

MODELING OF THE BROACHING PROCESS

Sara Whitby

Graduate Student

Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 1521

Matthew Glisson

Graduate Student

Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 1521

Anton Galkin

Graduate Student

Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The goal of this paper is to describe the operation and design of a computer tool that was created to help user's model new broaching tools. The difficulty in designing new broaching tools is that there is a very high cost per tool and, once created, there is very little flexibility in how the tools is used since only the only parameter that can be changed is the tool velocity. For these reasons it is important to design a broaching tool correctly.

The program that is presented here can take an initial profile and a desired final tooth profile and use them to create a series of intermediate tooth profiles based on specifications given by the user. The tool also models the forces on the teeth with both mechanistic and energy based equations. This allows the user to quickly and easily gain a good understanding of the forces that are at work during the cut. A simple user interface which gives the user control over a wide range of variables makes the program a useful tool for the early design stages of a new broach.

A. Introduction

Broaching is the process by which a single tool with multiple teeth is pushed through a hole or slot. Each successive tooth on the tool gets larger and closer in shape to the desired final profile of the hole.

The major benefit of broaching is that it is a very quick manufacturing method: the entire cut often takes less than a few seconds and can go from an unfinished hole to a hole with a complicated profile in one pass. In some cases it is the only process available to reliably create such complex interior geometries. Also, once the broaching tool is designed and setup each cut costs comparatively little and requires few operator adjustments. It is for these reasons that broaching is an attractive option for large scale, mass manufacturing operations.

The downside of broaching is that tool design can be costly and time consuming. The tool itself can only be used for the specialized cut it has been built for. Unlike standardized, interchangeable drills and milling tools, broaches must typically be custom-made.

For these reasons it is important for engineers to have the proper tools available to them while designing broaching tools. Since broaching is made costly primarily due to the creation of the tool, allowing a toolmaker to rapidly explore the design space by automating the computationally intense parts of the design process and iterating through a multitude of tool configurations would save time and money. The tool we created assists in the process of engineering broaches by giving engineers a way to quickly and easily see the forces experienced by different broach configurations early in the design process.

B. Related Research

Much research has been done into the mathematical modeling of the broaching process, as well as some research into computer modeling using FEM and even MATLAB. For example, in Vijayaraghavan [1], the stresses and displacements in a broach were analyzed using computer-based FEM for several rake angles, concluding that it is advantageous to use a tool with a zero degree rake angle to minimize cutting edge displacement for workpieces that offer a larger resistance to deformation. This same paper also discusses how to use FEM to evaluate the depth of the work-hardened zone of the workpiece. In Shi [3], an online machining process monitoring system is presented, composed of sensing, triggering, data acquisition, characterization, condition monitoring and feature extraction packages. Kokturk [4] discusses a methodology for designing optimal broaching tools by concentrating on the geometric and physical constraints. Computer software was written that implemented this methodology. The main goal was to optimize the broaching procedure primarily with respect to tool length; it seeks to create the shortest (and theoretically cheapest) tool possible.

Probably the research papers most relevant to our project are those by Ozlu [5] and Ozturk, Ozkan, and Erhan [2]. The former [5] proposes a series of algorithms that will simulate the broaching process. A computer software package makes use of the algorithms to calculate the physical parameters of the process and predict performance measures. The simulation can calculate cutting forces, power, maximum stress, chip thickness, rake angle distribution, and evolution of the work piece geometry. The latter [2] presents mathematical tool optimization methods and process models. Cutting forces, tooth stresses, and part deflections are modeled and analyzed using cutting models and FEA. Simulation was done with a MATLAB program, with the paper giving a simplified algorithm. The program outputs predicted forces, tooth stresses, and part deflections with inputs of material characteristics, tool and part geometry for two modes, with and without consideration of deflection effects on cutting force calculations. Simulation is carried out over time where the tool is advanced into the material in small increments.

C. Program Description and Walkthrough

C.1 The Graphical User Interface

The program is divided into four sub programs. The first program runs the Graphical User Interface; this is where the user can input the parameters of the broach tool, cutting process, and work piece. The GUI program takes the user input and passes it to the other three sub-programs. Since the GUI program calls the other sub-programs and passes information

between them it is also considered to be the master control program of the tool. A flowchart detailing the interactions of the sub-programs can be seen in Fig. C5.

The GUI is designed to provide the user a location where he can quickly and easily adjust a number of different parameters related to the broaching tool (Fig C3). Having all the parameters laid out in one place is important if the tool is going to be easy to use and help engineers examine many different broaching tool models in an efficient manner.

C.2 Tooth Profile Finder

When the GUI/Master program is run (via the calculate button on the interface) the first sub-program it calls is the Tooth Profile Finder (TPF). The GUI passes the TPF the radius of the initial profile, the image of the final tooth profile, and the resolution of the image (in mm/pixel). The TPF looks through the image of the final tooth profile and stores a series of coordinates recording where the black pixels in the image are. The program then fits a polynomial function to both the X and Y coordinates of the pixels. Relating the functions together in terms of a third variable gives the TPF a continuous, smooth line to work with (Fig C1). The line is scaled to match the size of the initial profile based on the resolution of the image.

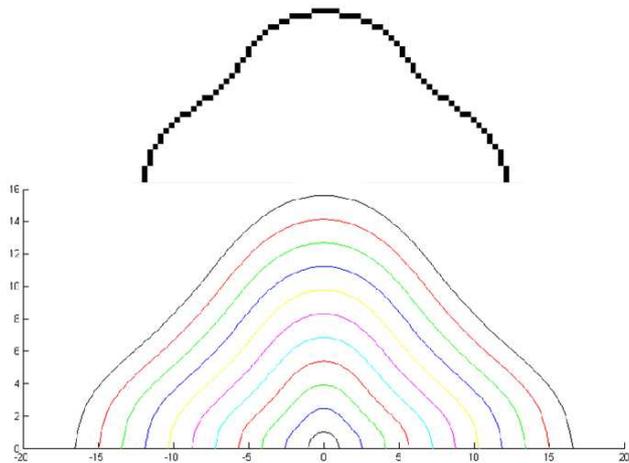


Figure C1: The Tooth Profile Finder uses a png image file (top) to generate a curve representing the desired final tooth profile. Then, using the number of teeth the user wants in their broach, the program creates a series of intermediate tooth profiles between the initial and final profile (bottom).

Next the program creates evenly spaced lines that radiate from the origin and extend from the initial profile to the poly fit line of the final profile. Based on the number of teeth the broach is supposed to have the radiating lines are broken up into points that are spaced evenly between the initial and final profiles. These points are then used to create the intermediate teeth profiles.

To find the uncut chip area of a given profile the TPF uses the points that make up the profile and the points that make up the previous profile to create a series of quadrilaterals, where each quadrilateral is made up of two points from each profile (Fig C2). The TPF program finds the area of each quadrilateral

on each profile and then returns the data as a matrix back to the GUI. It also displays the tooth profiles for the user (Fig C1).

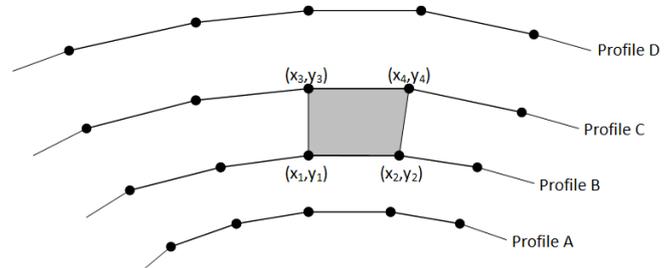


Figure C2: The Tooth Profile Finder uses points spaced along a given profile (here, Profile C is used) and the previous profile (Profile B) to make a series of quadrilaterals. Taken together, these quadrilateral approximate the uncut chip area for the given profile (Profile C). Note that the approximation gets more accurate as more points are used along a profile.

C.3 Individual Tooth Force Calculator

Next, the GUI passes all of the parameter values the user entered into program to the Individual Tooth Force Calculator sub-program (ITF). The GUI also passes along the area of each quadrilateral in each tooth profile.

The ITF treats each quadrilateral as a single tooth with the rake and inclination angles specified by the user. Using the information the GUI gave about the cutting process and workpiece the sub-program can find the cutting, thrust, lateral and machining forces on each quadrilateral. The ITF can use either energy or mechanical methods to find the forces, this is explained in more detail in section D. The ITF returns to the GUI the forces found on each quadrilateral and totals the forces found on the first and last tooth profile. Graphs showing the different profiles (a graph is created for cutting, thrust, lateral, and machine forces) is returned to the user.

C.4 Force Over Time Calculator

Next, the GUI passes parameters about the cut and force values for each quadrilateral to the Force Over Time sub-program (FOT). The FOT sub-program calculates the total time for the cut and which teeth are engaged with the workpiece at regular intervals throughout the cutting process. Since the sub-program knows which teeth are engaged at a given time and what forces those teeth experience it can plot the total cutting, thrust, and lateral forces experienced by the tool at any given moment.

A feature of the program that is worth noting is the ability to have the program automatically alternate the inclination angle of the teeth. For example, if the user wanted the teeth on a broaching tool to have an inclination angle of 20 degrees and activated this feature then the teeth on the broach would alternate between a 20 degree and -20 degree inclination. The alternating angle option is useful because it provides a simple way to drastically reduce the total lateral force on the tool. Once all the teeth are engaged with the workpiece the alternating lateral forces cancel each other out and keep the total lateral force closer to zero (Fig. C4).



Figure C3: The GUI can control various parameters about the teeth (blue section) as well as parameters about the cut and workpiece (green). In addition, the user can specify the image file and resolution being used for the final tooth profile (brown). Finally, if the user desires, they can bring up the force profiles for a specific tooth in the broaching tool (purple).

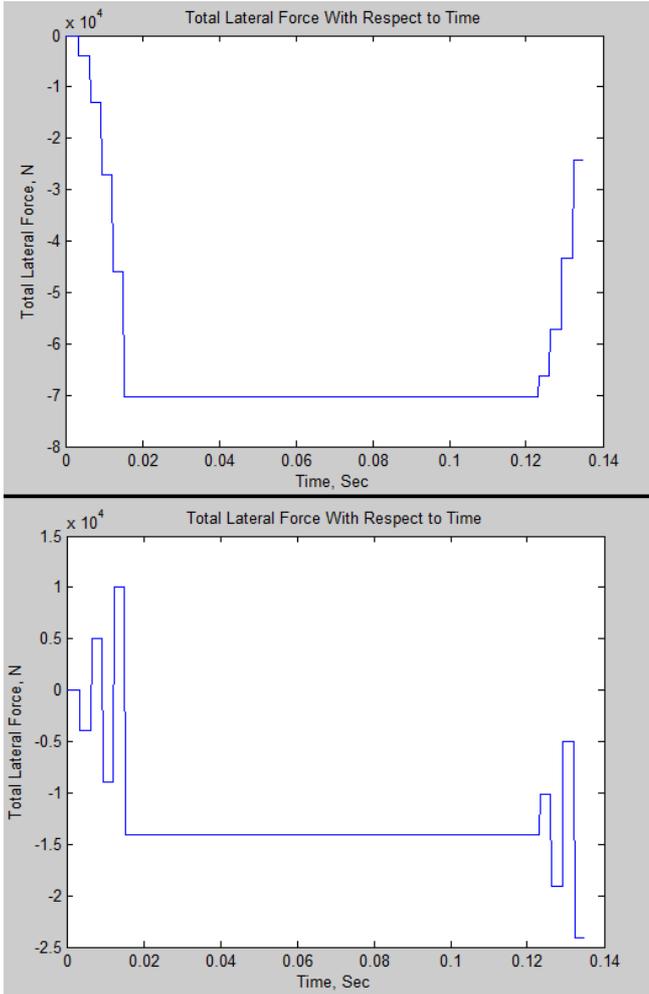


Figure C4: The total lateral force on two different five tooth broaching tools is shown over time. The broaching tools were identical (10 degree rake angle, 20 degree inclination angle, teeth spaced 5 mm apart) except for the broach for the top graph kept the same inclination angle while the tool for the bottom graph alternated the inclination angle. Note that the lateral force is, in general, much lower for the second tool.

C.5 Individual Tooth Force Profile Calculator

Finally, if the user desires, the GUI can print the thrust, cutting, and lateral force profiles over the perimeter of a selected tooth. The GUI uses the output from the Individual Tooth Force calculator to graph the requested data. The ITF has done the bulk of the work to find the force profiles since it has found the forces on each individual quadrilateral. The GUI handles the rest of the work and just totals the forces on the quadrilaterals from the requested tooth.

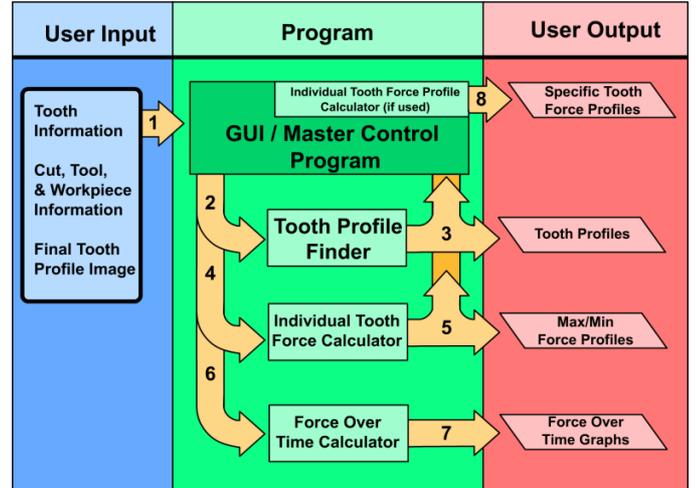


Figure C5: The flow of programming in the broaching tool is mapped out. First the user inputs the desired parameters, the GUI/Master Control Program runs the appropriate sub-programs, and finally the calculated data is displayed to the use.

D. Technical Approach

Two methods were used to find the forces on the teeth in the broaching model, a mechanistic approach and an energy based approach. Both methods are available for use with the program.

D.1 Mechanistic Approach

The equations used in the calculation of forces and geometry in our model are well known oblique cutting equations, along with a few base assumptions:

- Material at the shear plane is yielding, and thus at its maximum shear yield stress: $\tau_{\phi} = S_y/2$
- Shear angle is calculated based on the Lee & Schaffer model: $\phi_0 = \frac{\pi}{4} + (\gamma_0 - \beta_a)$
- Stabler's Rule for chip flow angle is in effect: $\eta_{\gamma} = \lambda$

From there, we derive relevant geometric relations including rake angles, shear angles and chip flow angles in normal and orthogonal planes:

- $\gamma_n = \tan^{-1}(\tan \gamma_0 \cos \lambda)$ (normal rake)
- $\phi_n = \tan^{-1}(\tan \phi_0 \cos \lambda)$ (normal shear)
- $\eta_{\phi} = \tan^{-1}\left(\frac{\tan \lambda \cdot \cos(\phi_n - \gamma_n) - \tan \eta_{\gamma} \cdot \sin \phi_n}{\cos \gamma_n}\right)$

The chip area can be determined from the given uncut chip width and depth of cut:

- $a = h * w$

Using our assumption of yielding at the shear plane, the first force calculated is the orthogonal shear force:

- $P_{\phi} = \frac{a * \tau_{\phi}}{\sin \phi_0}$

The normal rake face friction force is calculated using the friction angle:

- $\beta_a = \tan^{-1} \mu$
- $N_{\phi} = P_{\phi} * \tan(\phi_0 + \beta_a - \gamma_0)$

From this, we can find the overall machining force using a well understood equation:

- $$R = \frac{P_\phi}{\cos \theta_\lambda * \cos(\theta_n + \phi_n) * \cos \eta_\phi + \sin \theta_\lambda * \sin \eta_\phi}$$

Finally, the machining force can be split into its constituent parts along the cutting, thrust and lateral directions:

- $F_C = R * (\cos \theta_\lambda * \cos \lambda + \sin \theta_\lambda * \sin \lambda)$
- $F_T = R * (\cos \theta_\lambda * \sin \theta_n)$
- $F_L = R * (\sin \theta_\lambda * \cos \lambda - \cos \theta_\lambda * \sin \theta_n)$

This mechanistic force model is very general and could be applied to any oblique cutting problem. We used a numerical method for our approach, dividing each profile into discrete areas. The forces for each small area is calculated, recorded, and can be manipulated in various ways:

- Summation to find total force on one tooth profile
- Direct graphing to show how forces on an individual tooth change along its profile
- Optimization to find maximum and minimum forces on a tooth and identify areas of highest material stress

Our model makes use of this general mechanistic force model by calculating the chip area in a unique way. Each tooth experiences a force based on its associated chip area (Fig D1).

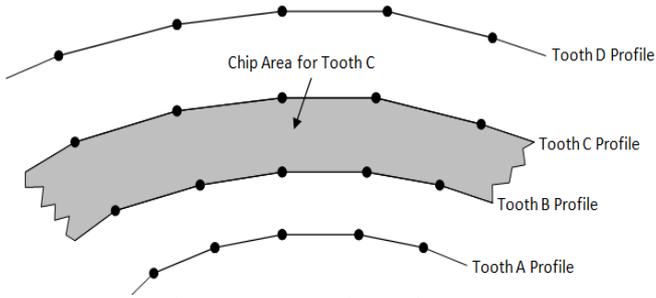


Figure D1: Chip area for a given tooth profile (Profile C)

In our numerical approach, each profile is divided into a series of points. As such, we calculate the chip area of each small section by finding the area of the irregular quadrilateral connecting two consecutive points on two consecutive profiles (Fig D2).

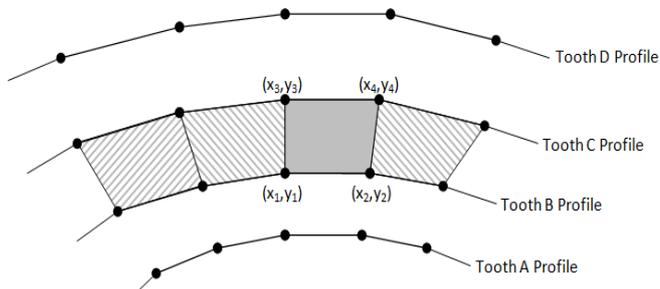


Fig. D2 Irregular Quadrilaterals connecting consecutive points on two adjoining profiles (Profile C and Profile B)

From this, we can find an equivalent depth of cut by averaging the distance between points 1 and 3 and points 2 and 4. The equivalent width of cut is thus the area divided by the depth of cut. We feel this is a good approach to determining the

chip area, much like the mesh formed on a solid body during a finite element analysis. Alternate approaches involving the calculation of depth of cut orthogonally to the profile proved problematic for a number of reasons:

Firstly: small, sharp turns in the final profile often result in 45° or 90° angles between consecutive data points on a single profile, yielding wildly variable and erroneous chip thickness.

Secondly: any formulation using this approach resulted in consistent overestimation of numerical chip area, yielding a compounding error that reduced confidence in our output.

For curved, indeterminate profiles, such an approach is difficult to implement numerically, and may best be used in a more analytical analysis of chip area for such problems.

D.2 Energy Model Approach

We compared our mechanistic model to one using specific energy to determine the forces. This approach used many of the same base equations as the mechanistic one, with the key difference that the forces are not based on the assumption of yielding at the shear plane. Instead, the first forces calculated are in the cutting and thrust directions:

- $F_C = u_c * a$
- $F_T = u_T * a$

The specific energy equations used were adapted from example problems as a way to compare the results of the two approaches. Though these equations are typically experimentally derived for particular combinations of workpiece and tool material, it is posited that they still hold some validity in the general case. These specific energy equations are taken from HW#2, Q5:

- $u_c = 3150h^{-0.233}V^{-0.122}e^{-1.318\gamma_n}$
- $u_T = 1175h^{-0.615}V^{-0.148}e^{-1.115\gamma_n} F_T = u_T * a$

From the cutting and thrust forces, all other forces can be derived in a similar manner to the mechanistic model approach.

D3. Model Validation

To confirm the validity of our model functions, we used a set of existing oblique and orthogonal force calculations pulled from available homework solutions. We compared our model's solution with the known value and calculated percent error (Table D1).

Table D1: Results of Our Oblique Cutting Models Compared With Known Homework Solutions

Source	Model Result	Known Value	Percent Error
HW#1, Q3	$F_C = 531.4747N$	$F_C = 531.5N$	0.00476%
HW#1, Q3	$F_T = 134.7577N$	$F_T = 134.8N$	0.03138%
HW#2, Q1	$P_\phi = 468.7500N$	$P_\phi = 470N$	0.26596%
HW#2, Q1	$P_\gamma = 305.7179N$	$P_\gamma = 305.7N$	0.00586%
HW#2, Q3	$\gamma_n = 8.6822^\circ$	$\gamma_n = 8.682^\circ$	0.00230%
HW#2, Q5	$F_C = 394.5634N$	$F_C = 395.29N$	0.18381%
HW#2, Q5	$F_T = 232.8983N$	$F_T = 233.27N$	0.15934%

After putting the model through its paces, we determined that it was properly calculating the appropriate angles, forces and velocities.

E. Results

E.1 Varying images

The program was tested in various ways to demonstrate the types of results that can be gotten for different images, inputs, and methods of force calculation. The following are examples of the output for three different input profiles with the same input values. The input values are the same as those listed in Figure C2, except for the image name. The results can be seen in the figures below. These results were found using the mechanistic method of force calculation.

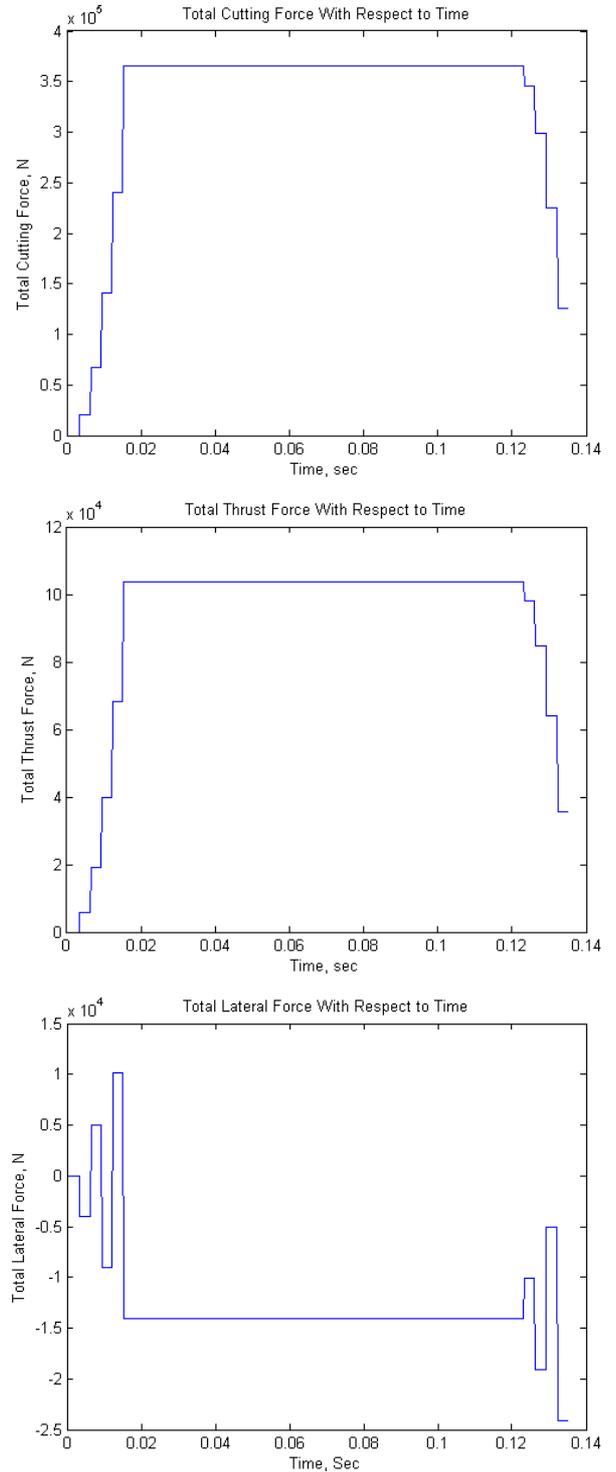
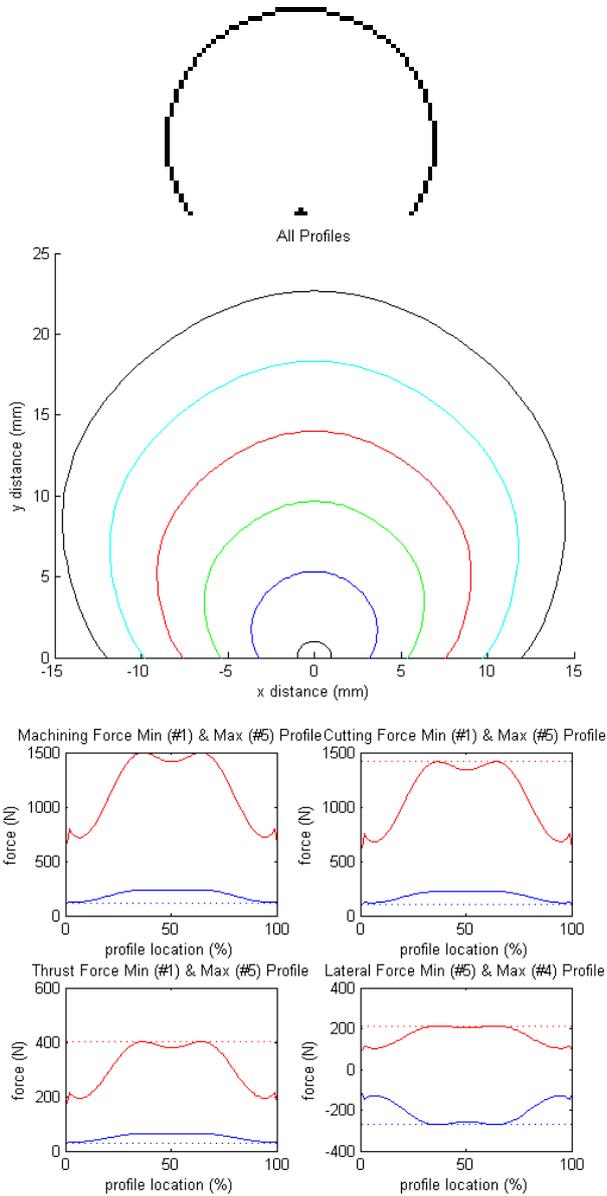


Figure E1: Results for Test Profile 1. From top to bottom: (a) The original outer profile input image (b) All of the profiles given and generated for the given outer and inner profiles, (c) The force distributions for cutting, thrust, lateral, and machining forces on the first and last teeth, (d) The total cutting force for the tool over time, (e) The total thrust force for the tool over time, (f) The total lateral force for the tool over time.

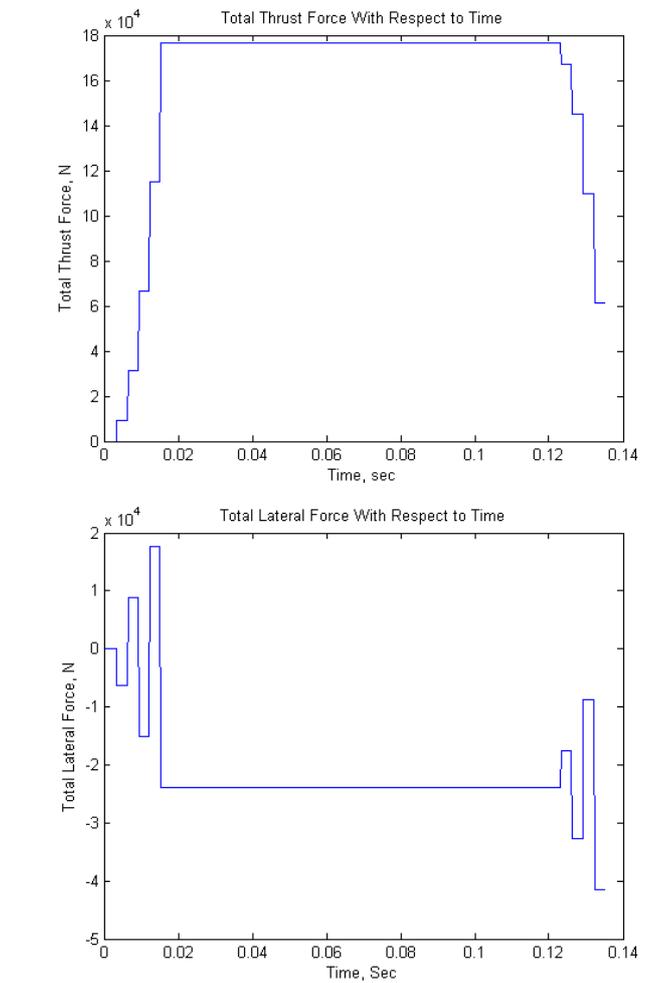
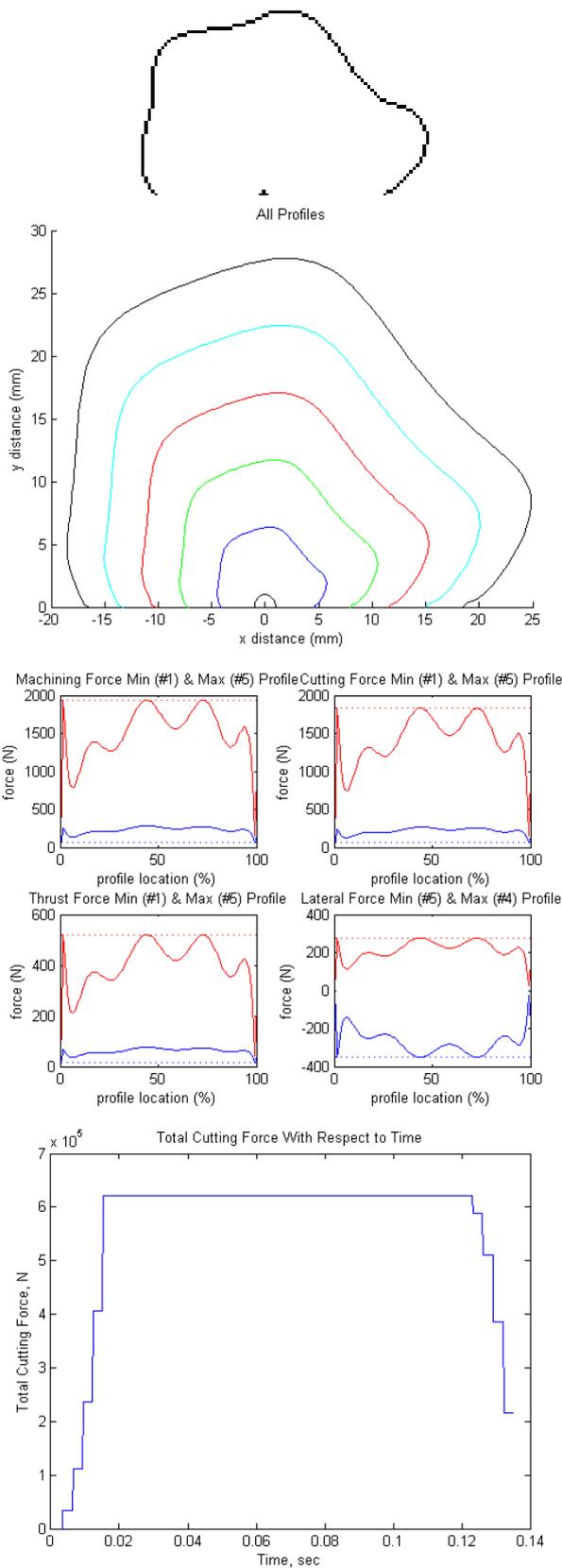


Figure E2: Results for Test Profile 3. From top to bottom: (a) The original outer profile input image (b) All of the profiles given and generated for the given outer and inner profiles, (c) The force distributions for cutting, thrust, lateral, and machining forces on the first and last teeth, (d) The total cutting force for the tool over time, (e) The total thrust force for the tool over time, (f) The total lateral force for the tool over time.

E.2 Varying Inputs

We also tested several different changes in the input parameters for the same outer and inner profiles. In this case, the outer profile is the same as that used in Figure E1 above. The following images show the output that changes with the stated change in input. We only give a few examples here, but the program can be used to easily compare the effects of any small changes on a given input configuration. These results were found using the mechanistic method of force calculation.

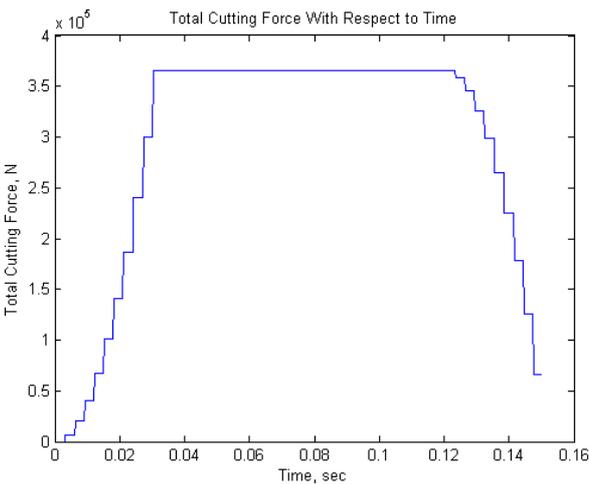
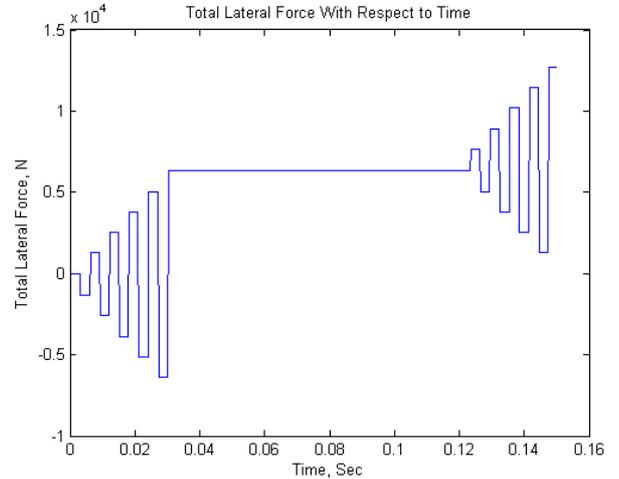
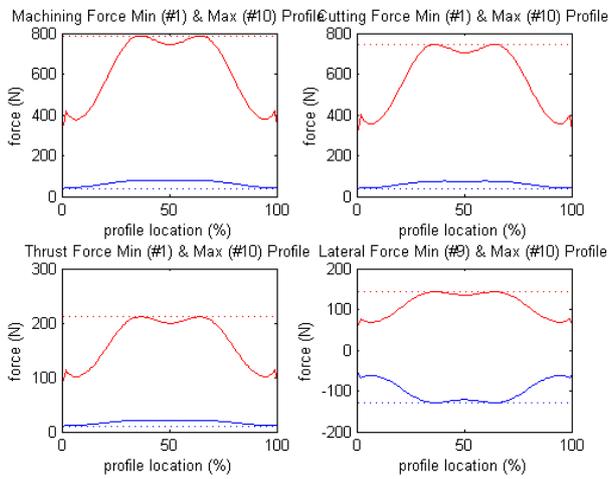
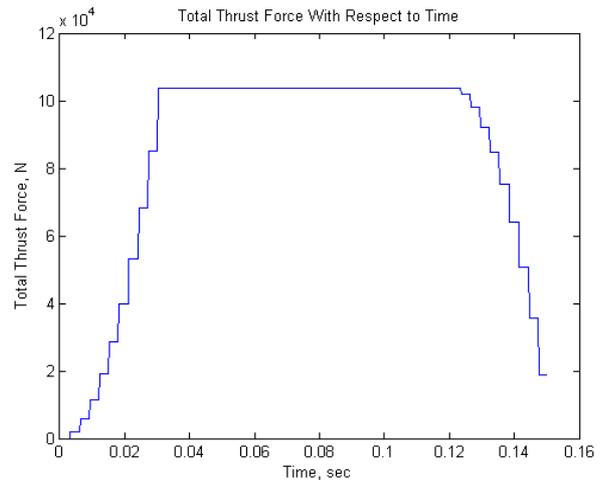
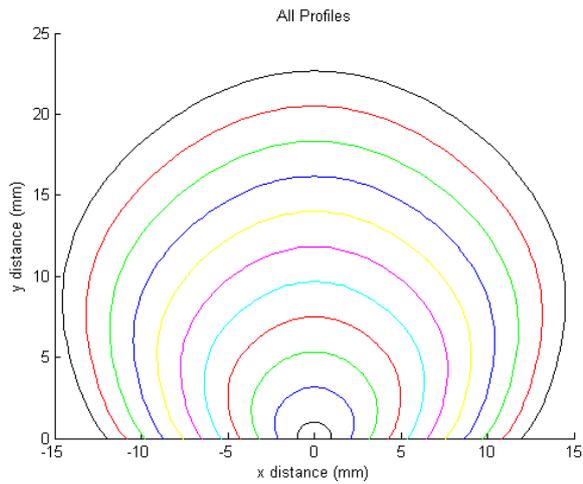


Figure E3: Results for Test Profile 1, with number of teeth increased from 5 to 10. From top to bottom: (a) All of the profiles given and generated for the given outer and inner profiles, (b) The force distributions for cutting, thrust, lateral, and machining forces on the first and last teeth, (c) The total cutting force for the tool over time, (d) The total thrust force for the tool over time, (e) The total lateral force for the tool over time.

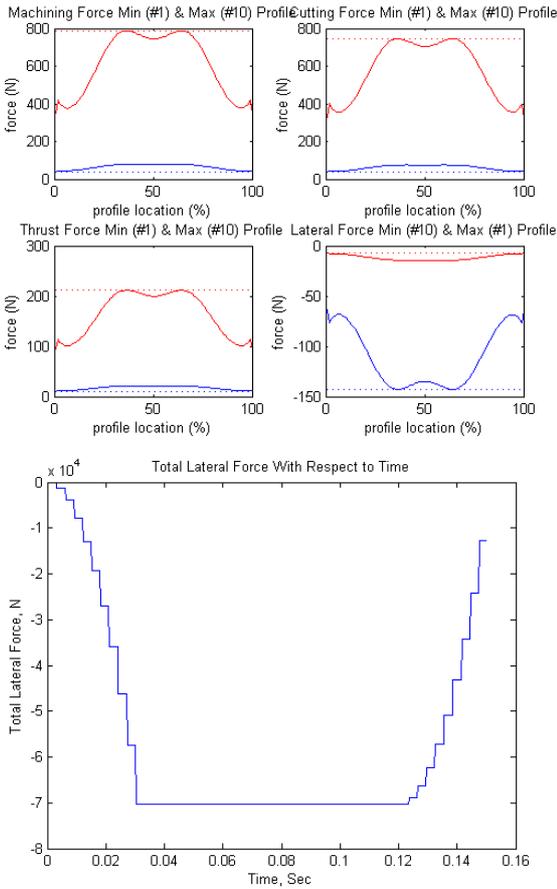


Figure E4: Results for Test Profile 1, with 10 teeth and without alternating inclination angles. From top to bottom: (a) The force distributions for cutting, thrust, lateral, and machining forces on the first and last teeth, (b) The total lateral force for the tool over time.

E.3 Varying Methods

We also made a version of the program capable of using energy methods for force calculation. The following example is the same in every way as that in Figure E1, but the forces were calculated using energy methods.

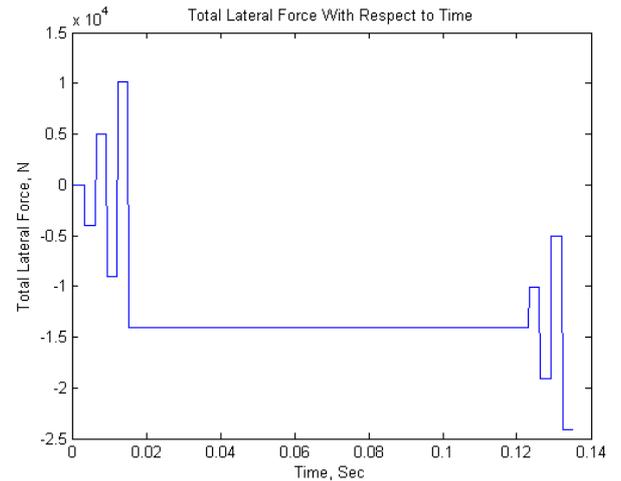
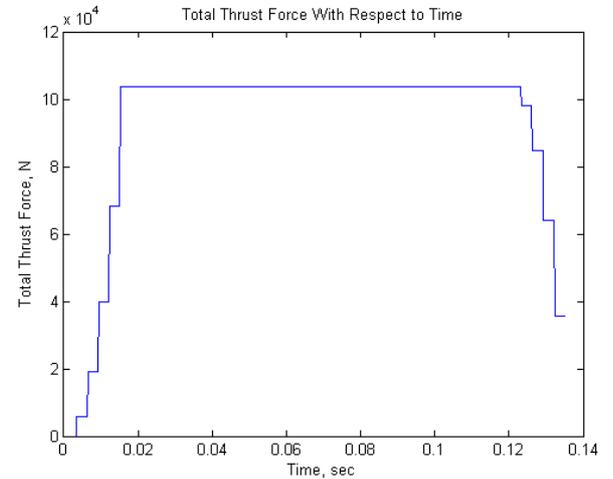
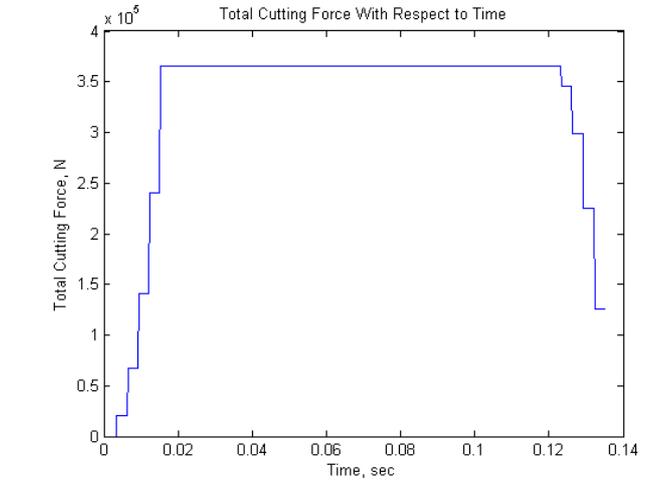
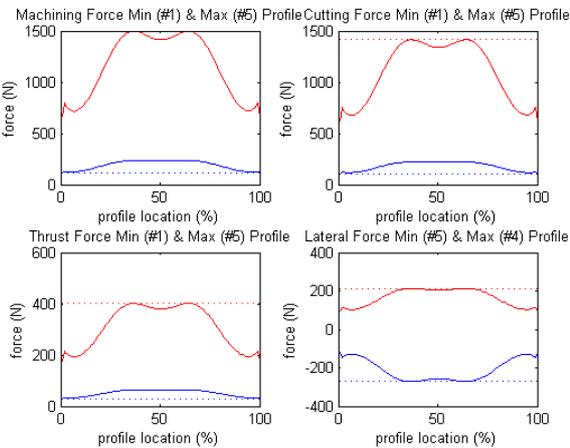


Figure E5: Results for Test Profile 1 using energy methods. From top to bottom: (a) The force distributions for cutting, thrust, lateral, and machining forces on the first and last teeth, (b) The total cutting force for the tool over time, (c) The total thrust force for the tool over time, (d) The total lateral force for the tool over time.

F. Discussion

Through the above tests, we discovered a few problems and limitations with our program. First of all, the graphs showing the force distributions over the individual teeth are somewhat incorrect with this implementation. The way it should work is that each small area of the profile should have the same effective chip width with the total chip area varying between areas because of a change in effective uncut chip thickness. However, at this point, the distances between points along a profile are related to the distances between the pixels in the image. Therefore, when the profile image has a straight line, the points are closer together than they are when the line is diagonal in the image. This causes profiles such as the circular one in Figure 4 to have a decrease in force at the top of the profile because of the smaller chip area due to smaller chip width, where this should actually be the part of the profile with the largest force for a given small chip area. This could be fixed by, after fitting a curve to the profile, redistributing the points along the curve in such a way that all of the points where the same total distance apart.

Additionally, the intermediate profiles are currently positioned based on the distance between corresponding points on the starting and ending profiles. Another option would be to position the intermediate profiles in such a way that the total chip area of every profile is equal. This would cause the total force added by a single tooth to be the same as the next such that there was no large jump in total forces on the tool.

There is another limitation in that there is no evidence of the gradual entry of an oblique tooth. This model shows the force over time graphs having the force simply jump up by a total when the next tooth enters the workpiece. This would be the case with an orthogonal tooth, or if you input an inclination angle of zero into our program. However, in reality one side of the profile would enter first, such that the forces would gradually increase as a result of that particular tooth entry until the whole tooth had passed into the workpiece. This limitation prevents us from seeing what the force over time would be like for some interesting cases, such as if the distance between two oblique teeth facing the same way was small enough that the second tooth started to enter the workpiece before the first tooth was all the way in.

Also, at this point, the outer profile is simply being fit to a polynomial curve using MATLAB's built in polyfit function. However, there are many ways to do curve and point fitting for such an application. Using polyfit, any sort of sharp curve or corner does not turn out very well, changing the shape significantly from the intended profile. This could probably be fixed by some research and experimentation with MATLAB's curve fitting utilities and other mathematical curve fitting algorithms.

For the specific energy approach, we use a given set of equations for specific energies. These equations are experimentally derived from a relationship between two specific materials. Therefore, these equations will vary for any set of tool and workpiece materials. It would be good to have some way of inputting these equations such that the pair of

materials can be changed. Also, as of this point, changing the velocity in the mechanistic approach has no effect on forces. This is because the forces are calculated based on the given yield stress. In reality, there is a relationship between the velocity and the yield stress that our program does not take into account. This relationship is also material specific. That is why velocity affects the forces for the energy model and not for the mechanistic model.

G. Summary and Conclusions

This paper discussed the creation of a program which can simulate a slot broaching process in a wide range of situations. It creates evenly spaced tooth profiles from digital images, and uses these to calculate the lateral, thrust, cutting, and machining forces over the perimeter of a specific tooth, as well as finding the total lateral, thrust and cutting forces over time. It automatically displays the highest and lowest lateral, thrust, cutting, and machining forces over the perimeter the first and last teeth (the minimum and maximum total forces, respectively). The tool can be used when designing a broach to quickly and easily experiment with different tooth profiles and geometries.

Some improvements that could be made to the current program are finding the intermediate profiles based on total chip areas, modeling the forces of gradual entry into the workpiece by an oblique tooth, and fixing the distribution of points for outer profile such that the small chip areas used for the force distribution curves have equal chip widths for a given profile. Also, we could allow for more versatile starting profiles, such that the program could take in an image that was not the specified resolution, it could deal with lines where some black pixels had three neighbors, etc. It could also allow for more flexibility in inputs and outputs. For example, you could input the orthogonal rake angle *or* the normal rake angle, you could input maximum forces and it would give you chip thicknesses and number of teeth, etc. It would also be interesting to use a similar approach to model different kinds of broaching other than just slot broaches, such as internal broaches.

H. References

- [1] Vijayaraghavan, L., R. Krishnamurthy, and H. Chandrasekaran. "Evaluation of Stress and Displacement of Tool and Workpiece on Broaching." *International Journal of Machine Tool Design and Research* 21.3-4 (1981): 263-70. Print.
- [2] Ozturk, Ozkan, and Erhan Budak. "MODELING OF BROACHING PROCESS FOR IMPROVED TOOL DESIGN." *Proceedings of IMECE'03*. Web.
- [3] Shi, D., D. A. Axinte, and N. N. Gindy. "Development of an Online Machining Process Monitoring System: a Case Study of the Broaching Process." *The International Journal of Advanced Manufacturing Technology* 34.1-2 (2007): 34-46. Web.
- [4] Kokturk, U., and E. Budak. "Optimization of Broaching Tool Design." *Intelligent Computation in Manufacturing Engineering*. Sabanci University Research Database. Web. 23 Mar. 2011. <<http://research.sabanciuniv.edu/8226/>>.
- [5] Ozlu, E., S. Engin, C. Cook, T. El-Wardany, and E. Budak. "Simulation of Broaching Operations for Tool Design Optimization." Web.

I. Contributions

- Matthew Glisson – Created the GUI and interfaced all of the subprograms with the master program. Created the subprogram that calculates forces over time.
- Anton Galkin – Helped to create the oblique cutting subprogram that calculates forces based on given chip geometries, etc. Created the subprogram that takes the output from the profile area finder and creates force profiles for a given tooth.
- Sara Whitby – Helped to create the oblique cutting subprogram that calculates forces based on given chip geometries, etc. Created the series of subprograms responsible for reading images into MATLAB, fitting curves, creating intermediate profiles, and finding small chip areas for each profile to use in force calculations.