Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 130 (2024) 1827–1831

www.elsevier.com/locate/procedia

57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024)

Transient Wear Modelling of Coated Cutting Tools

Jan Wolf [a,*], Tim Reeber [a], Nithin Kumar Bandaru[b], Martin Dienwiebel [b], Hans-Christian Möhring[a]

[a]*Institute for Machine Tools, University of Stuttgart, Holzgartenstraße 17, 70174 Stuttgart, Germany*
[b] *Institute for Applied Materials (IAM) Karlsruhe Institute of Technology, Am Forum 7, 76131, Karlsruhe, Germany*

* Corresponding author. Tel.: +49 711 685 83834; fax: +49 711 685700400. *E-mail address:* jan.wolf@ifw.uni-stuttgart.de

Abstract

Tool wear significantly contributes to machine downtime. Therefore, methods for describing tool wear are crucial for both end users and researchers aiming to develop wear-resistant coatings to enhance overall equipment effectiveness. Finite Element Method (FEA) cutting simulations have proven to be reliable predictors of wear, offering valuable insights into cutting tool behavior. These simulations focus on the thermo-mechanical steady-state of the cutting process and discretize the geometry update of cutting inserts into several steps, typically spanning several seconds. Consequently, manually updating nodes to model wear during the transient cutting process is impractical due to rapidly changing state variables. This limitation has led to less research on the transient phase of cutting regarding wear modeling. However, processes like profile grooving often entail significantly reduced cutting times compared to traditional lathe operations, amplifying the significance of the transient cutting phase. To facilitate a more comprehensive understanding of wear progression during the transient cutting phase, novel approaches are essential. This study introduces a methodology based on implementing Python algorithms to automate wear progress calculation in DEFORM-2D for each tool node in contact with the chip or workpiece. This approach executes node updates and continues the simulation in a loop, allowing users to freely define the geometry update interval, constrained only by the simulation's step size. As a result, wear progression during the transient cutting phase can be accurately modeled with greater precision through a fully automated Python script.

Keywords: wear simulation, transient cutting phase, node displacement

1. Introduction

In recent decades, finite element analysis (FEA) has emerged as an increasingly vital tool for understanding the machining process, owing to the progressive enhancement in computer performance and the resulting increase in simulation accuracy. Researchers and commercial software developers have progressively incorporated additional features into their work, such as coated tools or tool wear models. Since tool failure accounts for almost 20 % of machine down time [1], understanding and predicting failure mechanism and thus tool wear is necessary. Beyond the economic motivation to counter tool wear with more wear-resistant coatings [2], it is noteworthy that tool wear is also responsible for increased cutting forces, vibration and cutting temperatures which impact the surface integrity [3]. Two approaches are possible for modeling wear in FEA cutting simulation, both involving a geometry update of the tool due to wear. The first approach involves the local deletion of nodes when a specific wear condition is met. The second approach entails displacing each tool node that comes into contact with the workpiece or the chip. Consequently, the tool's mesh undergoes alteration during simulation, despite its typical representation as rigid [4]. The displacement of nodes, and thereby the distortion of the mesh, can potentially lead to degraded mesh quality and numerical challenges. This issue has been addressed through the implementation of temporal smoothing techniques for the state variables relevant for the applied wear model [5].

10.1016/j.procir.2024.10.323

Lagrangian and arbitrary-lagrangian-eulerian (ALE) formulations were employed for wear simulation. For the lagrangian formulation, the mesh follows the workpiece material which eventually leads to the formation of a chip whereas ALE formulation with lagrangian and eulerian boundaries requires a predefined chip [6] which requires preliminary experiments for the chip shape determination. An approach combining an initial lagrangian simulation to reach mechanical steady-state succeeded by multiple ALE simulations to reach the thermal steady-state has been employed to simulate tool wear with DEFORM 2D by utilizing a coupled abrasive-diffusive wear model to consider different wear mechanisms on the rake and flank face of the tool [7]. A similar approach is to perform an initial incremental lagrangian simulation and ALE simulation to reach the thermo-mechanical steady-state and then apply a lagrangian simulation iteratively with an update of the geometry in between the succeeding l agrangian simulations [8]. Other researchers have concentrated solely on the arbitrary-lagrangian-eulerian (ALE) formulation within Abaqus/Explicit. [9].

Despite the findings demonstrating the capability of finite element analysis (FEA) to simulate tool wear, as also affirmed in [10], the widespread adoption of this technique remains constrained by computational expenses. FEM simulations are computationally expensive and can last for days depending on the simulated cutting distances. Since cutting tools are designed to withstand the thermo-mechanical load of the cutting process, it is highly questionable if the simulation of the entire life span of a tool is feasible and even more if 3D models are considered [11]. In order to give some insights in the wear progression, two approaches are possible. Firstly simulating process conditions which won't be applied in reality e.g. by increasing the cutting speed to increase the thermo-mechanical load and thus increase the wear-rate. The second approach is based on discretizing the entire cutting process and calculating the tool wear at predefined timestamps. Concerning the time in between updating of the geometry, several values are present in literature. In [7] the time of each update of the tool geometry was set to 10 % of the estimated remaining tool life by applying a Taylor's law. In some simulations this was to values as high as 20 seconds. In [8] it was noted that one simulation loop corresponds to 30 seconds of machining. A subroutine for updating the worn tool was called after a cutting time of 45 seconds [12]. A constant wear rate of 15 seconds was implemented [5]. While determining the course of wear on coated cutting tools the wear rate was assumed to be constant for 5 seconds and the thermo-mechanical steady-state was assumed to be reached after a simulated cutting time of 1ms [9]. In contrast to the aforementioned approaches for wear simulation, a novel method aimed at reducing simulation time by circumventing the requirement for a simulation loop [13]. This was achieved by computing the time needed to attain a predefined wear state of the cutting insert. Here, the wear rate on the flank face was assumed to be constant for wear steps up to 100μm.

The underlying assumptions in the presented tool wear simulations is that a thermo-mechanical steady state has been reached which enables an efficient wear simulation due to the following mathematical exploitation of a wear-rate model, where the most widespread model known as the Usui wear rate model shown in equation 1 is used. Here, the wear rate depends on the state variables contact pressure σ_N, sliding velocity v and temperature T . The values of contact pressure and sliding velocity are underlying significant changes in the initial stages of the cutting process until the force stead-state is reached and a continuous chip formation is established. The surface of the cutting edge however is reported to reach the thermal steady state after few seconds. After reaching the thermo-mechanical steady-state the wear rate is assumed to solely dependent on the influence of the newly formed cutting tool geometry on the state variables of equation 1. Therefore, in the thermo-mechanical steady-state the wear for a tool geometry update can be calculated according to equation 2, where the value of Δt can be set freely. This results in the conclusion that an FEA wear simulation can be artificially enlarged by multiplying the wear rate obtained from steady state machining by a factor of Δt [14] which has been set according to the previously presented literature.

$$\frac{dW}{dt} = c_1 \sigma_N(t) v(t) e^{-c_2/T(t)} \tag{1}$$

$$W_{t_o+\Delta t} = W_{t_o} + \int_{t_o}^{t_o+\Delta t} c_1 \sigma_N(t_0) v(t_0) e^{-c_2/T(t_0)} dt \tag{2}$$

From this analysis, it can be concluded that the transient state of the cutting process where the chip is not fully formed and temperatures did not reach steady-state can hardly be modelled manually by the approaches above [15] since this would require thousands of iterations. Hence, there arises a pressing need for a fully automated tool geometry update system capable of efficiently managing such frequent updates, while remaining adaptable across various finite element method software platforms. Thus, research regarding cutting tool wear related to the initial cutting conditions is limited despite the fact that tool presented coatings for the purpose of a smooth initial cutting phase. In order to enhance the current state of the art of wear modelling the wear progress due to the transient phase of cutting needs to be considered which requires new automated methods. This enables tool manufacturers to develop coatings which are resistant to the process conditions encountered during the transient phase of cutting. Especially processes like profile grooving where the time per cut is short and the steady-state condition is not met could increase its productivity by providing new modelling methods for that transient phase. This is underlined by findings which suggest the thermal-steady state to be reached after several seconds of cutting time. Therefore, speeding up the manufacturing process can be achieved by improving the understanding of the transient wear process and the resulting improved design of tool coatings for wear resistance.

2. Applied methods for transient wear simulation

2.1. 2D FEM cutting simulation

Wear simulation is a challenging task and is like many other simulations highly dependent on input parameters of the user such as step increment, the chosen wear model, the mesh size and material models just to name a few. Based on the assumption that an interaction between the tool geometry and the workpiece determines the direction of the tool wear, the geometry of the workpiece has to be updated during the cutting simulation in order to achieve a realistic interaction. Due to the discretization of the tool geometry with nodes that make up elements of a mesh, the outmost nodes of the mesh determine the shape of the tool. This leads to the conclusion that these nodes need to be locally displaced by a value which is determined with a wear model. This work utilizes the software DEFORM 2D (version 13.1) to simulate the cutting process with nodal displacement. In DEFORM, a database containing all necessary information for the solver can be converted into a keyfile which is readable by humans and furthermore can be processed by a text editor. By adjusting certain values in this keyfile and converting it back to a database for the next simulation, the user has the option to alter the simulation in order to achieve a geometry update of the tool. By introducing a python program which automatically executes the cutting simulation, converts the database into a keyfile, displaces the nodes of the tool via an automated text editing, converts the textfile back to a database and starts the next simulation step, the geometry update can be described in a loop. This process is depicted in Figure 1. Therefore, the text-based mode of starting DEFORM simulations is exploited by the python program, which calls the preprocessor and starts the next simulation step by pasting the necessary commands in the command prompt of the used computer.

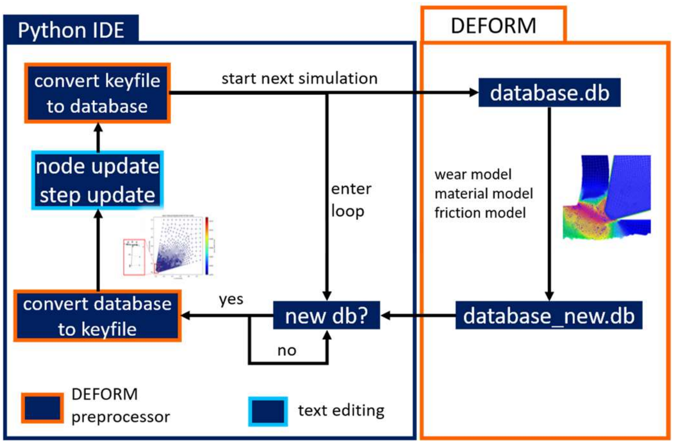

Fig. 1. Flow chart for automated tool geometry update with python.

Cutting speed v_c was set to 200 m/min and feed to 0.1 mm. Thermal conductivity for AISI 1045 was taken from [16]. The applied wear model in this study is the Usui wear model stated in equation 1. The values for c_1 and c_2 were taken from literature and determined for the pairing of TiAlN and AISI 1045 [17]. The Johnson cook material constitutive model described in equation 3 was implemented to account for the influence of strain hardening, viscous behavior and thermal softening [18]. The model parameters (A = 553.1 Mpa, B = 600.8 Mpa, C = 0.0134, n = 0.234, m = 1 and T_m = 1733 K) were taken from literature [19].

$$\sigma = (A + B\varepsilon_p^n)\left[1 + C\ln\left(\frac{\dot{\varepsilon_p}}{\dot{\varepsilon_0}}\right)\right]\left[1 - \left(\frac{T-T_0}{T_m-T_0}\right)^m\right] \qquad (3)$$

2.2. Algorithm for tool geometry update in DEFORM

Since the interaction of the workpiece and the tool is calculated in DEFORM on basis of a separately defined geometry and not the mesh itself [5], it is not sufficient to just alter the nodes of the mesh. Furthermore, the geometry which can be interpreted as the boundary between the outmost tool nodes and the environment needs to be displaced as well. However, DEFORM typically defines the tool geometry with very few contour points which makes matching of the geometry and the mesh impossible after a node displacement of the mesh took place. To satisfy this condition, an algorithm run prior to the database generation is implemented which matches the contour nodes of the mesh to the geometry and in addition fulfills the software’s demand by a counterclockwise definition of the geometry. The algorithm gets the coordinates of the tool mesh which can be exported during the initial database generation as an input, and prints the outmost nodes of the tool mesh in a counterclockwise manner to a specified keyfile which then serves as an input to the preprocessor. To determine whether a node of the mesh is a border node, a distinction between cases is implemented as follows. For each node a virtual coordinate system is defined which has its origin congruent to the node. Thereafter the remaining nodes of the mesh are sorted based on the quadrant they lie in. If each quadrant is assigned at least one node, it can be concluded, that the node is not part of the contour of the tool. This approach is depicted in Figure 2.

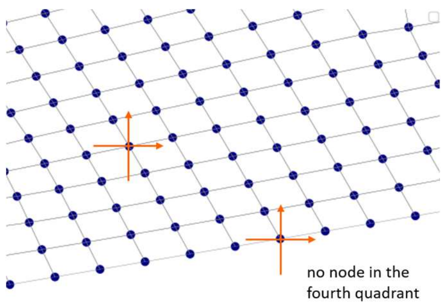

Fig. 2. Determination of geometry-nodes via quadrant technique.

This process is repeated for each node in the keyfile. With this algorithm the tool geometry is not defined by the default value of seven boundary nodes of the mesh but rather by each node which lies on the contour of the mesh. Thus, a congruent displacement of mesh and geometry can be guaranteed.

2.3. Algorithm for updating the tool node coordinates

For every node, the essential variables required to compute tool wear using the Usui wear model can be extracted from the converted text file. Apart from determining the wear depth for each node, it's crucial to define the direction of wear. In this investigation, the nodes are displaced orthogonal to the vector connecting their neighboring nodes. Moving nodes normal to the tools surface has been widely adopted by researchers and delivered acceptable results when simulating wear. Given that the contour nodes defining the tool geometry were pre-calculated, sorted counterclockwise, and saved to a text file, the task of identifying the neighboring nodes for each wear node can be accomplished using Python. This eliminates the necessity of interacting with the keyfile generated from the database. With respect to the nomenclature in Figure 3, the coordinates to define the vector $\overline{AB}$ can be determined. The next step of the algorithm is the calculation of an orthogonal uniform vector $\bar{n}$. This calculation could potentially lead to a vector pointing in the wrong direction and thus simulating the opposite effect of wear would take place. Hence, a case-by-case analysis is carried out for each node, as outlined in Equation 4.

$$\begin{pmatrix} B_x - A_x \\ B_y - A_y \\ 0 \end{pmatrix} x \begin{pmatrix} n_x \\ n_y \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix} \qquad (4)$$

Here the additional point c_1 or c_2 is defined by adding $\bar{n}$ to A regardless of the direction of $\bar{n}$. Then the cross product of (B-A)x(C-A) is calculated. If the third entry d of the resulting vector is negative, then $\bar{n}$ is not pointing towards the inside of the tool and needs to be multiplied by negative one to satisfy this condition. This approach is visualized in Figure 3. This way it can be ensured that the direction of moving nodes is as intended.

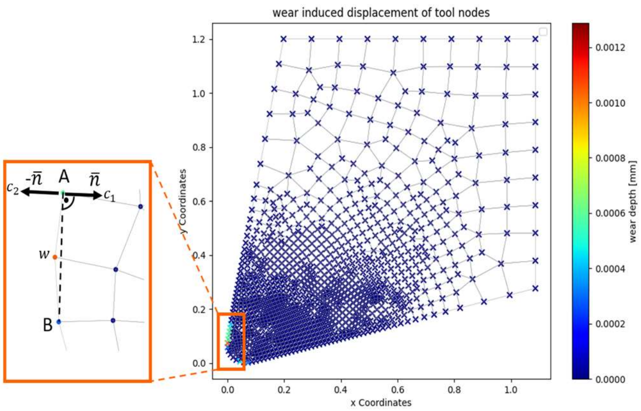

Fig. 3. Movement of nodes normal to the tools surface.

3. Results

The proposed python algorithm to perform a case by case study whether a node of the tools mesh is a border node was successfully implemented and the performance was checked with several meshes. As a next step the border nodes were successfully sorted with a subsequent algorithm to ensure a counter clockwise order of the geometry points. This came in handy to define the neighbouring nodes needed for the direction of the wear progress, because the nodes required to determine the direction of wear for node n are n-1 and n+1. Since geometry node one was chosen to be on the top right corner of the tool and never in contact with the chip, no case by case study is required for the nodes with the first or the lowest ID. Due to these sorting algorithms a fully automated definition of the geometry based on the mesh could be implemented which presents users an easy way to try different meshes for the cutting simulations without the need to manually sort the geometry defining nodes and match them with the mesh of the tool. This proves beneficial for mesh sensitivity analysis. The results of the geometry definition are depicted in Figure 4 where the discretization of the tools geometry (right) is noticeable compared to the default geometry (left).

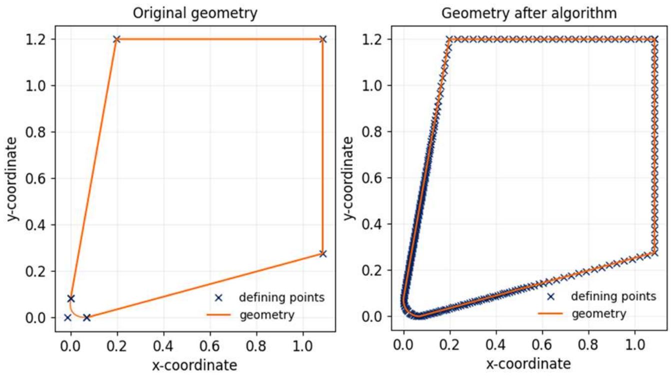

Fig. 4. Algorithm for matching the mesh to the geometry.

To provide insights to the wear behaviour before reaching the thermo-mechanical steady-state, three nodes were selected on the tools surface where the first node is allocated on the rake phase, the second at the tip of the tool with x-coordinate being zero (Figure 4 right) whereas the third node was selected on the flank face with y-coordinate equal to zero (Figure 4 right). With the applied models of chapter 2 the results of the wear rates in these models are depicted in Figure 5.

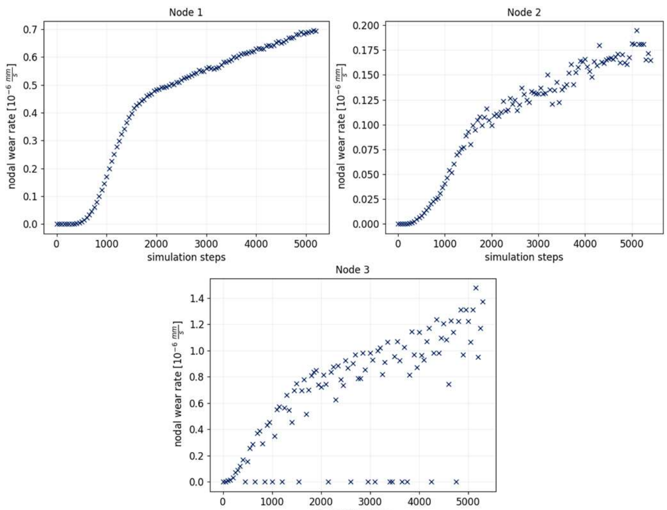

Fig. 5. Wear rates during the transient cutting phase at three predefined nodes.

Each data point in these graphs represents 50 steps of the FEA cutting simulation. In Figure 5 (bottom) it can be obtained that the wear rate is quite unstable at significant nodes throughout the simulation which is caused by the contact conditions. This issue was described as contact flickering in the context of FEA. In addition, the wear rates at both graphs in Figure 5 (top) underlie significant variation in a short timeframe. Thus, it can be concluded, that by simply choosing the last datapoint as the wear rate is not only overestimating the wear but is further introducing uncertainty to the simulation, which was only addressed in some previous studies [5]. Techniques like fitting a moving average to the obtained time series or data driven models for outlier detection should therefore be employed for countering the issue of contact flickering.

4. Conclusion

Previous studies on tool wear calculation with FEA found that tool wear can be calculated with wear models and the chip-tool interaction can be modeled via node displacement of the tools mesh and the geometry. These models were applied during the steady-state of cutting and less effort was put into simulating the transient machining process before the thermo-mechanical steady-state is reached. However, when developing wear resistant coatings, the transient phase becomes dominant for processes like profile grooving. Since previous modelling techniques based on the thermo-mechanical steady-state can't be effectively applied on the transient wear modelling, a new methodology is presented in this work based on DEFORM 2D. Due to a fully automated python script which controls the FEM software in text based mode via the command prompt, an algorithm was successfully implemented which accumulates the wear induced on the tool during the transient cutting phase via a discretized node displacement. Therefore, algorithms for matching the tools mesh with the geometry were successfully implemented. After a user-specified time step, the simulations database is altered upon the application of a wear rate model via text editing in python before the simulation is automatically continued. With the proposed approach for a fully automated wear modelling, the basis for a more in depth understanding of the transient cutting phase is provided. In future studies the presented approach will be utilized for the determination of a suitable wear model to describe an experimentally determined wear behavior of AlTiN coated cutting inserts for the cutting process of grooving.

Statement: During the preparation of this work the authors used ChatGPT in order to improve readability and language. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgements

The scientific work has been supported by the DFG within the research priority program SPP 2402. The authors thank the DFG for this funding and intensive technical support.

References

[1] Kurada S, Bradley C. A review of machine vision sensors for tool condition monitoring. Comput Ind 1997; 34: 55–72.

[2] Rech J, Claudin C, Polly P, Courbon C. New aspects of metrology of frictional behaviour in metal cutting. Mechanik 2016;11:1751- 1753. doi: 10.17814/mechanik.2016.11.520

[3] Li, B. (2017) A review of tool wear estimation using theoretical analysis and numerical simulation technologies. Int. Journal of Refractory Metals and Hard Materials 2017; 35:143–151. doi:10.1016/j.ijrmhm.2012.05.006

[4] Korkmaz ME, Gupta MK. A State of the Art on Simulation and Modelling Methods in Machining: Future Prospects and Challenges. Archives of Computational Methods in Engineering 2023;30:161–189. doi: 10.1007/s11831-022-09794-9

[5] Binder M, Klocke F, Doebbeler B. An advanced numerical approach on tool wear simulation for tool and process design in metal cutting. Simulatiuon Modelling Practice and Theory 2017;70:65-82. doi: 10.1016/j.simpat.2016.09.001

[6] Sadeghifar M, Sadeghati R, Jomaa W, Songmene V. A comprehensive review of finite element modeling of orthogonal machining process: chip formation and surface integrity predictions. The International Journal of Advanced Manufacturing Technology 2018;96:3747–3791. doi: 10.1007/s00170-018-1759-6

[7] Cepellini C, Abeni A. Development and implementation of crater and flank tool wear model for hard turning simulations. The International Journal of Advanced Manufacturing Technology 2022;120:2055–2073. doi: 10.1007/s00170-022-08885-y

[8] Attanasio A, Ceretti E, Fiorentino A, Cappellini C, Giardini C. Investigation and FEM-based simulation of tool wear in turning operations with uncoated carbide tools. Wear 2010;269:344-350. Doi: 10.1016/j.wear.2010.04.013

[9] Methon G, Courbon C, M'Saoubi R, Girinon M, Rech J. Numerical modeling of wear in orthogonal cutting of multilayer coated tools. Procedia Cirp 2021;102:67-72. doi: 10.1016/j.procir.2021.09.012

[10] Liu Z, Yue C, Li X, Liu X, Liang XY, Wang L. Research on Tool Wear Based on 3D FEM Simulation for Milling Process. J. Manuf. Mater. Process 2020;4:121. doi:10.3390/jmmp4040121

[11] Attanasio A, Veretti E, Rizzuti S, Umbrello D, Micari F. 3D finite element analysis of tool wear in machining. CIRP Annals 2008;57:61-64. doi: 10.1016/j.cirp.2008.03.123

[12] Abeni A, Cappellini C, Attanasio A. Finite element simulation of tool wear in machining of nickel-chromiumbased superalloy. ESAFORM 2021; doi: 10.25518/esaform21.4302

[13] Hosseinkhani K, Ng EG. A Unique Methodology for Tool Life Prediction in Machining. J. Manuf. Mater. Process. 2020;4:16. doi: 10.3390/jmmp4010016

[14] Zhang Z, Klippel H, Kneubühler F, Afrasiabi M, Röthlin M, Kuffa M, Bambach M, Wegener, K. Study on the effect of wear models in tool wear simulation using hybrid SPH-FEM method. Procedia CIRP 2023;117:414-419. doi: 10.1016/j.procir.2023.03.070

[15] Rech J, Giovenco A, Courbon C, Cabanettes F. Toward a new tribological approach to predict cutting tool wear, CIRP Annals 2018;67:65-68. doi: 10.1016/j.cirp.2018.03.014

[16] Rech J, Claudin C, D'Eramo E. Identification of a friction model - Application to the context of dry cutting of an AISI 1045 annealed steel with a TiN-coated carbide tool. Tribology International 2009;42:738-744. doi: 10.1016/j.triboint.2008.10.007

[17] Binder M, Klocke F, Lung D. Tool wear simulation of complex shaped coated cutting tools. Wear 2015;330:600-607. doi: 10.1016/j.wear.2015.01.015

[18] Karkalos NE, Markopoulos A. Determination of Johnson-Cook material model parameters by an optimization approach using the fireworks algorithm. Procedia Manufacturing 2018;22:107-113. doi: 10.1016/j.promfg.2018.03.017

[19] Jaspers SPFC, Dautzenberg, JH. Material behaviour in conditions similar to metal cutting: flow stress in the primary shear zone. Journal of Materials Processing Technology 2002;122:322-330. doi: 10.1016/S0924-0136(01)01228-6