Tool Wear Prediction Using Artificial Neural Networks and Regression Models

Necip Fırat Helvacıoğlu
Robotic Engineering
Izmir Katip Celebi University
Izmir, Turkey
firat_helvaci@hotmail.com

Abstract— **In this project, we aimed to predict the tool wear when using CNC milling tool under various cutting conditions. As a result of the experiment, the system is optimized. In article which we selected interesting on the tool life modeling of the materials to be processed using the artificial intelligence tools is discussed in detail. In this article we use neural network and decision tree models to describe the engineering phenomena. After that we give brief observation from models.**

In previous reference studies, optimization of the system was not perfect performed. In this report, we have made modeling and analysis for each output data of the system using the optimization technique for the system, taking the system's data as a reference.

We have calculated and eliminated the outliners in this data. Using the models which are created and we have found the validation accuracy and accuracy values of the system. As a result, by using optimal model we were able to predict the tool wear before it is happened.

Keywords— Tool wear, CNC Milling, Regression Model, Neural networks

I. INTRODUCTION

In this study, speed, feed rate, lamp pressure, tool position and electrical conditions were selected. A Neural Network models were developed and compared using these milling parameters. In this study, the effects of system parameters on tool wear and parameters of optimum machining have been determined.

In this article we also used regression models to describe the engineering phenomena. After that we use stochastıc optimization method with neuron regression analysis for modelling tool life in face milling base on Python programing language.

In a metal cutting process, the most significant aspect of tool life assessment is the measure of tool wear. The factors that affect tool wear are divided in to three categories the machine, the machining parameters, and the work piece material. Tool wear prediction is an important factor that has profound influence on productivity in industrial activities. High metal removal rate is intended to reduce the manufacturing cost and operation time [5]. The productivity in terms of machining cost, machining operation and quality of the work piece and its integrity strongly depend on tool wear and consequently it depends on the life of the tool. The maximum utilization of cutting tool is one of the ways for an industry to reduce its manufacturing cost [6]. In order to predict tool wear from a machining process an accurate process model must be constructed for a CNC milling process with speed, feed rate, lamp pressure, tool position and electrical conditions as input variables and tool wear as the output variable.

In the literature, tool wear and tool life model have been extensively studied. Oraby and Hayhurst (7] developed models for wear and tool life using nonlinear regression analysis techniques in terms of the variation of a ratio of force components acting at the tool tip. Richetti et al [8] investigated the effect of the number of tools used in face milling operations and related it to the establishment of a tool life under specified cutting conditions. Choudhury et al [9] predicted the response variables like flank wear, surface finish and cutting zone temperature in turning operations using design of experiments and the neural network technique and the values obtained from both methods were compared with the experimental values to determine the accuracy of the prediction. Yongjin and Fischer [10] developed tool wear index (TWI) and the tool life model for analyzing wear surface areas and material loss from the tool using micro-optics and image processing/analysis algorithms. Huang et al [11] developed a multiple regression model in detecting the tool breakage based on the resultant cutting forces in end milling operations. Srinivas and Kotaiah [12] developed a neural network model to predict tool wear and cutting force in turning operations for cutting parameters: viz., cutting speed, feed and depth of cut. Chattopadhyay [13] used the feed forward back propagation artificial neural network for evaluation of wear in turning operations using carbide inserts taking speed, feed and depth of cut as input parameters. Thomas et al [14] investigated the effect of cutting parameters on tool stiffness and damping, and obtained an empirical model for predicting the behaviour of the tool stiffness variation. Arsecularatne [15] developed a semi empirical method for predicting tool life in machining with restricted contact (RC) tools. Lin et al [16] investigated the effect of a cutting tool's geometrical shapes on tool wear, roughness of the machined surface and cutting noise produced and constructed a tool life prediction system. Jaharah A G Hani et al [17] described the tool life model when end milling tool steel using coated carbide tipped tool. Yao et a,. [18] present a new method of tool wear detection with cutting conditions and detected signals which includes the model of wavelet fuzzy neural network with acoustic emission and the model of fuzzy classification with motor current. Puri et al. [19] present the use of fuzzy logic in the Taguchi method to optimize Electro Dischargre Machining (EDM) process with multiple quality characteristics. QunRen et al.[20] presents a tool wear monitoring method using Takagi-Sugeno-Kang fuzzy approach. Palanikumar et al. [21] employ the Taguchi method with fuzzy logic to optimize the multiple performance characteristics of a machining process.

Tool wear can be defined as gradual failure of cutting tools in normal operation. Tool were should be predict before its happening. It will be uneconomical because cost of tool replacement will be high and also damaged tool can damage catting material badly. To predict tool life supervised binary classification is performed for identification of worn and unworn cutting tools. Eight experiments were run with an unworn tool while ten were run with a worn tool.

II. Method

a. Exprimenting

To predict the tool were, series of experiments were done. A series of machining experiments were run on 2" x 2" x 1.5" wax blocks in a CNC milling machine in the System-level Manufacturing and Automation Research Testbed (SMART) at the University of Michigan. Machining data was collected from a CNC machine for variations of tool condition, feed rate, and clamping pressure. Each experiment produced a finished wax part with an "S" shape - S for smart manufacturing - carved into the top face, as shown in Figure 1.

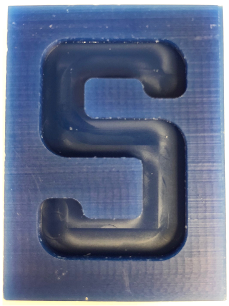

Figure 1: S shape carved into the top face of wax

Total of 18 experiment were done. Every iteration done with different conditions. Feed rate and clamp pressure has changed for every experiment. Time series data was collected from 18 experiments with a sampling rate of 100 ms and are separately reported in files experiment_01.csv to experiment_18.csv. Each file has measurements from the 4 motors in the CNC (X, Y, Z axes and spindle). From first experiment path of the tool can be seen in Table 1.

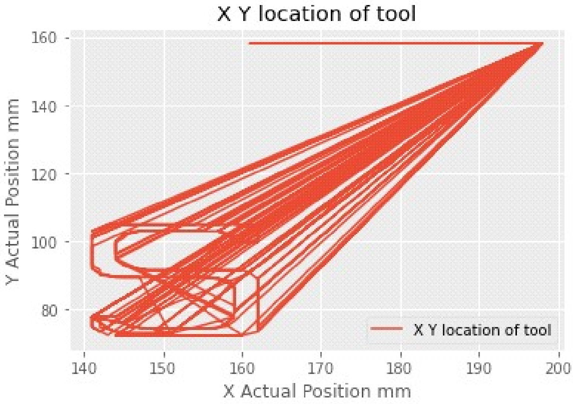

Figure 2: Actual position of tool in experiment 1

b. Detecting Outliners

In these experiments data were captured using several sensor. 100 ms capture rate used for these experiments. And data sets are collected for each experiments. And X Y Z and spindle location data were collected. Also motor and machine condition data were also collected. During this data collecting process there can be some interferences in those data. These interferences can be caused by problems with sensor or some other environmental problems. The reason of these interferences has not been identified. So in order to process of the data that collected we need detect the outliners (unwanted data) and replace with a new value.

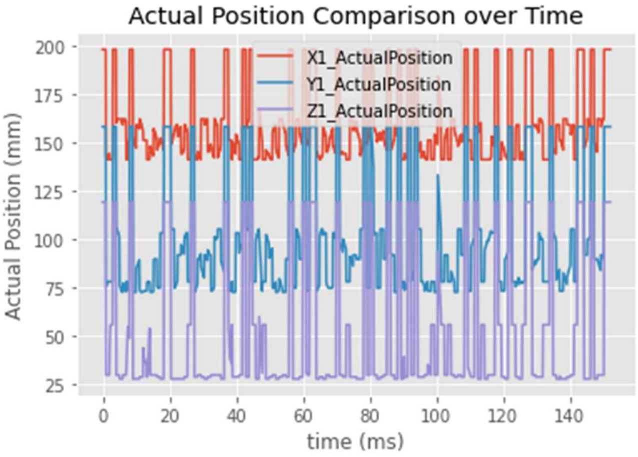

Figure 3: Actual position of tool in experiment 3

To detect the outliners we used IQR methodology. This method is useful for detecting in a series of sensor measurements. Sensor sometimes can gives us high or low range values when in error mod and IQR methodology can identify unwanted values.

To use IQR mythology first we need to detect fist and third quartiles of data which represents a quarter and three-quarters of the way through the list of all data. And by subtracting the quartiles we can find the range of the interquartile range of the data. The interquartile range shows how the data is spread about the median

$$IQR = Q_3 - Q_1 \quad (1)$$

And using IQR value we can determine the high and low ranges of our data.

$$High\ Ouliner\ range = Q_3 + 1.5 * IQR \ (2)$$

$$Low\ Ouliner\ range = Q_1 - 1.5 * IQR \ (3)$$

With this we can eliminate the outliners. And giving a median value to an outliner we can smooth out the data that we are working on.

c. *Implamanting Nerual Networks*

Neural network is a circuit that can simulate behaviors of neurons in the human brain. This simulations are done in computer environment. This neural network is useful for solving decision making problems. Each connections like synapsis can transmit signals to the other neurons. The neurons process these signals and sends these values to other networks and these process is done for every neurons.

Best way to simulate these neural networks in artificial environment, we create a mathematical models of the neurons. Every neurons we simulate called as layer. And these layers has mathematical function attached to them. And connections to those layers has some weigh to them. Connections that connected to the layers summed so layers can process the values.

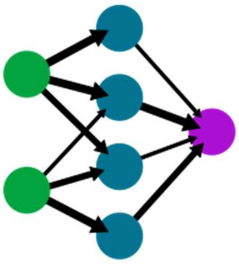

Figure 4: Simple neural network

To create the model, best number of neurons in the layers must be chosen. The function of the attached to the layers also has a great importance. For the data, functions are chosen as tanh, relu and sigmoid functions.

With these elements the construction of neural network can be done and can be trainable. But big models suffers from over training. Over trained models are learned only the data that it is given. It cannot predict correctly data that not used for training.

To overcome over training, we used 2 methods. Firstly, we limited the number of connections that done in the connections using dropout method. Also we used Batch normalization method. This method, normalizes the output of a previous layer by subtracting the batch mean and dividing by the batch standard deviation. With the given methods best model was created.

Layer (type)	Shape
(Dense)	512
(Dropout)	512
(Batch)	512
(Dropout)	512
(Dense)	256
(Batch)	256
(Dropout)	256
(Dense)	64
(Batch)	64
(Dropout)	64
(Dense)	32
(Batch)	32
(Dense)	1

Table 1: Neural Network Model Parameters

d. Implementing Regression Models

In decision tree regression, trees used for regression and trees used for classification have some similarities - but also some differences, such as the procedure used to determine where to split. Some techniques, often called *ensemble* methods, construct more than one decision tree:

- **Boosted trees** incrementally building an ensemble by training each new instance to emphasize the training instances previously mis-modeled. A typical example is AdaBoost. These can be used for regression-type and classification-type problems.
- **Bootstrap aggregated** (or bagged) decision trees, an early ensemble method, builds multiple decision trees by repeatedly resampling training data with replacement, and voting the trees for a consensus prediction.
  - A **random forest** classifier is a specific type of bootstrap aggregating
- **Rotation forest** – in which every decision tree is trained by first applying principal component analysis (PCA) on a random subset of the input features.

A special case of a decision tree is a decision list which is a one-sided decision tree, so that every internal node has exactly 1 leaf node and exactly 1 internal node as a child (except for the bottommost node, whose only child is a single leaf node). While less expressive, decision lists are arguably easier to understand than general decision trees due to their added sparsity, permit non-greedy learning methods and monotonic constraints to be imposed.

Decision tree learning is the construction of a decision tree from class-labeled training tuples. A decision tree is a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label. The topmost node in a tree is the root node.

There are many specific decision-tree algorithms. Notable ones include:

- ID3 (Iterative Dichotomiser 3)
- C4.5 (successor of ID3)
- CART (Classification And Regression Tree)
- CHAID (CHi-squared Automatic Interaction Detector). Performs multi-level splits when computing classification trees.[11]
- MARS: extends decision trees to handle numerical data better.
- Conditional Inference Trees. Statistics-based approach that uses non-parametric tests as splitting criteria, corrected for multiple testing to avoid overfitting. This approach results in unbiased predictor selection and does not require pruning.

ID3 and CART were invented independently at around the same time (between 1970 and 1980) yet follow a similar approach for learning a decision tree from training tuples.

Algorithms for constructing decision trees usually work top-down, by choosing a variable at each step that best splits the set of items.

Used by the ID3, C4.5 and C5.0 tree-generation algorithms. Information gain is based on the concept of entropy and information content from information theory.

Entropy is defined as below

$$\mathrm{H}(T) = \mathrm{I}_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

Where p_1,p_2….. are fractions that add up to 1 and represent the percentage of each class present in the child node that results from a split in the tree.

$$\overbrace{IG(T,a)}^{\text{Information Gair}} = \overbrace{\mathrm{H}(T)}^{\text{Entropy}} - \overbrace{\mathrm{H}(T|a)}^{\text{Weighted Sum of Entropy}}$$

$$= -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_a p(a) \sum_{i=1}^{J} -\Pr(i|a) \log_2 \Pr(i|a)$$

Information gain is used to decide which feature to split on at each step in building the tree. Simplicity is best, so we want to keep our tree small. To build the tree, the information gain of each possible first split would need to be calculated.

Introduced in CART variance reduction is often employed in cases where the target variable is continuous (regression tree), meaning that use of many other metrics would first require discretization before being applied. The variance reduction of a node *N* is defined as the total reduction of the variance of the target variable *x* due to the split at this node:

$$I_V(N) = \frac{1}{|S|^2}\sum_{i\in S}\sum_{j\in S}\frac{1}{2}(x_i - x_j)^2 - \left(\frac{1}{|S_t|^2}\sum_{i\in S_t}\sum_{j\in S_t}\frac{1}{2}(x_i - x_j)^2 + \frac{1}{|S_f|^2}\sum_{i\in S_f}\sum_{j\in S_f}\frac{1}{2}(x_i - x_j)^2\right)$$

Where S,S_t , and S_f are the set of presplit sample indices, set of sample indices for which the split test is true, and set of sample indices for which the split test is false, respectively.

Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds. Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. The Random Forest algorithm randomly selects observations and features to build several decision trees and then averages the results. Another difference is that decision trees might suffer from overfitting. Random Forest prevents overfitting most of the time, by creating random subsets of the features and building smaller trees using these subsets. Afterwards, it combines the subtrees. Note that this doesn't work every time and that it also makes the computation slower, depending on how many trees your random forest builds.

III. Results and Discussion

Since the data that working on is token from sensors, we eliminated the outliners from our data sets. We used IQR methodology. There are 19 experiments and we try to use this method to every experiments and compare the results.

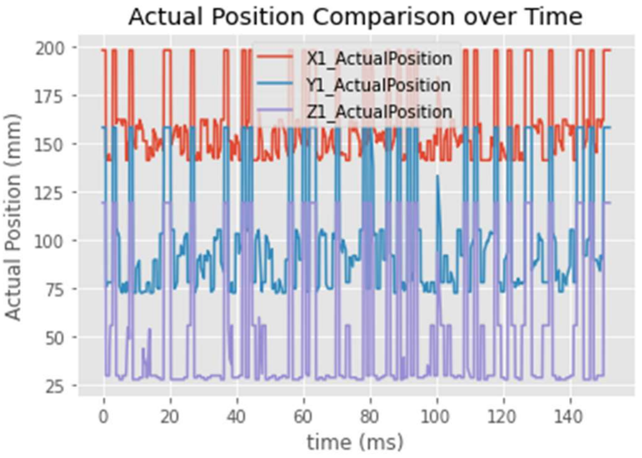

Figure 5: Actual position of tool in experiment 3 with Outliner

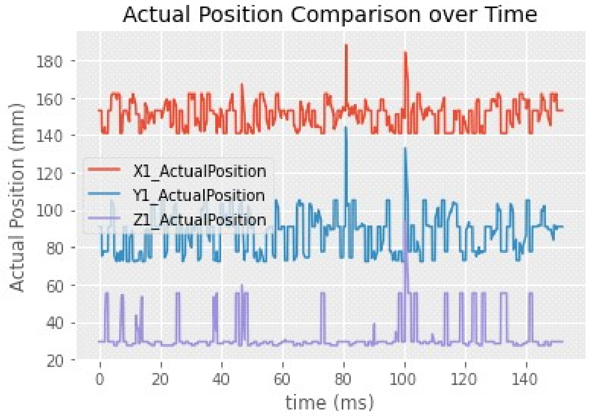

Figure 6: Actual position of tool in experiment 3 without Outliner

As we can clearly see in experiment 3 that sensor data is smooth out by IQR methodology. We found outliners and give maiden of the data. The resulting new data still has some noisy data. But unfortunately eliminating all of them needs a better algorithm. Best way of eliminating is fixing measuring errors in hardware site. The noisy signals caused by environment can affect the sensors.

The neural network we built trained 3000 iteration with 10000 patches. Adam optimizer used. It took 10 minutes to fully train. We used binary accuracy metric to compare the results. We recorded binary accuracy values and loss values for every iteration and plotted them.

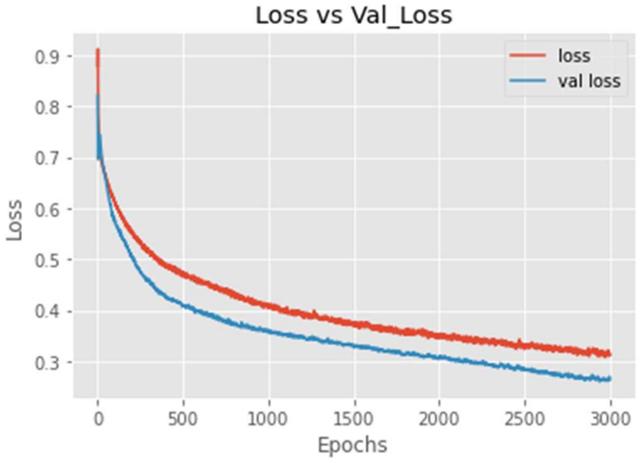

Figure 7: Loss of validation and train data for every iteration

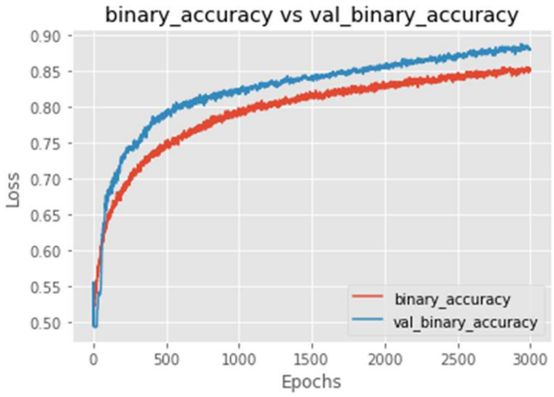

Figure 8: Binary accuracy of validation and train data for every iteration

The results of the neural network training has a test accuracy of 0.8527 and validation accuracy is 0.8814. It also has Loss 0.3118 in test data and 0.2628 in validation. The test results are showing us that our model is trained without any over fitting. The validation data have higher accuracy than the train data.

Regression and decision methods are applied. Our main algorithms are random forest and desiccation tree. We draw Roc curve of the results. Roc is a performance measurement method. It shows the true and false rate of the data. And high area under the curve means better performance.

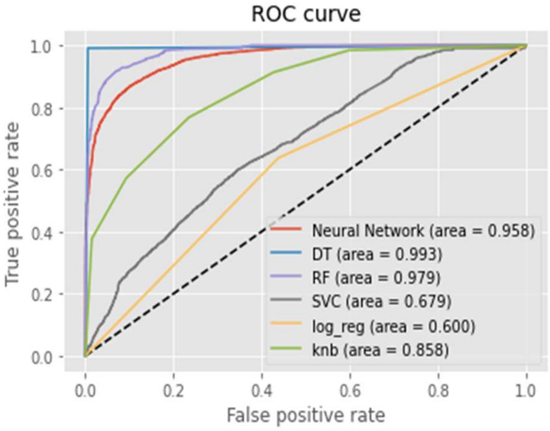

Figure 9: Loss of validation and train data for every iteration

As a result, we can see in Figure 9 neural network, decision tree and random forest has high performance. Decision tree can predict the system better than others. Logistic regression method has least predication performance. This behavior is expected because our data is highly non-linear data and logistic regression algorithm is good for predicting more linear data.

	Accuracy	F1 score	Area
Neural Network	0.8749	0.8726	0.958
Decision Tree	0.9926	0.9929	0.993
Random Forest	0.9010	0.9121	0.979
Support Vector Regression	0.6016	0.5840	0.679
KNeighbors	0.7665	0.7745	0.858
Logistic Regression	0.6012	0.6253	0.600

Table 2: Results of different algorithms

IV. Conclusion

As result shows us that we can predict the tool wearing before it is happening based on sensor data. The results are very good and meaning full. We used various algorithms to predict the tool wearing. This paper aimed to increase the product quality of the materials that are processed in surface milling and this goal has been achieved. With the predicting the tool wearing the product quality increases and after that the workload, the working hours of the machine decreases and the cost decreases.

References

1. Elmagrabi N H, Shuacib F M and Haron CHC, 'An overview on the cutting tool factors in achinability assessment', Journal of Achievements in Materials and Manufacturing Engineering, 2007, Vol,.3, o.2,.87-90.

2. EJ.A. Armarego and R.I, Brown, The machining of Metals, Prentice=IIall, Englewood Cliffs, 1969.,

3. FJaklistsch, Metal cutting Technology, Valeron, Berkly, 1983.

4. MC. Shaw, Metal Cutting Principes, Oxford, Newyark, 1991,

5. M.A. Lajis, A.N. Mustafizul Karim, A.K.M..Nurul Amin, A.M.K Hafiz, L.G. Tumad,' Prediction of tool life in end milling of hardened stecl AISI D2' Ewropean Journal of Scientific Research, 2008, 4,p. 592602.

6. P. Palonisamy, I. Rajendran and S$. Shanmugasundaram, * Prdiction of tool wear using regression nd ANN models in end milling operation', fnternational Journal of Manufacturing Technology, 008,37, p.29=41.

7. S.E Oraby and R . Hayhurst, 'Tool life determination based on the measurement of wear and toolforce ratio variation', international Journal of Machine Tools &Manufacture, 2004, 44, p.1261-1269,

8. A.Richett, A R Machado, Da Silva, E O Ezugwu, } Bonney," Influence of the number of inserts fortool life evaluation in face milling of steels ', International Journal of Machine Tools Manufacture, 2004, 44, p. 695=700.

9. §.K .Choudhury, G. Bastarya, ' Role of temperature and surface finish in predicting tool wearusing neural net work and design of experiment', International Journal of Machine Tools &Manufacture, 2003,43, p.747=753.

10. YougjinKwon, Gary W. Fischer, "A novel approsch to quantifying tool wear and tool lifemeasurements for optimal tool management', international Journal of Machine Tools &Manufacture+ 2003,43, p. 3599368.

11, P. T. Huang, J.C Chen, C.Y Chou, ' A statistical approach in detecting tool breakage in end millingoperations', Journal of indusirial Technology, 1999,15, p 1=7.

12. J.Srinivas Rama Kotaiah, 'Tool wear monitoring with indirect methods', Manufacturing echnology Today india, 2005,4 , p.7-9.

13. AB .Chattopsdhyay, S Roy, * Evaluation of wear of turning carbide inserts using neural net works',International Journal of Machine Tools &Manufacture, 1996,36, p.789=797.

14, M.Thomas , Y. Beauchamp, * Statistical investigation of modal parameters of cutting tools in dryturning', /nternational Journal of Machine Tools & Manufacture, 2003,43, p.1093=1106,

1S. J. A Arsecularatne, ' On prediction of tool life and tool deformation conditions in machining threstricted contact tools', fnternational Journal of Machine Tools &Manufacture, 2003, 43, p 6S7=669.

16. S. Y.Lin, J.C.Lin, C.C.Lin, W. Y Jywe, BJ. Lin, * Life prediction system using a tool's geometricshape or high — speed milling', /nternational Journal of manufacturing Technology, 2006.30.p.622-630 .

17. Jaharah A. Ghani , Fifdaus Mohamad Hamzah, Mobd. Nizam Ab. Rahman, Baba Md. Deros, ' hereliability of tool life prediction model in end milling', Journal Mekanial, 2006,21, p.65=71. .R.Nagpal, Tool Engincering and Design, Khanna Publishers, 2002.

18. Yingxue Yao, Xiaoli Li, Zhejun Yuan , "Tool wear detection with fuzzy classification and waveletfuzzy neural network", International Journal of Machine Tools &Manufacture, 1999,39, p, 1525-1538.

19. Puri Y.M and Deshpande N.V, "Simultaneous optimization of multiple quality characteristics ofWEDM based on fuzzy logic and Taguchi technique", Proceedings of the Fifth Asia PacificIndustrial Engineering and Management Systems Conference, 2004, p. 14-26.

20. Qun Ren, Marek Balazinski, Luc Baron, " Uncertainty precision for tool wear condition using type2 TSK fuzzy approach", IEEE ,/nternational conference on systems , man and cybernetics, 2009, p.660-665, 2009.

21. Palanikumar K, Karunamoorthy L, Karthikeyan R and Latha B, " Optimisation of machining parameters in turning GFRP composites using a carbide tool based on the Taguchi method withfuzzy logics", International Journal of Metals and Materials, 2006, Vol. 12, No.6, pp. 483-491.

22. ISO, Tool life testing with single point turnini tools,ISO 3685, 2nd edn. International organization or Standards, Geneva, 1993.

23-25. http://w3.balikesir.edu.tr/~bsentuna/wp-content/uploads/2013/03/Regresyon-Analizi.pdf

24. http://piyasarehberi.org/yatirim/portfoy-yonetimi/140-regresyon-nedir-nasil-hesaplanir

26. https://www.surveygizmo.com/resources/blog/regression-analysis/

27. https://blog.industryolog.com/regresyon-analizi/

28. Gass, S.I. (2000), Making Decisions with Precision, Business Week October 30, 2000.