euspen's 21st International Conference & Exhibition, Copenhagen, DK, June 2021
www.euspen.eu

A modular node-based modeling platform for the simulation of machining processes

Bernd W. Peukert[1], Andreas Archenti[1,2]

[1]*KTH Royal Institute of Technology, Department of Production Engineering, Manufacturing and Metrology Systems Division, 11428 Stockholm*
[2]*KTH Royal Institute of Technology, Department of Sustainable Production Development, Industrial Dependability Division, 15181 Södertälje*

bpeuke@kth.se

Abstract
Increasing precision requirements motivate research on the system perspective of machining systems. Virtual environments for the prediction of the Process Machine Interaction (PMI) are one key to obtain a priori understanding of machining accuracy, save expensive testing and planning time of industrial users, and increase the overall resource utilization. This paper introduces the modular node-based software platform SharpCut (#cut). The platform follows a sequential modeling approach of manufacturing processes and machine tools by using nodes. The novelty of the method lies in its open architecture with a strong focus on extendibility, adaptability, and research applicability.

Besides the node-based modeling approach, four implemented software modules are described. The four modules are the workpiece discretization, the tool and cutting inserts, the material removal, and the system's kinematics. The machine tool's motion and kinematic errors are modeled with Homogeneous Transformation Matrices (HTMs). The workpiece discretization is performed with a memory-efficient implementation of Depth Elements (Dexels). The material removal follows from the modelling of the Tool-Workpiece Engagement (TWE) with a generic model of the tool and the computationally efficient Möller-Trumbore ray-triangle intersection algorithm. With the presented modules, it is possible to predict the results of machining processes and perform sensitivity or root cause analyses. The paper concludes with an example model and simulation of a face-milling process.

Virtual Machine Tool, Virtual Machining, Process Simulation

1. Introduction

Physics-based domain knowledge on the kinematic, static, dynamic, and thermo-elastic behavior is vital for understanding the structural behavior of machine tools [1]. However, a holistic approach also needs to consider the superposition of the static sequential deviation effects of each machine tool component as well as the dynamic effects in the structural feedback loop of the tool-workpiece engagement (TWE), i.e., the Process-Machine-Interaction (PMI) [2, 3].

Besides the process dynamics, also the Computer Numerical Control (CNC) capabilities affect the feedback loop. Previous approaches by Fesperman et al. [4] use a generalized machine tool model and include the kinematics as well as the controller. Theissen et al. [5] present a similar approach and include the static behavior of machine tools. Schützer et al. [8] present a virtual machine tool for the holistic simulation of micro-milling processes by incorporating different modules for the respective physical domains.

Brecher and Trofimov [6] use machine and process models for the simulation of the parallel milling with two spindles. Hömberg et al. [7] present an approach to use virtual machine tools for the prediction of the process stability. Okafor [9] gives a recent overview on the state of the art as well as modeling examples for kinematic errors of three different machines with a virtual CNC machine tool.

The previous works highlight the deep understanding of the underlying physics for modelling and simulating the machine tool mechanics. This paper builds on top of this knowledge and presents an intuitive way of building research-oriented machine tool models based on nodes.

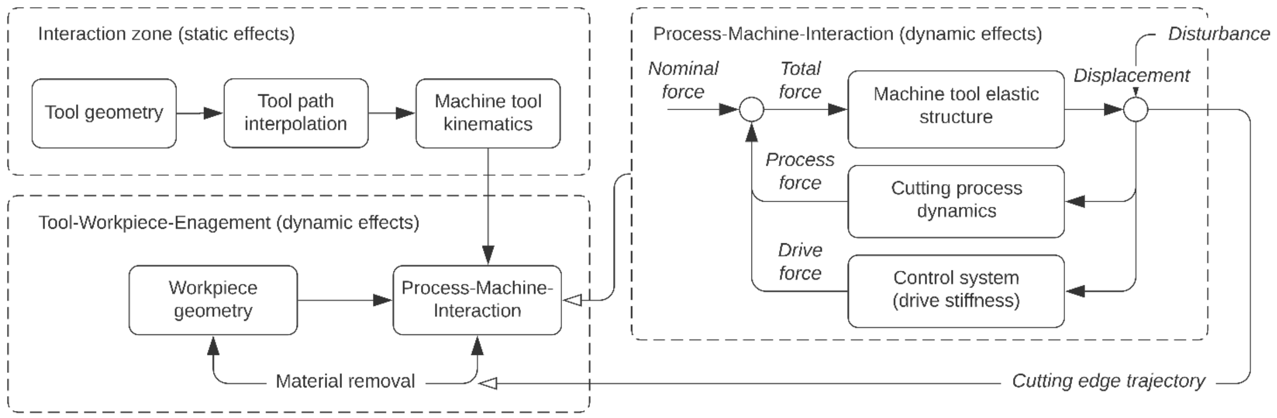


Figure 1. Function block diagram of static and dynamic effects during the machining

2. Methodology

Fig. 1 displays the function block diagram of the machining process. The ideal interaction zone's definition consists of the spatial position of the tool geometry, the generated tool path, and the machine tool kinematics including the axis deviations. The TWE is a dynamic effect that requires the modeling of the PMI together with a material removal simulation. The PMI is not within the scope of this paper but is exemplified here for completeness.

2.1. Node-based modeling

The #cut platform implements a node editor to represent the different modules' connectivity, also referred to as nodes. The node editor is based on the open-source `qtpynodeeditor` framework written by Ken Lauer. Commercial simulation environments such as ANSYS Workbench and MATLAB Simulink inspired the user interface.

Node-based modeling approaches are especially suited for machining simulations, as they force system thinking and emphasize the sequential character of the various contributors. As a research-oriented approach, the focus is on extendibility and root cause investigations.

Generic mathematical nodes accompany the machining-specific nodes to increase the customizability of the models, e.g., to model spindle imbalances, tool wear, or environmental disturbances. Fig. 2 displays the node editor with the mathematical nodes and the context menu for modeling machine tools and machining processes. The machining-specific nodes are described in the subsequent Sec. 2.2.

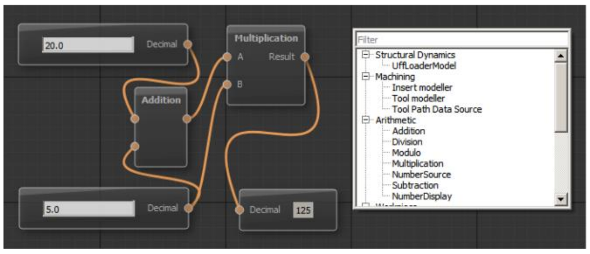


Figure 2. Context menu with generic mathematical functions for the custom model generation

2.2. Physics-based modelling nodes

2.2.1 Kinematics

Homogeneous Transformation Matrices (HTMs) are suitable for the modeling of kinematic chains as well as component and location errors. The approach is well-known in the field of robotics and machine tools, and many researchers successfully applied it for the modeling of kinematics, see [10] for further details and references in the literature.

The HTM transforms a vector from one coordinate system a into another coordinate system b, see Eq. 1. It consists of a linear transformation matrix $\boldsymbol{T}$ and a rotation matrix $\boldsymbol{R}$.

$$^{a}\boldsymbol{HTM}_{b} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ 0 & 1 \end{bmatrix} \qquad (1)$$

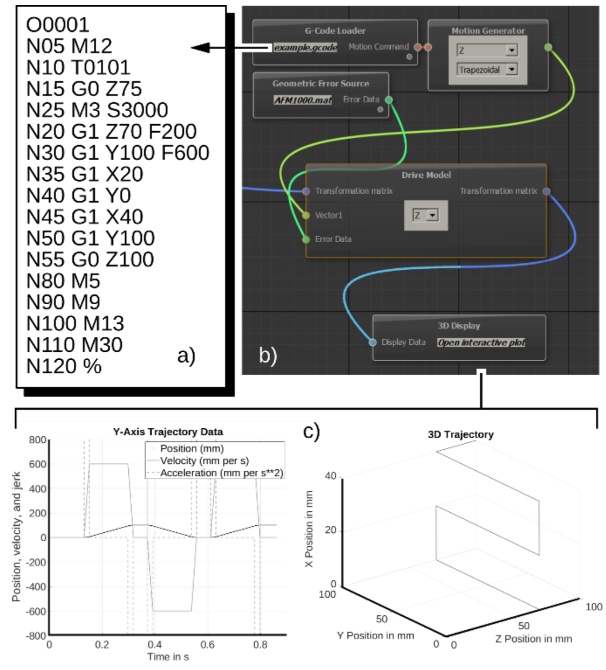
The multiplication of the HTM in Eq. 1 allows for the sequential transformation through multiple coordinate systems. Hence, in practice, the machine tool kinematics are modeled by building a chain of different static and variable transformations in the form of the Eq. 1.

For variable transformations, e.g., actuated drives, the #cut implementation uses a variable representation of the components in the rotary and linear transformation parts of the HTM in Eq. 1. The function is implemented in the drive node.

Besides the source transformation, the variable transformation also has an additional input to load different drive commands as well as the axis' location and component errors, $\boldsymbol{LOC}$ and $\boldsymbol{COM}$, respectively. These errors are internally also modeled with HTMs, see Eq. 2.

$$^{a}\boldsymbol{HTM}_{b} = {}^{a}\boldsymbol{LOC}_{b} \; {}^{a}\boldsymbol{COM}_{b} \qquad (2)$$

Frist, a basic G-code (ISO 6983) parser extracts the motion commands and generates the drive commands. Second, a motion generator interpolates the respective axes' inputs according to the selected triangular or trapezoidal motion profile. Fig. 3 exemplary depicts the implementation of the modeling nodes and a generated toolpath.

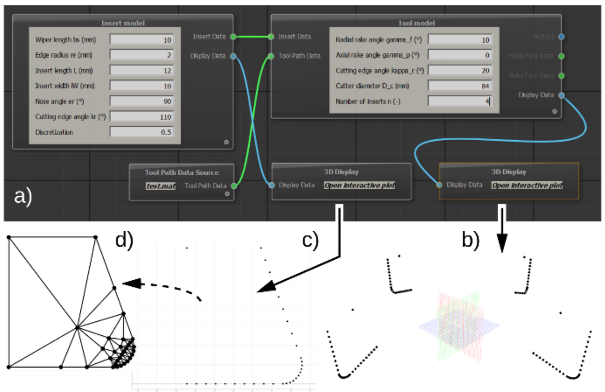


```
O0001
N05 M12
N10 T0101
N15 G0 Z75
N25 M3 S3000
N20 G1 Z70 F200
N30 G1 Y100 F600
N35 G1 X20
N40 G1 Y0
N45 G1 X40
N50 G1 Y100
N55 G0 Z100
N80 M5
N90 M9
N100 M13
N110 M30
N120 %
```

Figure 3. Modeling of kinematics; a) G-code source for motion generation, b) node schematic of three-axis machine tool kinematic (excerpt) with loaded geometric errors, c) visualizations of the y-axis motion state variables and the generated 3D tool trajectory

2.2.2 Cutting inserts and tools

Kaymakci, Kilic, and Altintas [11] introduce a unified cutting force model alongside procedures for the modeling of tools and inserts. The model describes the cutting edge geometry with analytical equations based on standardized tool and insert geometry data (ISO 13399), and positions the cutting insert in the tool with HTMs. #cut uses the same approach but also allows for the direct extraction of the cutting edge when the tool's and the insert's Computer-Aided Design (CAD) data is available.



Insert model	
Wiper length bs (mm)	10
Edge radius re (mm)	2
Insert length L (mm)	12
Insert width W (mm)	10
Nose angle er (°)	90
Cutting edge angle kr (°)	110
Discretization	0.5

Tool model	
Radial rake angle gamma_f (°)	10
Axial rake angle gamma_p (°)	0
Cutting edge angle kappa_r (°)	20
Cutter diameter D_c (mm)	84
Number of inserts n (-)	4

Figure 4. Tool and insert definition; a) node schematic in the node editor, b) 3D plot of the tool, c) 2D plot of the cutting edge discretization, d) visualization of the rake face triangulation

After extracting the cutting edge geometry, #cut performs a line discretization and builds a triangulated model of the insert's rake face. The triangulation is based on the open-source library `Triangle` written by Jonathan SHEWCHUK. The triangulation refines automatically towards the tool radius to improve the approximation of the round edge.

Whenever the insert is spatially repositioned, the module automatically produces a triangulated flank face on the output. Fig. 4 depicts the node's implementation together with the generated insert and tool model as well as the rake face triangulation.

2.2.3 Workpiece discretization

The discretization of the workpiece is an important step for the simulation of the material removal during machining. Previous researchers present approaches based on the voxelization of the workpiece surface [12] or the use of depth-elements [13] (Dexels). The #cut platform follows a similar path as described in [14], which models the workpiece with a structured grid of parallel vectors $\boldsymbol{d}_{\text{in}}^{\text{out}}$, spanning from a dexelization plane outside the workpiece into the material volume, see Fig. 5 for a depiction of the approach.

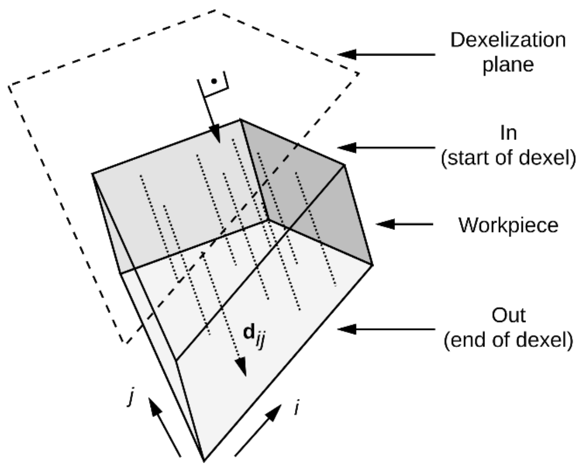


Figure 5. Depiction of the dexelization approach for the discretization of workpieces in the i, j plane

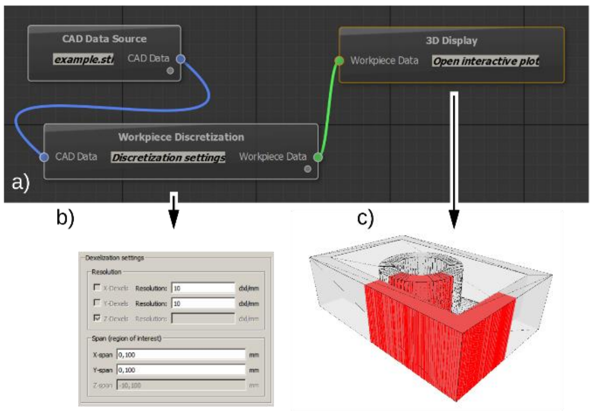
The discretization is based on a parallel implementation of the MÖLLER-TRUMBORE raycasting algorithm [15], which is widely used in computer graphics for calculating ray-triangle intersections. In this case, the Dexel itself is the ray, and a triangulated surface, e.g., from CAD data in the form of Standard Triangulation Language (STL) files, is used to locate the material of the workpiece.

One advantage of a structured grid over an unstructured grid is the efficient storage in the computer memory. The spatial position of each Dexel i, j is known from the grid size s , and only scalars for the span covering the workpiece material are stored or updated during the material removal simulation, see Eq. 3 and Eq. 4.

$$\boldsymbol{d}_{ij}{}_{\text{in}}^{\text{out}} = \boldsymbol{d}_{k}{}_{\text{in}}^{\text{out}} = [x_{in,1} \quad x_{\text{out},1} \quad \cdots \quad x_{\text{in},n} \quad x_{\text{out},n}] \qquad (3)$$

$$k = f(i,j) = i + s \cdot j \qquad (4)$$

To decrease the sensitivity to discretization errors, #cut implements a multi-Dexel approach allowing for up to three perpendicular dexelization planes and variable resolutions along the i, j axes of the planes. Fig. 6 depicts the node implementation with a workpiece blank loaded from CAD data and a single axis discretization. Within the discretization settings, the user can specify the Dexel grid's resolution per unit length (typically in mm) and select a subset of the CAD geometry for the discretization.



Dexelization settings

Resolution

☐ X-Dexels	Resolution:	10	dxl/mm
☐ Y-Dexels	Resolution:	10	dxl/mm
☑ Z-Dexels	Resolution:		dxl/mm

Span (region of interest)

X-span	0, 100	mm
Y-span	0, 100	mm
Z-span	-10, 100	mm

Figure 6. Workpiece discretization; a) node schematic in the node editor, b) discretization settings, c) single-axis discretization

2.2.4 Tool-workpiece engagement

The TWE describes the actual material removal during the machining process. The goal is to translate the cutting edge trajectory into a change in the discretized workpiece contour. The #cut approach implements an interaction node, which uses the above referenced MÖLLER-TRUMBORE algorithm to check the intersection of the stored Dexels with the rake and flank faces of an insert triangulation.

The material removal is estimated by analyzing the Dexels in and out coordinates during a raycast along each Dexel. There are five possible cases detected which then transmit a respective output signal for the update of the workpiece discretization. The cases are depicted in Fig. 7.

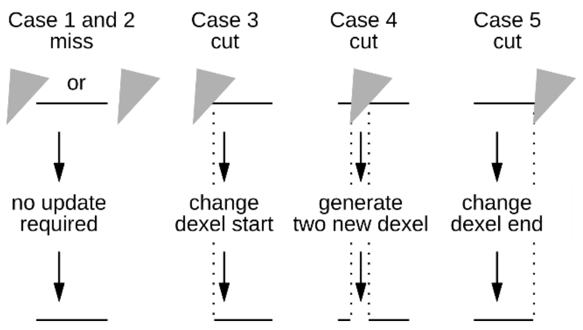
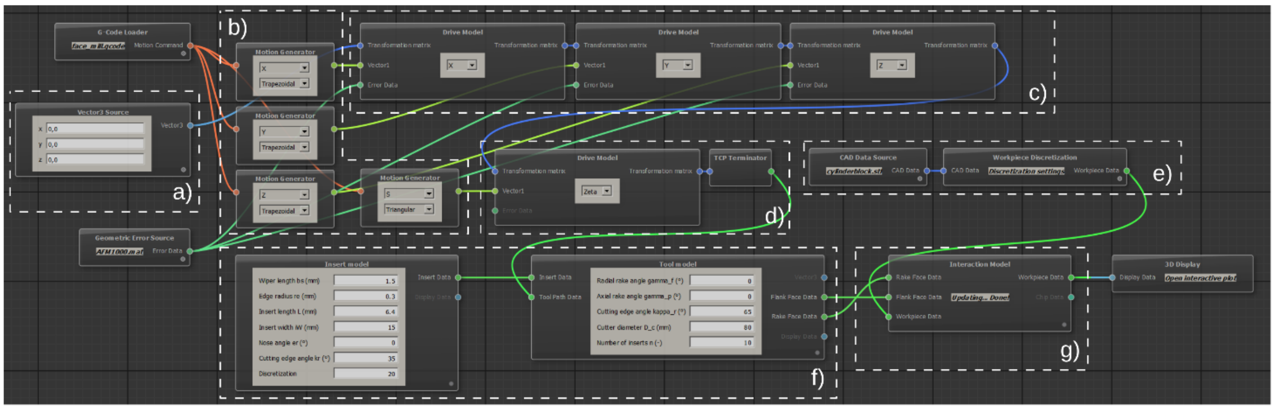


Figure 7. Dexel-insert interaction with their respective signals for the implementation of material removal

The detected Dexel-insert interactions are clustered in point clouds on the cutting inserts' rake and flank faces to estimate the instantaneous chip area. Then, the respective alpha shapes are calculated and sent to the output port of the interaction node. This output could be further processed by connection to, e.g., mechanistic cutting models, which is not within the scope of this paper. Together with dynamic models of the machine tool structure and the CNC, this information would allow for the modeling and simulation of the PMI as shown in Fig. 1.

3. Exemplary use case

A face-milling process of a cylinder block is selected to demonstrate the #cut modeling approach. The cylinder block is a CAD model of a Suzuki Hayabusa made by Samu SUOJANEN. Fig. 8 depicts the generated node schematic for modeling a basic three-axis milling machine tool type AFM R1000 by the company AFM, Andrychów, Poland, and the material removal simulation of a face-milling process.



Vector3 Source: x 0,0; y 0,0; z 0,0

Motion Generator: X, Trapezoidal
Motion Generator: Y, Trapezoidal
Motion Generator: Z, Trapezoidal
Motion Generator: S, Triangular

Drive Model: X
Drive Model: Y
Drive Model: Z
Drive Model: Zeta

Insert model	
Wiper length bs (mm)	1.5
Edge radius re (mm)	0.3
Insert length L (mm)	6.4
Insert width iW (mm)	15
Nose angle er (°)	0
Cutting edge angle kr (°)	35
Discretization	20

Tool model	
Radial rake angle gamma_f (°)	0
Axial rake angle gamma_p (°)	0
Cutting edge angle kappa_r (°)	65
Cutter diameter D_c (mm)	80
Number of inserts n (-)	10

Figure 8. Node schematic of the exemplary face-milling of a cylinder block in a three-axis milling machine tool; a) World Coordinate System, b) motion generators for G-code interpolation, c) linear axis models, d) spindle model and TCP kinematic terminator, e) Computer-Aided Design data loading and Dexel discretization, f) insert and tool model (CoroMill R365 with R365-1505ZNE-PM 4220 insert by company SANDVIK COROMANT, Sandviken, Sweden), g) tool-workpiece engagement

The first modeling step is the generation of the machine tool kinematics as depicted in Fig. 8a) and c). The motion is then transmitted to the axes and spindle, see Fig. 8b) and d). Fig. 8e) shows the discretization of the workpiece from CAD data, Fig. 8f) is the modeling of the insert and the tool. The outputs are finally fed to the TWE node, see Fig. 8g). Fig. 9 shows the results of the simulation of the cut surface. As no dynamics are involved in this example, the results can be understood as the ideal cutting surface resulting solely from the machine tool axes errors, the cutting process parameters, and the chosen cutting insert.

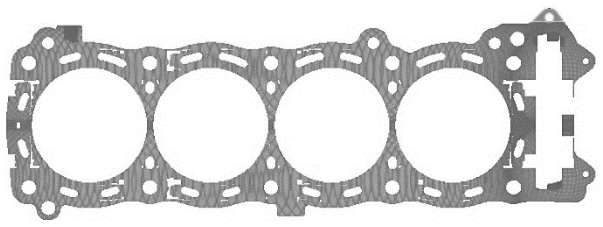


Figure 9. Visualization of the simulated cutting surface (ideal roughness, as only kinematics data is used in this simulation)

4. Summary and outlook

Modular, node-based modeling platforms are very well suited for the modeling and simulation of machine tools and machining processes. The interaction of the relevant physical domains can be modeled via a chain of function blocks. These function blocks are visualized as nodes and allow for the visual scripting of machine tool and process models. The software architecture is highly extensible and allows for the implementation of new custom nodes. The same nodes can also be used to model other machines, e.g., industrial manipulators.

The presented #cut framework compiles previous research work in reusable modules and accommodates the machine-specific modeling functions with general mathematical modeling nodes for the exploration of design changes, root causes or process parameters. The future work will focus on the realistic verification of manufacturing processes. Therefore, new nodes for the dynamic effects are required for the modeling of the PMI as described in Fig. 1. The node editor will be extended to support feedback loops and use ordinary differential equation solvers.

In its current state, #cut does not offer the reliability and simplicity expected from industry-viable software. However, after further maturing, it is planned to eventually release the modeling platform as an open-source research tool in the upcoming years.

Acknowledgment

The authors would like to thank VINNOVA, the Swedish innovation agency, for funding this research as a part of the research project: A simulation-based guide to machinability assessment (Reference number 2016-05397). The authors would also like to thank the Centre for Design and Management of Manufacturing Systems for their support.

References

[1] F. Koenigsberger and J. Tlusty, *Machine tool structures*, vol. **1**. Pergamon Press, 1970.

[2] C. Brecher, M. Esser, and S. Witt, "Interaction of manufacturing process and machine tool," *CIRP Ann. – Manuf. Technol.*, vol. **58**, no. 2, pp. 588–607, 2009.

[3] C. Brecher *et al.*, "Modeling and Simulation," in *Process Machine Interactions*, Springer Heidelberg, pp. 29–51, 2013.

[4] R. R. Fesperman, S. P. Moylan, G. W. Vogl, and M. A. Donmez, "Reconfigurable data driven virtual machine tool: Geometric error modeling and evaluation," *CIRP J. of Manuf. Science and Techn.*, vol. **10**, pp. 120–130, 2015.

[5] N. Theissen, T. Laspas, K. Szipka, and A. Archenti, "Virtual machining system simulator: Analysis of machine tool accuracy," *Proc. Manuf.*, vol. **25**, pp. 338–343, 2018.

[6] C. Brecher, Y. Trofimov, and S. Bäumler, “Holistic modelling of process machine interactions in parallel milling,” *CIRP Ann. – Manuf. Technol.*, vol. **60**, no. 1, pp. 387–390, 2011.

[7] D. Hömberg, E. Uhlmann, O. Rott, and P. Rasper, "Development of a Stability Prediction Tool for the Identification of Stable Milling Processes," in *Process Machine Interactions*, Springer Heidelberg, pp. 203–224, 2013.

[8] K. Schützer, L. W. da S. de L. Ramos, J. Mewis, M. O. Tamborlin, and C. R. Baldo, "Simulation tool for prediction of cutting forces and surface quality of micro-milling processes," *Int. J. of Adv. Manuf. Techn.*, vol. **99**, no. 1–4, pp. 225–232, 2018.

[9] A. C. Okafor, "Virtual CNC machine tool modeling and machining simulation in high speed milling", in *High-Speed Machining*, Elsevier Academic Press, 2020.

[10] N. A. Theissen, "Physics-based modelling and measurement of advanced manufacturing machinery' s positioning accuracy," Licentiate Thesis, Kungliga Tekniska högskolan, 2019.

[11] M. Kaymakci, Z. M. Kilic, and Y. Altintas, "Unified cutting force model for turning, boring, drilling and milling operations," *Int. J. of Mach. Tools and Manuf.*, vol. **54–55**, pp. 34–45, 2012.

[12] M. Witt, M. Schumann, and P. Klimant, "Real-time machine simulation using cutting force calculation based on a voxel material removal model," *Int. J. of Adv. Manuf. Techn.*, vol. **105**, no. 5–6, pp. 2321–2328, 2019.

[13] S. Stifter, "Simulation of NC machining based on the dexel model: A critical analysis," *Int. J. of Adv. Manuf. Techn.*, vol. **10**, no. 3, pp. 149–157, 1995.

[14] B. Denkena, T. Grove, and O. Pape, "Optimization of complex cutting tools using a multi-dexel based material removal simulation," Procedia CIRP, vol. **82**, pp. 379–382, 2019.

[15] T. Möller and B. Trumbore, "Fast, Minimum Storage Ray-Triangle Intersection," J. of Graph. Tools, vol. **2**, no. 1, pp. 21–28, 1997.