Proceedings of IMEC 2019

International Mechanical Engineering Congress (IMEC-2019),

29th November- 1st December 2019, NIT Tiruchirappalli, India.

IMEC2019-xx-xxxx

**Analysis of cutting forces in Helical Ball end milling process using machine learning**

Ananth Narayana\*, Naman Yadav, Asim Tewari#

Department of Mechanical Engineering, IIT Bombay

\* Corresponding author email: ananth.na95@gmail.com

+ Corresponding author email: asim.tewari@iitb.ac.in

**ABSTRACT**

In this paper, variation of cutting forces exerted on tool with respect to cutting parameters for ball end milling process is analysed using deep neural network. A neural network was fabricated to predict cutting forces in all three orthogonal directions for a given set of cutting parameters. Analysis was done by varying axial depth of cut and feed-rate while keeping all other parameters constant, because the variation of cutting forces was most significant for these two parameters. For generating data CutPro simulation software was used along with shell scripts to automatically vary the parameters and simulate. Finally, after proper tuning of hyperparameters of the neural network, maximum percent deviation of predicted values over test dataset was brought down to less than 1 percent.

**INTRODUCTION**

Machining is the most widespread metal shaping process in mechanical manufacturing industry. Machining is a subtractive process where a raw material is turned into a desired product by removing the material around it. The study of forces generated during machining, Workpiece and tool material, tool and workpiece geometry, machine parameter settings influence the process efficiency and output quality.

Titanium alloy finds its application in myriad of fields and industries including space, aviation, biomedical, automotive and oil industries [1-3]. The properties such as high strength to weight ratio, high corrosion resistance, low young’s modulus and bio compatibility are exploited to manufacture components for their respective fields. 70% by weight of titanium- based alloys are used in Aero-Space industries. Ti-6Al-4V was introduced for applications demanding exceptional mechanical and chemical properties at high temperatures. Ti-6Al-4V alloy is particularly known to exhibit high strength to density ratio, has good resistance to corrosion, high chemical reactivity, high cutting force generation during machining, low thermal conductivity, high rigidity. These properties render titanium’s machinability very difficult.

Many researchers have studied machining of Titanium alloy and effects of cutting parameters on the cutting forces. [4-10].

A FEM based micro end milling of cutting force model of Ti-6Al-4V was presented by Tej Pratap, Karali Patra, Dyakonov A.A [4]. Slight discrepancies were observed in simulated cutting forces and in experimental results. In Experimental scenario tool wear and tool dynamics caused variation in cutting forces. It was found that increase in number of cutter rotation increases cutting forces.

Experimental study on high speed end milling of titanium alloy was performed by V Krishnaraj, S Samsudeensadham, R Sindhumathi, P Kuppan [5]. It was observed that with increase in depth of cut and feed-rate the cutting forces increased whereas increasing in cutting speed had no effect on tangential or feed forces.

Szymon Wojciechowski, et al [6] proposed a cutting force model for finishing ball end milling of inclined surfaces. Surface inclination angle (α) in the range 0≤ α≤15 showed decrease in absolute value of cutting forces with increase in surface inclination angle. With increase in inclination angle (α) it was observed that the edge cutting coefficients (Kae, Kte, Kte) decreased.

Although research in the area of titanium machining from experimental and physics-based modelling exists, there is very little research in data-driven modelling. Hence, the aim of this work was to perform machining study on Ti-6Al-4V alloys, based on data driven machine learning techniques.

**DATA GENERATION**

For data generation, CutPro software was used to simulate the milling operation for different cutting parameters and for each set of parameters cutting force in three mutually perpendicular directions was obtained. To automate the process in order to generate more training data shell scripting was used. Axial Depth of Cut [ADOC] was varied from 1.0mm to 1.9mm in the steps of 0.01mm and Feed-rate was varied from 0.200 mm/flute to 0.290 mm/flute in steps 0.01 mm/flute. It should be noted that the feed-rate was defined as feed per tooth, thus the definition of feed-rate also accounts for the tool RPM, and thus RPM is not needed to be taken as a separate variable.

For various Combination of Feed-rate and ADOC, time domain simulations were conducted and instantaneous forces (Feed, Axial, Normal) values were stored in excel Files. A total of 910 files were generated for various combinations of cutting parameters.

Generated dataset was then shuffled and divided into two separate datasets, namely training and validation dataset, by randomly picking datapoints.

**CUTTING CONDITIONS CONSIDERED**

Slotting operation was chosen for the machining simulation using a 4-Flute Ball End Mill tool with constant helix angle.

|  |  |
| --- | --- |
| Table 1: Geometrical Parameters of the tool | |
| Parameter | **Value** |
| Diameter | 20 mm |
| Length | 150 mm |
| Flute Height | 10 mm |
| Helix Angle | 300 |
| Rake Angle | 50 |

|  |  |
| --- | --- |
| Table 2: Properties of the workpiece (Ti-6Al-4V alloy) required for the simulation | |
| Property | **Value** |
| Hardness | 340HB |
| Density | 4.706g/cm3 |
| Thermal Conductivity | 6.000W/m-K |
| Young’s Modulus | 1.15\*1011 N/m2 |
| Tensile Strength | 9\*108 N/m2 |
| Yield Strength | 8.30 \*108 N/m2 |
| Shear Strength | 7.60 \*108 N/m2 |

|  |  |
| --- | --- |
| Table 3: Other cutting parameters | |
| Parameter | **Value** |
| Spindle Direction | Clockwise |
| Milling Mode | Slotting |
| RPM | 300 |
| Number of revolutions | 1 |
| Sampling Frequency | 10 |

**DEFINING INPUT AND OUTPUT VARIABLES**

Input variables- (Axial Depth of Cut) ADOC, Feed-rate.

Output variables- Ftang, Fnormal, Faxial.

Ftang – Tangential Force.

Fnormal – Normal Force.

Faxial – Axial Force.

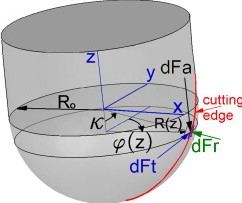
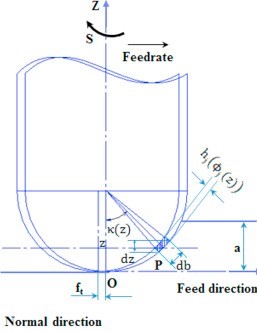


Figure 1: Pictorial representation of input and output variables

|  |  |
| --- | --- |
| **Table 4: Abbreviations used in the paper** | |
| **A** | Axial depth of cut (ADOC) [mm] |
| **ft** | Feed/Tooth (feed-rate) |
| **dft** | Differential Tangential force (F­tang) |
| **dFr** | Differential Normal force (Fnormal) |
| **dFa** | Differential Axial force (Faxial) |

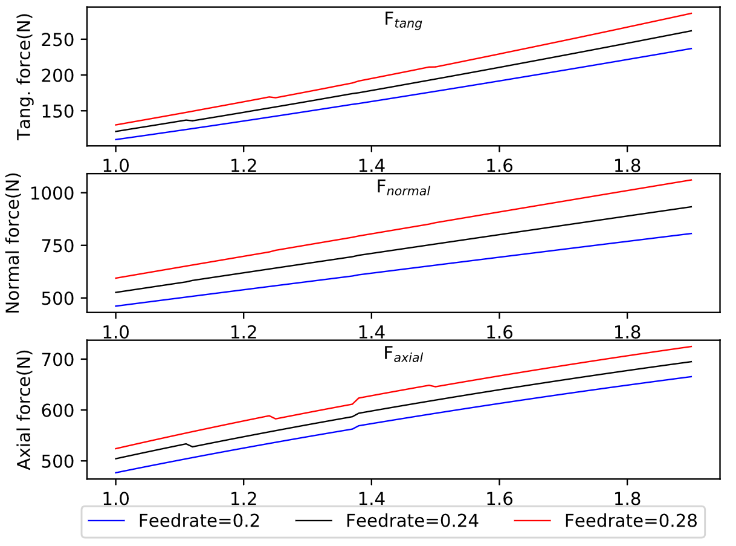
In the following figures, actual variation in cutting forces with respect to the cutting parameters (axial depth of cut and feed-rate) is shown.

Figure 2:Cutting forces vs ADOC for 3 different values of Feed-rate

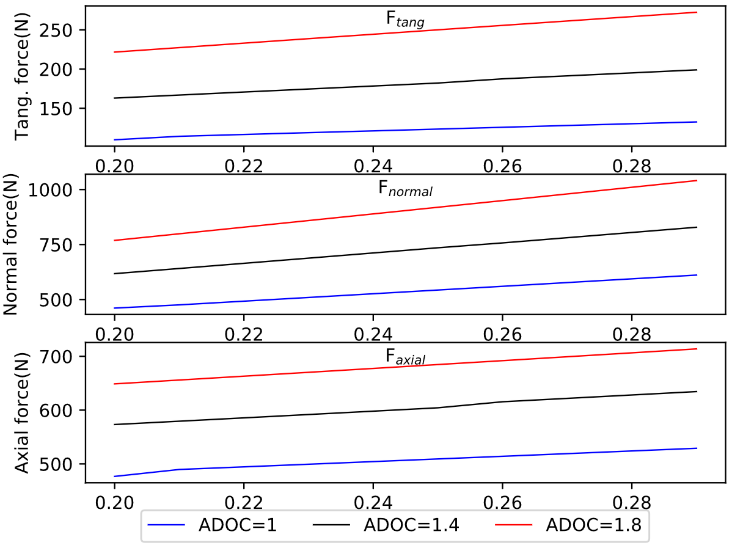


Figure 3: Cutting forces vs Feed-rate for 3 different values of ADOC

In figure 3, curve with higher force values belong to the higher feed-rate, thus increase in feed-rate shifts force graph upwards.

From figures 2 and 3 it can be observed that maximum variation is in the normal/radial direction, while in other two directions amount of variation is similar. The magnitude of axial force was found to be much more than that of feed /tangential force.

**NEURAL NETWORK ARCHITECTURE**

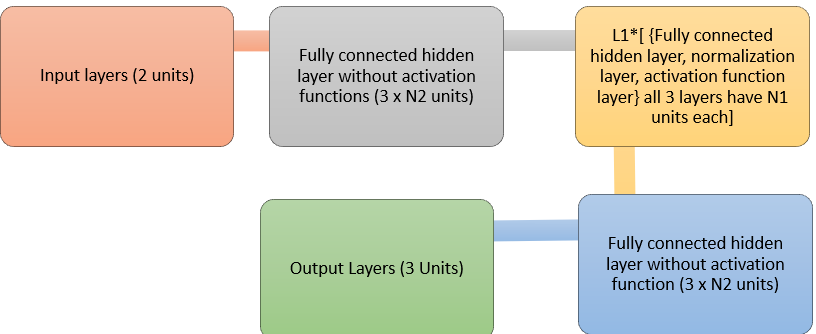
****1. **Neural network to be used for learning**.

Figure 4: Neural network structure, showing details of each layer

2. **Layer Normalisation** [11].

Output of normalisation layer is given by :

|  |  |  |
| --- | --- | --- |
|  |  | () |

,

where, is the value at ith hidden unit of lth layer and H is the number of hidden units in a layer. All the hidden units in a layer share the same normalization terms and , but different training datapoints have different normalization terms. γ and β are learnable parameters.

3. **Activation function** [12] – PReLU

|  |  |  |
| --- | --- | --- |
|  | PReLU() = *max*(0,)+a0∗*min*(0,) | () |

where, a0 is a learnable parameter.

4. **Loss function** - Mean squared percent error.

|  |  |  |
| --- | --- | --- |
|  |  | () |

Where, N is the number training examples, actij is actual value of ith output of jth training example and predij is predicted value of ith output of jth training example.

5. **Adam optimizer** [13] used for minimising the loss, betas = {0.9, 0.999}, along with mini-batch gradient descent.

6. Choice of activation function, loss function and optimizer was made after training and evaluating some simple neural networks on small dataset. It was observed that PReLU, mean squared percent error and Adam gave best results. Layer normalisation did not significantly improve the accuracy; however, it still gave positive results.

**TRAINING THE NEURAL NETWORK**

1. Various models were trained on each of the above cases by tuning following hyperparameters:

I. **L1** = Number of hidden layers with layer normalisation and activation.

II. **N1** = Number of units in “**L1**” hidden layers.

III. **N2** = Parameter defining number of units in remaining hidden layers.

IV. **Initial LR** = Initial learning rate.

V. **Update period =** Number of iterations after which the learning rate will be updated.

VI. **Update factor =** Factor by which to divide on each update.

2. For tuning the hyperparameters mentioned above, several models were trained by randomly sampling the values of each hyperparameter from a certain interval best suited for that hyperparameter. Sampling interval of all hyperparameters was narrowed down over subsequent iterations, around the region giving best performance for finer tuning.

|  |  |
| --- | --- |
| Table 5: ML training parameters | |
| Hyperparameters | **Values** |
| # Training datapoints | 890 |
| # Iterations | 10,000 |
| Initial LR | 0.01 |
| Update period | 1,000 |
| Update factor | 2 |

3. Dataset generated earlier was shuffled and then datapoints were randomly selected from it, to generate training and validation datasets. Models were trained over training dataset and then, validation dataset was used to pick the best model. Finally, the model that performed best on the validation dataset was evaluated on the test dataset and results are tabulated in the next sections. After training few simple models, following values of learning rate parameters, mini-batch size number of mini-batches, number of iterations, etc. worked best:

4.Models were trained using Pytorch [14] on NVIDIA GeForce GTX 1060 6GB GPU.

**RESULTS OBTAINED AFTER TRAINING**

1. After training several models, neural network similar to the following architecture worked best for amount of data that was generated for training. For the given values of hyperparameters deviation of the worst datapoint among the validation dataset was brought down to less than 5 percent:

**L1** = 4, **N1** = 270, **N2** = 3

|  |  |  |
| --- | --- | --- |
| Table 6: Percent Deviation of ML model predictions | | |
| Output | **Mean Percent deviation** | **Max. Percent deviation** |
| Ffeed | 0.13 | 0.49 |
| Fnormal | 0.13 | 0.55 |
| Faxial | 0.25 | 0.94 |

2. Apart from the training and validation dataset a separate dataset was also generated to report the results. In the following table, maximum and mean percent deviations of predicted values from actual values is written:

3. Graphs shown below depicts the training and validation errors over the course of training.

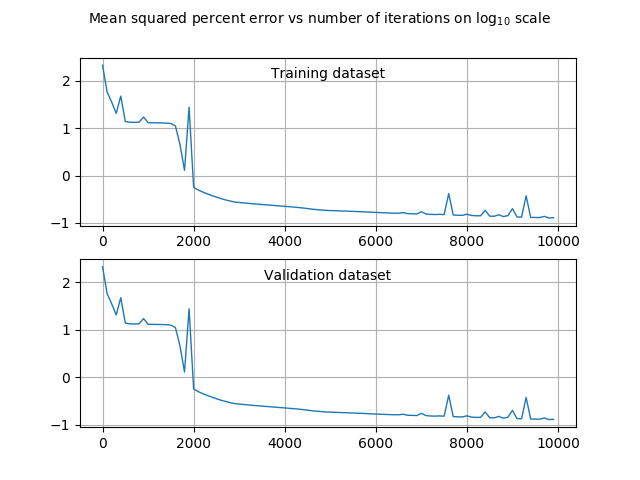


Figure 5: Mean squared error vs iterations.

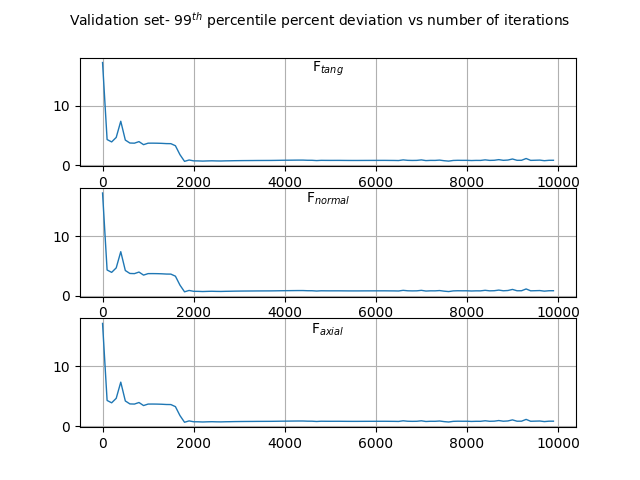


Figure 6: 99th percent error vs iterations.

**GRAPHICAL COMPARISON BETWEEN ACTUAL AND PREDICTED VALUES**

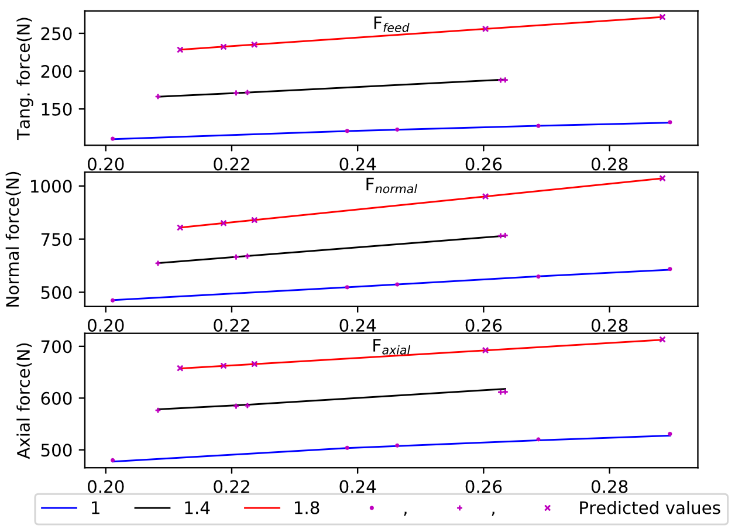
1. Following figures show graphical comparison between predicted values plotted against actual curves of cutting forces vs Axial depth of cut and feed-rate. Figure 7 shows graphical comparison of predicted and actual values for three different axial depth of cuts, while figure 8 is showing for three different feed-rates.

Figure 7: Cutting forces vs feed-rate for different values of ADOC.

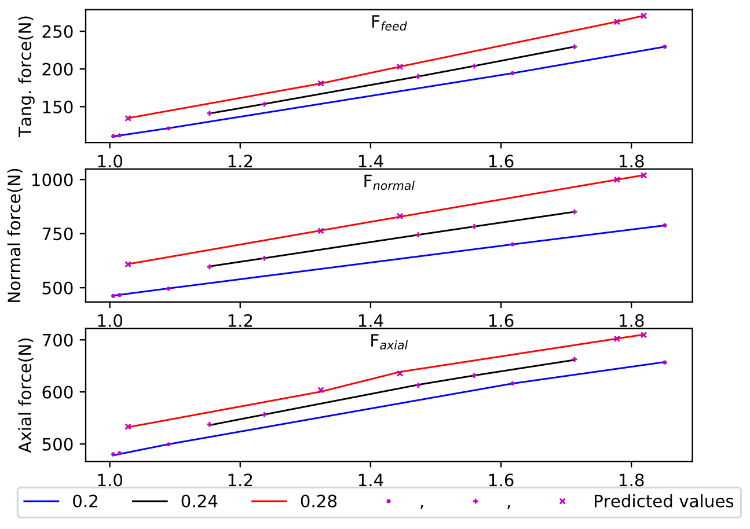


Figure 8: Cutting forces vs ADOC for different values of feed-rate.

2. Figures 9 and 10 shows percent deviations corresponding to the plots in figures 7 and 8 respectively.

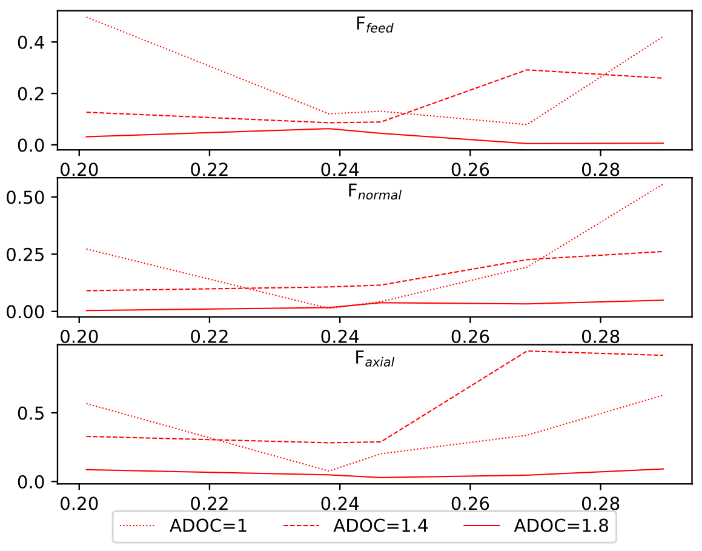


Figure 9: % deviations vs feed-rate for different values of ADOC

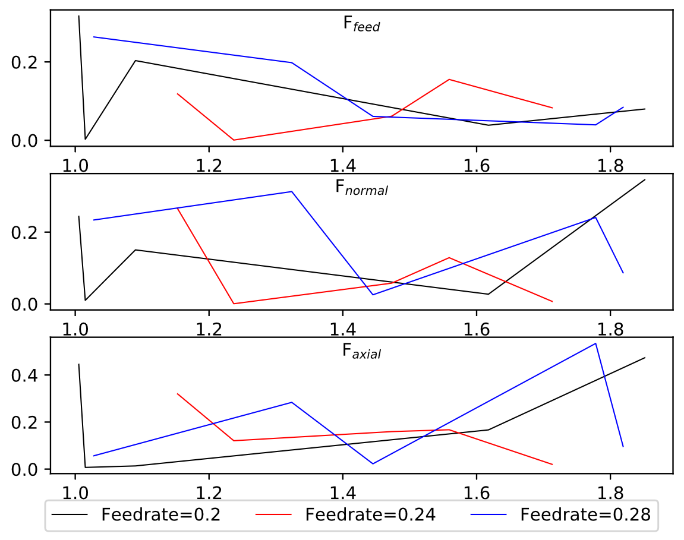


Figure 10: % deviations vs ADOC for different values of feed-rate

**OBSERVATIONS**

From the results obtained in the previous sections, we can easily observe that with just 890 data points used for training the neural network, maximum deviation of the predicted value from actual value over the entire test (unseen) dataset is less than 1 percent. This shows that the accuracy of the predicted values is more than enough for any practical purposes. Also, since maximum percent deviation is less than 1 percent, we can state with very high confidence that for any data point within the range of input values used for training, cutting forces can be predicted very accurately without deviating from actual values by more than 1 percent.

**CONCLUSION**

Although from actual curves it may seem that variation of cutting forces with respect to cutting parameters is linear and thus, linear regressing would have also given similar accuracy. Therefore, a linear surface was fitted over the data by switching off the activation function and layer normalisation in the neural network and it was observed that linear regression was giving more than 5 percent deviation in some of the predicted values. So, we can say that even though 2D graphs are looking like straight lines, there is some curvature in 3D plot which was not captured by linear regression. Also, neural networks are far more robust to wrong datapoints in training data than linear regression.

**FURTHER APPLICATIONS**

Since even very simple neural networks are able to predict cutting force values for titanium machined using ball end mill with such high accuracy, therefore, we can easily extend this work to other tools and workpiece metals. Although there are plenty of software to simulate milling or any other machining operation, but they not free and take a lot of space on hard disk. Also, performing one simulation over these software take a lot of time compared to neural networks, thus a machine learning model built using python can replace these costly simulators and time-consuming software.

**REFERENCES**

[1] Sanchez, P.N. (2010). Titanium alloys: Preparation, properties and applications.

[2] Wollmann, M. & Kiese, J. & Wagner, Lothar. (2012). Properties and applications of titanium alloys in transport. Ti-2011. 2. 837-844.

[3] Williams, David. (2001). Titanium for Medical Applications. 10.1007/978-3-642-56486-4\_2.

[4] Pratap, Tej & Patra, Karali & Dyakonov, Aleksandr. (2015). Modeling Cutting Force in Micro-Milling of Ti-6Al-4V Titanium Alloy. Procedia Engineering. 129. 134-139. 10.1016/j.proeng.2015.12.021.

[5] Vijayan, Krishnaraj & Samsudeensadham, S & Sindhumathi, R & Kuppan, P. (2014). A study on High Speed End Milling of Titanium Alloy. Procedia Engineering. 97. 10.1016/j.proeng.2014.12.248.

[6] Wojciechowski, Szymon. (2014). The estimation of cutting forces and specific force coefficients during finishing ball end milling of inclined surfaces. International Journal of Machine Tools and Manufacture. 89. 10.1016/j.ijmachtools.2014.10.006.

[7] Wojciechowski, Szymon & Mrozek, Krzysztof. (2017). Mechanical and technological aspects of micro ball end milling with various tool inclinations. International Journal of Mechanical Sciences. 134. 10.1016/j.ijmecsci.2017.10.032.

[8] Budak, Erdem & Tunc, Taner & Budak, Erhan. (2011). Analytical methods for increased productivity in 5-axis ball-end milling. Int. J. of Mechatronics and Manufacturing Systems. 4. 238 - 265. 10.1504/IJMMS.2011.041471.

[9] Kao, Yung-Chou & Nguyen, Nhu-Tung & Chen, Mau-Sheng & Huang, Shyh-Chour. (2015). A combination method of the theory and experiment in determination of cutting force coefficients in ball-end mill processes. Volume 2. Pages 233-247. 10.1016/j.jcde.2015.06.005.

[10] Mhamdi, M-B & Boujelbene, Mohamed & Bayraktar, Emin & Zghal, Ali. (2012). Surface Integrity of Titanium Alloy Ti-6Al-4V in Ball end Milling. Physics Procedia. 25. 355-362. 10.1016/j.phpro.2012.03.096.

[11] Lei Ba, Jimmy & Ryan Kiros, Jamie & E. Hinton, Geoffrey. (2016). Layer Normalization.

[12] He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. IEEE International Conference on Computer Vision (ICCV 2015). 1502. 10.1109/ICCV.2015.123.

[13] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[14] Pytorch, <https://github.com/pytorch/pytorch>