance on the project.

Sincerely Yours,

Malcolm Williams

20530280

3A Mechatronics Engineering

Table of Contents

[List of Figures iv](#_Toc454836632)

[List of Tables v](#_Toc454836633)

[Summary vi](#_Toc454836634)

[1. Introduction 1](#_Toc454836635)

[1.1 Background 1](#_Toc454836636)

[1.2 Objective 3](#_Toc454836637)

[2. Methods and Procedures 4](#_Toc454836638)

[2.1 Selection of Critical Parameters 4](#_Toc454836639)

[2.2 Building the Kinematic Model 6](#_Toc454836640)

[2.3 Optimization Algorithm Selection 9](#_Toc454836641)

[3. Analysis of Results 11](#_Toc454836642)

[3.1 Algorithm Tuning: Number of Data Points 12](#_Toc454836643)

[3.2 Algorithm Tuning: Cooling Time 13](#_Toc454836644)

[3.3 Discussion of Results 14](#_Toc454836645)

[4. Conclusions 17](#_Toc454836646)

[5. Recommendations 18](#_Toc454836647)

[6. References 19](#_Toc454836648)

List of Figures

[Figure 1: Calibration Procedure Flow Chart vii](#_Toc450550737)

[Figure 2: Popularity of Delta Printers 1](#_Toc450550738)

[Figure 3: Typical Delta 3D Printer 2](#_Toc450550739)

[Figure 4: Common Printer Terminology 3](#_Toc450550740)

[Figure 5: Critical Tower Location on XY Plane Shown is in pink 5](#_Toc450550741)

[Figure 6: Towers should be Vertical 5](#_Toc450550742)

[Figure 7: Skeletal SolidWorks model of a printer 7](#_Toc450550743)

[Figure 8: Visual Demonstration of the intersection of 3 spheres 9](#_Toc450550744)

[Figure 9: The effect of the number of data points 12](#_Toc450550745)

[Figure 10: Effects of Cooling Time of Global Minimum Finding 14](#_Toc450550746)

[Figure 11: Test of Algorithm Consistency 15](#_Toc450550747)

List of Tables

[Table 1: Algorithm Selection 10](#_Toc450550748)

Summary

A 3D printer must be able to accurately and reliably position itself in 3D space for correct operation. Due to the unique configuration of delta style 3D printers, small mechanical misalignments can translate into large positioning errors. Manually fixing the mechanical misalignments can be a very time consuming and an essentially impossible task for the user. Thus to achieve maximum performance from the printer a software based calibration suite is needed.

To achieve this goal two main steps had to be taken. First, derive a complete kinematic model for the printer. Second, develop an algorithm to correct for the deviations between the software model of the printer and the actual printer.

Current implementations of the printer kinematics contain large assumptions that are unacceptable when attempting to calibrate. Thus, a complete kinematic model had to be derived and implemented. This procedure explained in further detail in the first section of the report.

Next an algorithm was developed to correct for misalignments in the printer build. The algorithm adjusts the printer parameters to make the software model accurately reflect the real world printer. Once an accurate software model is obtained the kinematic equations can accurately position the printer. To improve the algorithm several tests were run on it. The results of these tests are outlined in the second section of the report.

Overall the project was a partial success. The kinematic equations are very helpful in understanding what the critical parameters of the printer are. The optimization algorithm is reliably able to make intelligent guesses about the printer errors and suggest corrections. Unfortunately, gathering reliable data with a printer is a challenge that is not yet solved and severely limits the usefulness of the calibration.

# 

# Introduction

## Background

As 3d printers have continued to increase in popularity different style of printers have begun to emerge on the market [12]. The style that will be discussed in this report is the “delta style”. The delta platform is a parallel robot platform that was originally developed in 1985 for the packing industry, where a robot capable of rapid positioning of lightweight objects was needed [1]. Since then the form factor has been adapted to the 3D printer industry and has been steadily gaining popularity, Figure 1 shows the frequency of the search term “delta printer”[10]. The form factor offers many benefits for 3D printing, ranging from high acceleration [11] to simple, robust construction.

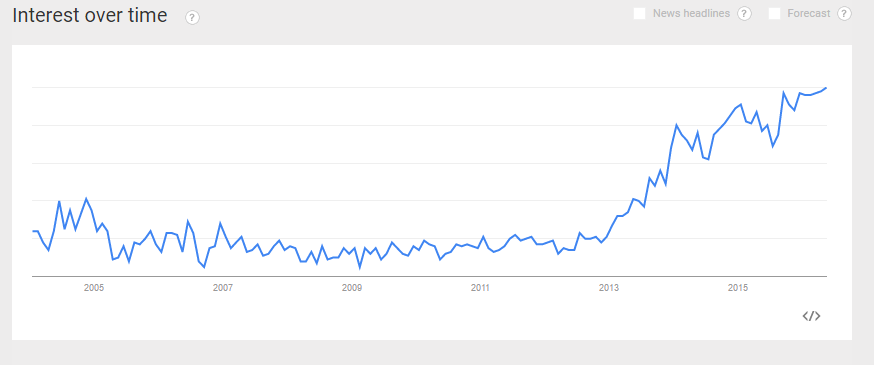


Figure 1: Popularity of Delta Printers [10]

Figure 2 shows a typical consumer delta 3D printer [2]. A carriage is driven up and down each vertical tower of the printer. A pair of rigid arms attached to each carriage and keeps the end effector parallel and a set distance from the carriage. In order to have accurate positioning there must be accurate measurements of the physical parameters of the printer. Due to the size of the machine, accurate measurements can be very difficult, and even the best construction will have some tolerance and variation.

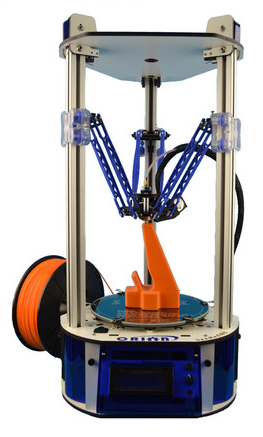
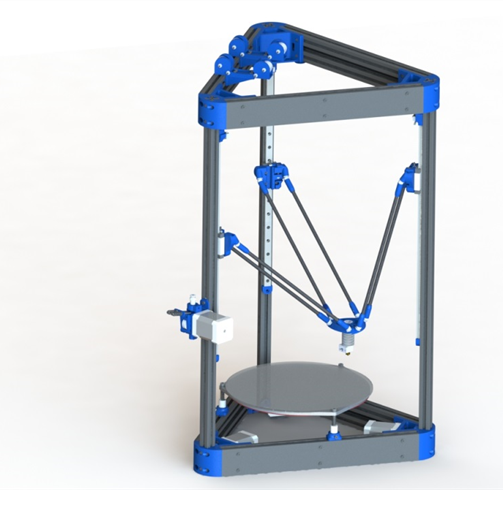


Figure 2: Typical Delta 3D Printer

Printer specific terminology that is critical for understanding this report is presented in Figure 3.



Carriage

Arms

End Effector

Bed

Motor

Tower

Figure 3: Common Printer Terminology

## Objective

The entire problem of accuracy stems from not having an accurate kinematic model of the printer in software. The goal of this project is to construct a piece of software that understands the general kinematics of a delta printer and given a series of training points be able to output the parameters for the printer that best reflects the real printer.

# Methods and Procedures

## Selection of Critical Parameters

The kinematic equations that translate the motor rotation into end effector positioning rely heavily on the mechanical structure of the printer. This section will explore the numerous sources where error can be introduced into the system.

The first source of error is the towers of the printer. To begin, the towers are assumed to be straight. For the printer that this project was done on, the carriages run on 12mm Hiwin recirculating ball carriages, that have a max height variance of 0.015mm and width variance of 0.02 [3], which is small enough that it does not need to be considered. There are three main ways in which the towers can affect accuracy. Firstly, if the printer is viewed from above they should be at the vertices of and equilateral triangle, with the centroid of the triangle being collinear with the center of the bed. Secondly, they should be perfectly vertical, or in other words parallel to each other and perpendicular to the bed. Thirdly they should be facing towards the center of the bed and not rotated along a vertical axis. Figures 4 and 5 show these errors. In an attempt to reduce the number of variables that require tuning the third source of error will be eliminated. This error should not translate into positioning errors, the only potential downside is that it introduces backlash into the end effector.

|  |  |
| --- | --- |
| Capture3 | Capture4 |
| Figure 4: Critical Tower Location on XY Plane  Shown is in pink | Figure 5: Towers should be Vertical |

Another potential source of error is the arms that connect the end effector to the carriages. To achieve maximum accuracy the arms should be measured and then grouped in pairs that are closest matched in length. The arms were measured using precision ball bearings, a granite surface plate and a Mitutoyo vertical height gauge. The respective tolerances are (0.7um [4], 0.01mm [5] and 0.0001 inch [6]). Since the arms can be accurately measured and will not change, they do not need to be calibrated, they just simply need to be accounted for in the kinematic equations. To simplify the process the arm lengths will be entered in the kinematic equation as the average of the matched pairs. If each arm length is accounted for the system will become over constrained. In the real world model there is a certain amount of lash that the mathematical model cannot account for.

The final assumption that can be made is that the bed of the printer is perfectly flat. This is a reasonable assumption because the bed is a 3mm thick borosilicate glass plate. The glass plate was measured off a granite surface plate (deviation less than 0.0001 inch [6]) and was below the detectable level of flatness.

## Building the Kinematic Model

Now armed with the knowledge of which parameters were important I was ready to begin work on the kinematic model. To decrease the development time a SolidWorks skeletal model of the printer was made. This model simplifies the printer down to the base elements and can be used to double check the calculations. The parameters from the program can be exported and subsequently used to drive the SolidWorks model in near real-time. This lets me quickly verify that the kinematic model is in fact working as desired. Figure 6 shows the bare bones kinematic model of the printer. Only the elements of the printer that are essential for the positioning of the end effector are kept.

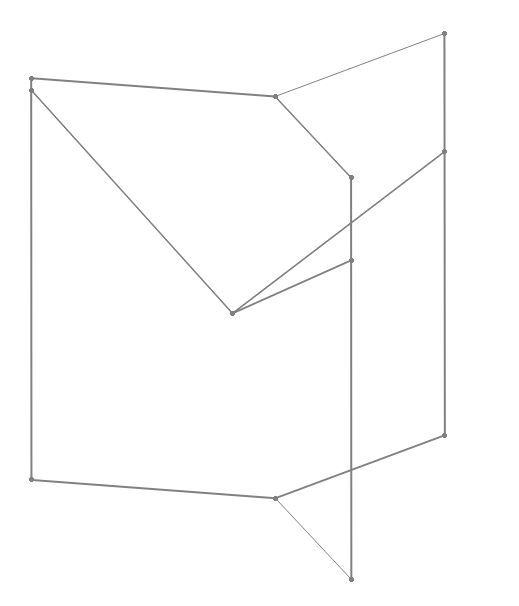


Figure 6: Skeletal SolidWorks model of a printer

The work on the kinematics was divided into two sections. The first section is translating motor rotation into xyz position of each carriage. The second section is converting xyz carriage position into xyz location of the end effector.

Deriving the first set of equations is relatively simple. The only mathematical knowledge needed is length of a line in 3 dimensions and similar triangles in 3 dimensions. Equations 1 and 2 show these respectively. These calculations are done for each tower (three in total).

The second set of equations is slightly more involved. Instead of looking at the arms as lines and needing to calculate angles of intersection, imagine them as spheres with the radius being the arm length. The problem has now become the intersection of three spheres. This problem can either have 0, 1, 2 or infinite number of solutions. Due to the constrained nature of the printer it is a safe assumption that there will always be two solutions. The correct solution to choose will be the one with the lower z value. Figure 7 amply demonstrates the reason for this. To reduce the work a python symbolic numeric equation solving library, SymPy, was used.

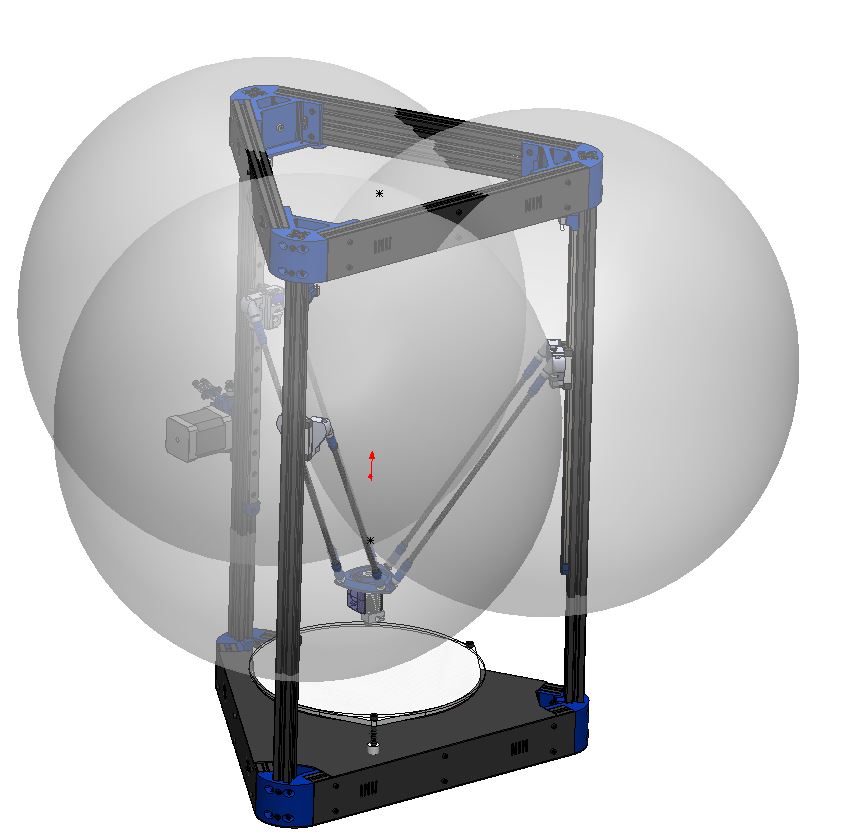


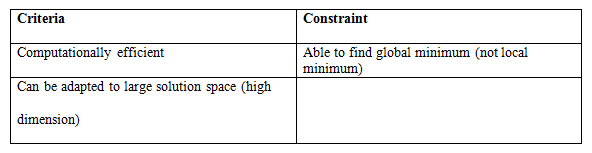
Figure 7: Visual Demonstration of the intersection of 3 spheres

## 

## Optimization Algorithm Selection

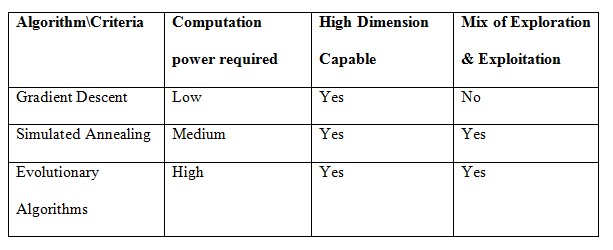
With the completed kinematic eqautions the project is now an optimisation problem. There were several criteria used when selecting an appropriate algorithm. First off, the algorithm should be relatively cheap computationally, since eventually the peoject will be ported to a developed system. Secondly there are at least 12 variables that need to be optimized. The algorithm must be able to handle a solution space of this size. Thirdly the algorithm needs to have a good mix of exploration and exploitation. This means the algorithm must be able to find a global minimum, not just a local one. A summary of the criteria and constraints is presented in Table 1, below.

Table 1: Criteria and Constraints



There were three main algorithms that were good candidates for this problem. Research was done on each and and the most suitable was selected. The three algorithms considered were simulated annealing, evolutionary algorithms and hill climb algoritthms. Evolutionary algorithms are very powerful, but can be very memory intensive. Hill climb algorithms are efficient and simple, but rely almost entirely on exploitation. It will almost cetainly get immediately caugth in a local minimum. Simulated annealing is a reasonable midpoint between the two. Table 2 outlines the decision process and what algorithms were considered [7],[8],[9].

Table 2: Algorithm Selection



The algorithm that was selected is simulated annealing. The general concept for simulated annealing is quite simple and will be outlined here. First, the parameters have some starting point. In this case this is our best guess as to what the printer parameters actually are. The parameters are randomly varied and the cost of the new function is calculated. The cost is the root mean squared delta between the target end effector position and the new end effector position. Depending on the cost of the new parameters there is a chance that they will be accepted, otherwise they will be rejected. During the early iterations it is very likely that the new parameters will be accepted, even if they cost more. The more the algorithm progresses the less likely it is that a move that costs more will be acccepted. This results in an algorithm that explores more of the solution set before honing in on an optimal solution [8]. This procedure is outlined in Figure 8.

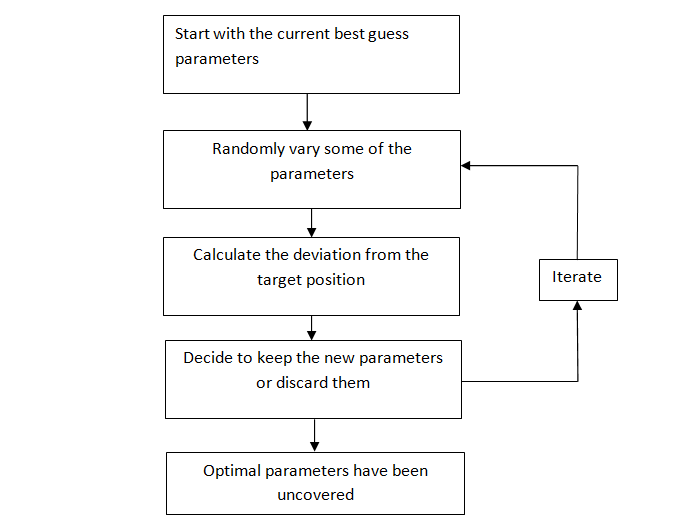


Figure 8: Calibration Procedure Flowchart

The algorithm is called simulated annealing because it emulates the annealing process, where a metal is heated and then cooled to reorient the crystal structure and change the material properties. The slow cooling gives the metal time to orient itself in the ideal crystal structure.

# Analysis of Results

Simply implementing an algorithm is not enough to reliably converge on a correct solution. Numerous issues such as over fitting to data or poor selection of minima will be present if the algorithm is not properly tuned. Tuning is discussed in the following sections.

## Algorithm Tuning: Number of Data Points

This section explores how the number of data points affects the quality of the solution. This was entirely tested in software. The test procedure was as follows:

1. Generate the “correct” model for the printer. Choose a carriage position. Determine the end effector location based on the selected carriage position.
2. Modify the printer parameters to emulate a printer that is out of calibration. Run the calibration with the selected data points and compare the result to what the printer was actually set to.
3. Re-run steps 1 and 2, but choose more carriage positions. Compare how increasing the number of datapoints affects the correctness of the solution.

Figure 9 shows the results of these experiments. It is clearly visible that too few datapoints is a problem since the algorithm tunes the printer to be accurate at these positions, but will likely be even further out of calibration at other positions. For example, the calculated cost (deviation in end effector positions) is very small when there are few datapoints, but the calculated model is very different then the correct model. As the number of datapoints is increased the fitness of the solution tends to improve. Due to constaints when implementing the algorithm on the actual printer the number of points selected should not be excessively large. Ten points appears to be a reasonable comprimise.

Figure 9: The effect of the number of data points

## Algorithm Tuning: Cooling Time

In order to assure the algorithm is operating at maximum efficiency, the cooling speed should be tuned. For example, cooling too quickly will result in not enough exploration of the solution space and will be suboptimal. Conversely, cooling too slowly may make the solution diverge too far and will be computationally expensive.

The cooling time of the algorithm was varied and run the algorithm was run multiple times on the same set of data. Figure 10 visualises the data that was gathered. When the cooling time is too long, the algorithm diverges too far and is very computationally expensive. When the cooling time is too short the algorithm does not do enough exploration and the solution found is not optimal. 1300 iterations appears to be the ideal balance. Of interest, note the characteristics of a simulated annealing algorithm: a highly explorative stage to start, eventually homing in on a correct solution.

Figure 10: Effects of Cooling Time of Global Minimum Finding

## Discussion of Results

Once the algorithm had been selected two procedures were implemented in order to verify its functionality and improve the results generated. The first step was to select the correct number of data points. This is important because it greatly affects the accuracy of the result and the computation time required. Ten points was seleted as being the ideal balance. The second step was to tune how long the algorithm was run. There needs to be enough iterations to fully explore the solution set and hone in on the global minimum. It was shown that 1300 iterations is the ideal number of iterations.

Running the algorithm multiple time on the same dataset and analysing the consistency of the results is a good method of determining whether or not enough exploration has been done. If the algorithm does not consistently arrive at a similar solution it has not done enough exploration. Figure 11 shows an algorithm that has been run multiple times and the solution is consistent. Note how there is divergence to start, but all the solutions eventually converge.

Figure 11: Test of Algorithm Consistency

# Conclusions

Small mechanical misalignments in the physical construction of the printer was shown to severely impact the accuracy of the printer. Such misalignements are essentially impossible to measure by hand, but can be simulated in software. The goal of this project was to build a program that would be able to recreate an accurate kinematic model of the printer. The mechanical misalignments do not need to be corrected, they simply need ot be accounted for in software.

A more complete kinematic model of the 3D printer was derived, verified against a solidworks model and shown to be correct. This model has far more flexibility then the default kinematics and is able to simulate far more of the misalignments that may be present in a physical printer. The model would need to be heavily optimized before being suitable for deployment in an embedded system.

A series of algorithms were considered, one was selected and tuned to optimize the multivariable kinamtic equations. Spending an appropriate amount of time in the algorithm selection and planning stages greatly benefited the final quality of the project.

Testing was not done with a real world printer to fully asses the effectiveness of the autocaliibration suite. Currently such testing requires precise measuring tools which defeats the purpose of the project. Before the project can be considered a success a new method of gathering data for calibration must be determined.

# Recommendations

The most important reccomendation is to improve the data gathering process for actual printers. Reliable data is crutial for calibration. Since there is currently no easily acheivable method of gathering data the project cannot be considered entirely complete and its usefulness is severly limited.

Further more, intelligent and optimized algorithm tuning would help the quality of the final result. The manual tuning that was done was very beneficial, but there is still room for improvement.

Moving away from numerical solwing of the kinematics and properly solving the equations with calculus will greatly decrease the computation time and thus will allow for more freedom in optimization algorithm selection.

Improvements on the usability of the software would allow it to be released publically. Currently the software is in a state of research and development quality. The ability to gather larger sets of data from people around the world would unlock further potential for more complex calibration routines.

# References

[1] Evangelista, Sandy. “The Delta robot – Swiss-made and fastest in the world!” . News

MediaCom. Web. 14 November 2011. <https://actu.epfl.ch/news/the-delta-robot-swiss-made

and-fastest-in-the-worl/>. 7 May 2016.

[2] “ORION DELTA™ DESKTOP 3D PRINTER”. SeeMeCNC. Web.

<https://www.seemecnc.com/collections/ready-to-print-3d-printers/products/orion-delta-3d

printer>. 7 May 2016.

[3] “2-4 MG Series - Miniature Linear Guideway”. HIWIN. Web.

<<http://motioncontrolsystems.hiwin.com/Asset/MG-Series-Catalog.pdf>>. 5 May 2016.

[4] “Precision Hardened Balls - Metric / Inch”. MiSUMi USA. <http://us.misumi

ec.com/vona2/detail/110300262010/>. 3 May 2016.

[5] “ABSOLUTE Digimatic Height Gage Series 570-with ABSOLUTE Linear Encoder”

Mitutoyo USA. Web. <http://ecatalog.mitutoyo.com/ABSOLUTE-Digimatic-Height-Gage

Series-570-with-ABSOLUTE-Linear-Encoder-C1270.aspx>. 3 May 2016.

[6] “SURFACE PLATE 12IN. X 18IN. X 3IN” BusyBee Tools. Web.

<http://www.busybeetools.com/products/surface-plate-12in-x-18in-x-3in.html>. 3 May 2016.

[7] Ruder, Sebastien. “An overview of gradient descent optimization algorithms”.Web.

<http://sebastianruder.com/optimizing-gradient-descent/>. 8 May 2016.

[8] Geltman, Katrina Ellison. “The Simulated Annealing Algorithm”. Web. 20 February 2014.

<http://katrinaeg.com/simulated-annealing.html>. 1 May 2016.

[9] Smith, Eiben. “What is an Evolutionary Algorithm?”. Web.

<http://www.cs.vu.nl/~gusz/ecbook/Eiben-Smith-Intro2EC-Ch2.pdf>. 1 May 2016.

[10] “Google Trends”. Google Inc. Web.

<https://www.google.ca/trends/explore#q=delta%20printer>. 1 May 2016.

[11] “Why a Delta?”. Build a 3d Pritner.eu. Web. <http://builda3dprinter.eu/information/why-a

delta/>. 1 May 2016.

[12] “3D Printing Trends May 2016”. 3D Hubs. Web. <https://www.3dhubs.com/trends>. 1 may

2016.