CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF MECHANICAL ENGINEERING

DEPARTMENT OF PRODUCTION MACHINES AND EQUIPMENT

MASTER'S THESIS

EFFECTIVE CONTROL OF CNC MILLING MACHINE TOOL WHEN

MACHINING COMPLEX PARTS

2022 **Filip Bartoš**

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Bartoš Filip**	Personal ID number:	**467317**
Faculty / Institute:	**Faculty of Mechanical Engineering**		
Department / Institute:	**Department of Production Machines and Equipment**		
Study program:	**Mechanical Engineering**		
Branch of study:	**Production Machines and Equipment**		

II. Master's thesis details

Master's thesis title in English:

Effective Control of CNC Milling Machine Tool when Machining Complex Parts

Master's thesis title in Czech:

Efektivní řízení frézovacího CNC stroje při obrábění tvarově složitých dílců

Guidelines:

Topic description: The student will work on an algorithm for dynamic control of technological conditions in machining of complex parts, combining two approaches with respect to machine tool CNC control: a) speed control to achieve a constant cutting speed, b) feed-rate control with respect to the angular velocity. Syllabus of work: A) Overview of the state of the art in the field of dynamic control of techn. conditions in multi-axis machining. B) Search for parameters and characteristics of CNC milling machines influencing the possibilities of using dynamic speed and feed-rate control. C) Proposal of solution variants. D) Creation of an algorithm for feed-rate correction. E) Implementation of the algorithm into the postprocessor. F) Proposal and realization of machining tests with dynamic control of tech. conditions. G) Analysis of the effects of dynamic control of technology conditions on productivity and roughness of the machined surface.; Extent of the text part: 60 - 80 pages; Scope of the graphic part: selected flowcharts.

Bibliography / sources:

SOUČEK, P.: Servomechanismy ve výrobních strojích. 1. vydání. Praha: ČVUT, 2004. 210 s. ISBN 80-01-02902-6.; VAVRUŠKA, P.: Technologické nadstavby postprocesorů pro víceosé CNC stroje. Praha, 2013. Disertační práce na ČVUT v Praze. Fakulta strojní. Ústav Výrobních strojů a zařízení.; Li, J., Kilic, Z. M., Altintas, Y.: General Cutting Dynamics Model for Five-Axis Ball-End Milling Operations. ASME. J. Manuf. Sci. Eng. 2020; 142(12): 121003.

Name and workplace of master's thesis supervisor:

Ing. Petr Vavruška, Ph.D., Department of Production Machines and Equipment, FME

Name and workplace of second master's thesis supervisor or consultant:

Ing. Tomáš Kozlok, TOS VARNSDORF

Date of master's thesis assignment: **21.10.2021** Deadline for master's thesis submission: **03.01.2022**

Assignment valid until: **20.02.2022**

Ing. Petr Vavruška, Ph.D. Supervisor's signature	Ing. Matěj Sulitka, Ph.D. Head of department's signature	prof. Ing. Michael Valášek, DrSc. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt Student's signature

Declaration

I hereby submit for the evaluation and defense the master's thesis elaborated at the CTU in Prague, Faculty of Mechanical Engineering.

I have no relevant reason against using this schoolwork in the sense of § 60 of Act No. 121/2000 Coll. on Copyright and Rights Related to Copyright and on Amendment to Certain Acts (the Copyright Act)

I declare I have accomplished my final thesis by myself and I have named all the sources used in accordance with the Guideline on ethical preparation of university final theses.

Prague, 3.1.2022 ....

Filip Bartoš

Acknowledgement

Firstly, I would like to say thank you to all my colleagues from RCMT and CTU, that gave and shared the knowledge with me that was needed to finish this thesis and also for their help with the measurements and evaluation of the results. Furthermore, I am grateful for the help and supervision from Petr Vavruška, who discussed with me every step of the solutions and gave me precious advices.

Finally, I would like to express my gratitude for the support that my family gave me during my studies.

The topic of this master's thesis is funded by the support from the Czech Ministry of Education, Youth and Sports under the project CZ.02.1.01/0.0/0.0/16_026/0008404 “Machine Tools and Precision Engineering” financed by the OP RDE (ERDF). The project is also co-financed by the European Union.

Anotace

Autor:	Filip Bartoš
Název:	Efektivní řízení frézovacího CNC stroje při obrábění tvarově složitých dílců
Rozsah práce:	85 stránek, 71 obrázků, 6 tabulek, 8 příloh
Školní rok vyhotovení:	2022
Škola:	ČVUT, Fakulta strojní
Ústav:	Ú12135 – Ústav výrobních strojů a zařízení
Vedoucí DP:	Ing. Petr Vavruška, Ph.D.
Konzultant DP:	Ing. Tomáš Kozlok
Klíčová slova:	optimalizace, NC kód, dokončovací obrábění, postprocesor, řízení otáček vřetene, řízení posuvových rychlostí, dynamické obrábění
Anotace:	Tato práce navazuje na existující diplomové a bakalářskou práci, které byly zaměřeny na vývoj vlastních optimalizačních funkcí. Rešerše této práce mapuje současný stav optimalizací obráběcího procesu v oblastech CAM, nadstavbách CAM a řídicích systémech se zaměřením na dodržování technologických parametrů. Na základě předchozích řešení jsou provedeny návrhy možných úprav stávajících funkcí a realizace vybraných variant v postprocesoru a C#. Dále jsou aplikovány upravené optimalizační funkce na reálném dílci. Závěrem je diskutován přínos jednotlivých zhotovených optimalizací na strojní čas, opotřebení nástroje a výslednou kvalitu povrchu.

Annotation

Autor:	Filip Bartoš
Title:	Effective Control of CNC Milling Machine Tool when Manufacturing Complex Parts
Extent:	85 pages, 71 figures, 6 tables, 8 appendices
Academic year:	2022
University:	CTU, Faculty of Mechanical Engineering
Department:	Ú12135 Department of Production Machines and Equipment
Supervisor:	Ing. Petr Vavruška, Ph.D.
Consultant:	Ing. Tomáš Kozlok
Key words:	optimization, NC code, finishing machining, postprocessor, spindle speed control, feed rate control, dynamic machining
Annotation:	This thesis follows already existing theses which were focused on developing custom optimization functions. The first section presents the state of the art of the optimizations for the cutting processes implemented into CAM and CNC controls. Based on the previous versions of the optimization functions, several options for possible improvement of the existing functions are presented. The selected option is programmed in C# and implemented into the postprocessor. Followed by a demonstration of the altered optimization function on a real part. At the end the benefits of the optimization functions on the machining time, tool wear and surface roughness are discussed.

Contents

Overview of Used Units

a	[min^{-2}]	Acceleration
$\vec{b}$	[-]	Solution vector
$A_{1,2}$	[mm]	Intersection points
A_x	[mm]	X coordinate of point A
A_y	[mm]	Y coordinate of point A
A_z	[mm]	Z coordinate of point A
B	[mm]	Following contact point
B_x	[mm]	X coordinate of point B
B_y	[mm]	Y coordinate of point B
B_z	[mm]	Z coordinate of point B
c	[mm]	Point on a plane
d	[mm]	Tool diameter
d_1	[mm]	Plane constant
d_2	[mm]	Plane constant
$\vec{e}$	[-]	Tool axis vector
f_n	[mm/min]	Feed rate
f_z	[mm]	Feed per tooth
f_{opt}	[mm/min]	Optimized feed rate
$\vec{j}$	[-]	Normal vector
k_v	[s^{-1}]	Speed gain
n	[rpm]	Spindle speed
$\vec{n}$	[-]	Unit normal vector
n_{lim}	[rpm]	Limit spindle speed
n_{prev}	[rpm]	Previous spindle speed
p_i	[mm]	Input points
r_1	[mm]	Circle radius
r_2	[mm]	Circle radius
R_{ratio}	[mm]	Radius ratio
$S_{cx,y,z}$	[mm]	Sphere center
s	[mm]	Distance
Δs	[mm]	Positional error
T	[-]	Transformation matrix
t	[min]	Time
$\vec{v}$	[-]	Directional vector

v_{is}	[m/min]	Current velocity
v_c	[m/min]	Cutting speed
$\vec{x}^*$	[-]	Least-squares solution
z	[-]	Number of teeth

List of Acronyms

AI	Artificial intelligence
AFC	Adaptive feed control
CAD	Computer aided manufacturing
CAM	Computer aided machining
CL data	Cutter location data
CNC	Computer numerical control
CP	Contact point
DoC	Depth of cut
NC	Numerical control
OCM	Optimized contour milling
UDE	User defined event

1 Introduction and Goals of the Thesis

Machining is a complex process, where the input material is being transformed into a required part. Machining is generally used for parts, which consist of complex shapes with high demands on surface quality. These parts are also often made of a difficult-to-cut materials, which need to be manufactured with reasonable productivity. One option how to produce a high-quality complex geometrical shapes and parts is the milling technology.

To achieve a satisfactory productivity and surface quality, the Computer Numerical Control (CNC) is nowadays the most used option. If the part is from the geometrical perspective a complex shape, the tool path preparations are done solely using a Computer Aided Manufacturing (CAM) software, where some process optimizations may also take place. Much work has focused on the optimization of roughing operations, but little attention has been paid to the of finishing operations. Therefore the motivation for this work is to reduce the machining times, improve surface quality and to save money while using the actual machine tool and cutting tools for finishing operations.
This master's thesis is about algorithms of optimization functions, their modifications, and applications, which compensate cutting conditions e.g., cutting speed and feed rate and also about their verification on various milling machine tools.

The main goals of this thesis are to program an algorithm that can optimize cutting speed during milling with the consideration of a remaining stock material and an algorithm that optimizes feed rate values based on the tool path and the part surface geometry. Another goal is to concatenate these two principles together and implement them into a postprocessor along with a control panel and logic. Lastly, the optimizations have to be verified during machining process and the pros and cons of the optimizations evaluated and summarized on a test part.

The first section of the thesis includes state of the art of the optimization principles and commercially available products. Furthermore presents the influence of a machine tool and its control system onto the cutting speed and feed rate control.

In the second section, the custom optimization algorithms are proposed, and the best options are selected and implemented into a postprocessor. Moreover, the optimizations verifications are presented along with a machining time, tool life and surface roughness evaluation.

2 Machining Process Optimizations

Nowadays, there is a high demand to produce huge numbers of parts with costs pro part as low as possible, therefore a lot of companies primarily in Automotive and Aviation industry tend to use the most modern software and tools that are currently on the market. Given this fact, there is also a lot of companies that try to satisfy this demand with cutting edge optimization tools for machining to reduce machining time, improve part surface quality and tool life.

All the machining process optimizations that are commercially available on the market are presented in this chapter.

The main areas where the optimizations take place are CAM software and Machine Control Systems.

2.1 CAM Software Optimizations

The big companies in this market sector such as Autodesk, Dassault Systèmes, Siemens PLM Software, CNC Software and others offer already built-in manufacturing strategies that use some of the known optimization principles such as Constant Tool Engagement, Maximal Tool Cutting Length and Corner Feed Rate Reduction. Those principles are mostly used for roughing operations, where Material Removal Rate is the priority.

Following roughing operations, there are also optimization principles for finishing operations, which work with dynamical feed rate control and air cuts reduction.

Constant Tool Engagement

This optimization principle considers the diameter of the tool and the part geometry. Using this information an optimized toolpath is created for roughing operations, which maintains constant cutting tool engagement along the whole

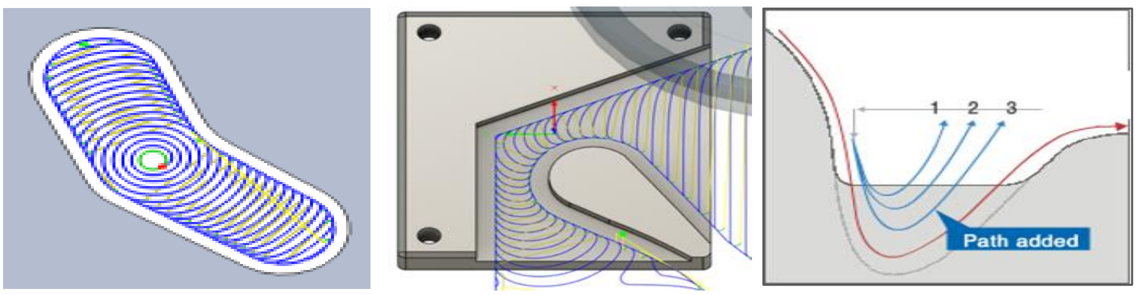


Figure 1 – Trochoidal tool paths and regions with added paths (cadcamlessons, 2019); (Autodesk, 2016); (NCBrain, 2017)

operation. Widely seen example of this principle are adaptive toolpaths or regions with added tool paths as seen in Fig. 1.

Maximal Tool Cutting Length

With the advancement of the modern production machines and tool materials, there is also a new trend of roughing operations which utilize the whole cutting length of a tool while reducing the radial stepover and dramatically increasing the cutting speed (Fig. 2), which results into lower cutting forces during machining. This strategy is often used together with Constant Tool Engagement principle to assure the best material removal rates and tool life. (Fritz, 2015)

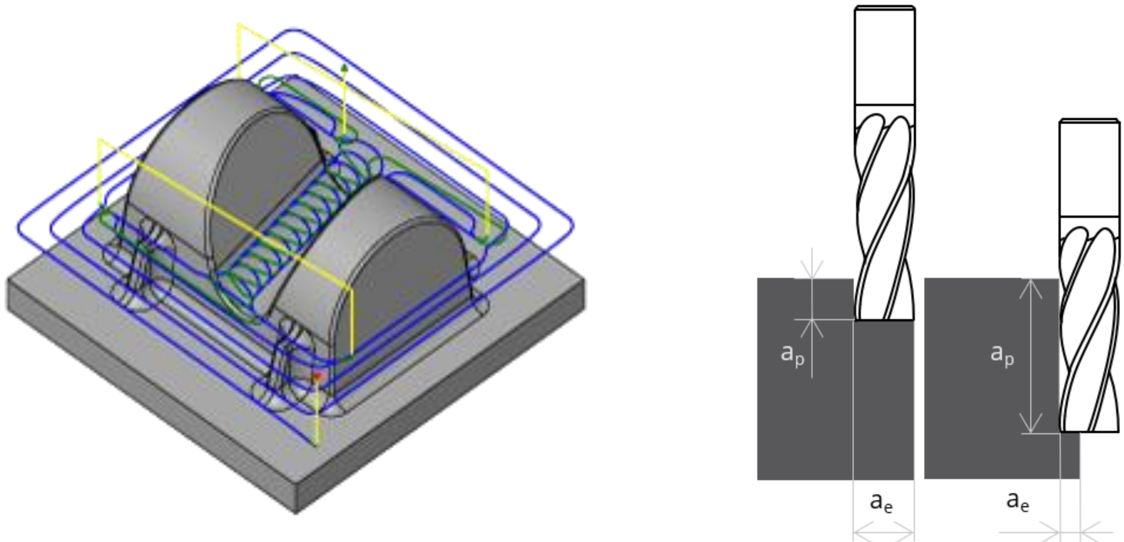


Figure 2 - Maximal Tool Cutting Length example operation and principle (Autodesk, 2016)

Corner Feed Rate Reduction

Another method how to improve effectivity and ensure smooth machining process is Corner Feed Rate Reduction. As can be seen in the Fig. 3, the user has to manually fill in the Target Material Removal Rate and the Feed Rate Adjustments. The software then does the calculations and returns an optimal feed rate for the given tool positions so that the material removal rate stays constant.

The same optimization principle can be used with different input parameters, e.g. Feed Adjustment on Arcs as can be seen in Fig. 4, which recognises G02 circular interpolations in the generated toolpath and based on the radius value adjusts the feed rate value by Compensation Factor specified by the user while considering whether its convex or concave shape.

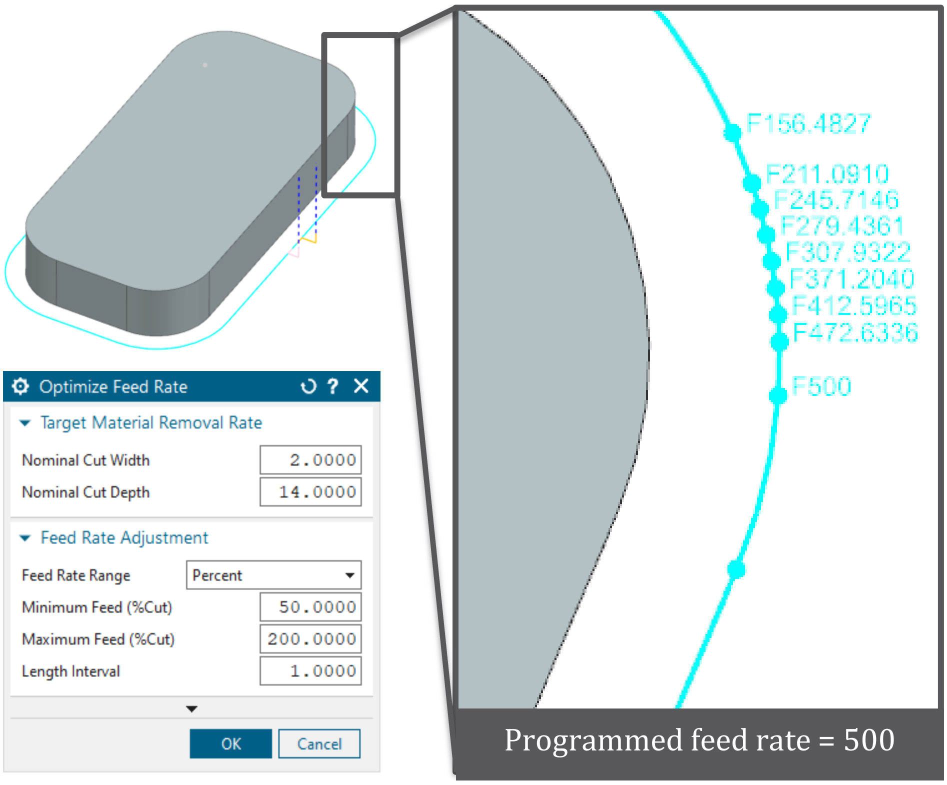


Figure 3 - Corner Feed Rate Reduction - Maximal MRR

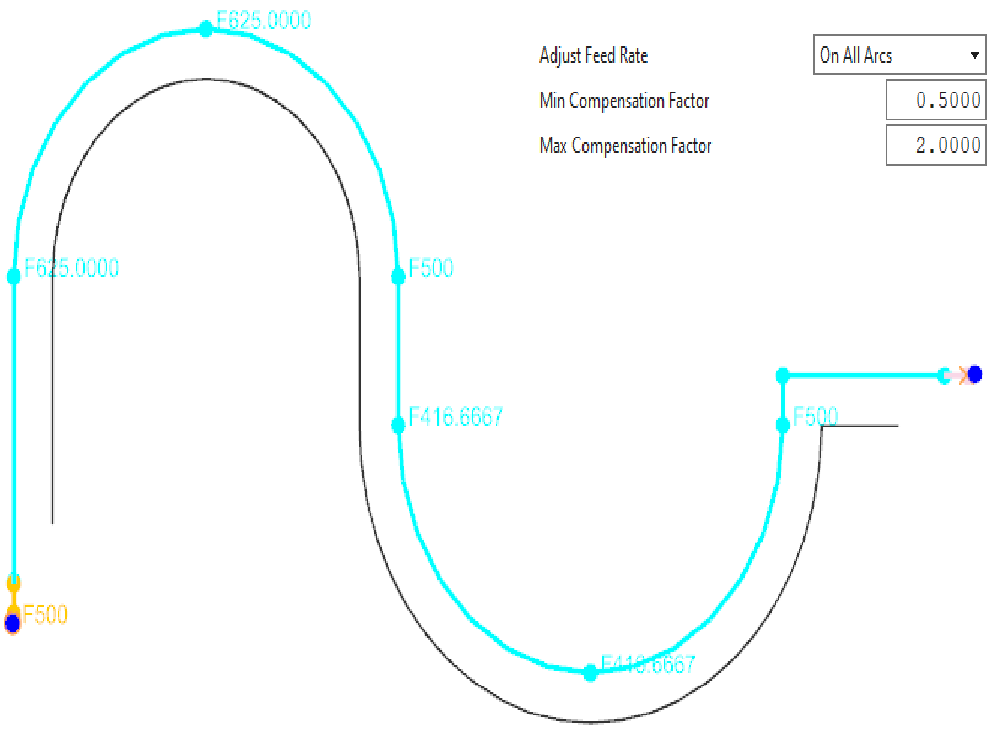


Figure 4 - Corner Feed Rate Reduction - All arcs compensation

As the last input option is the specification of the Maximum Directional Change or Minimum Radius as can be seen in Fig. 5.

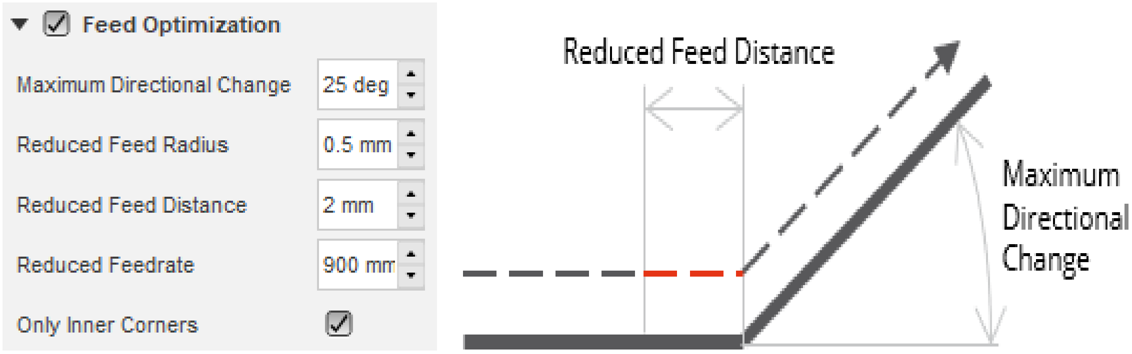


Figure 5 - Corner Feed Rate Reduction - Maximum directional change

Dynamical Feed Rate Control

While finishing 3D geometries either on a 3-axis or 5-axis milling machine or with a Ball, Barrel or Corner Radius Endmill, there could be some situations where the remaining stock material is not equally distributed along the part surface (Fig. 7). This could cause serious problems during machining in regard of tool breakage or reduced productivity of the whole process because of under valuated feed rates. To keep the productivity as high as possible, companies ICAM and CGTech came with optimization modules for finishing machining called SmartCut™ and OptiPath. These modules calculate the material removal rate at each point of the toolpath and according to this value adjust the feed rate value (Fig. 6). As a last step they overwrite the feed rate values in the NC code. (CGTech, 2010) (Icam, 2010)

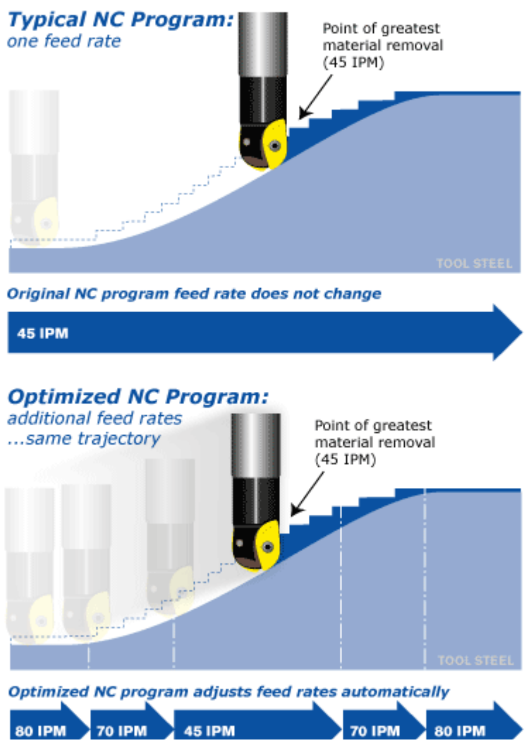


Figure 6 - OptiPath module demonstrator (CGTech, 2010)

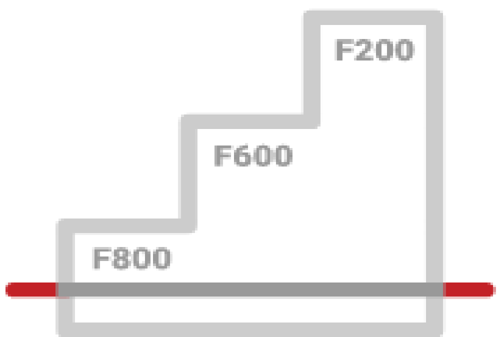


Figure 7 - Unequally distributed material along the tool path (Icam, 2010)

Toolpath Filtering

Every toolpath is created either by linear interpolations (G01 commands), circular interpolations (G02 and G03 commands) or more sophisticated geometries such as splines. In order to ensure a smooth tool motions during machining, the generated point density of an NC code should be generally low (but must not sacrifice accuracy) in order to reduce the number of inputs for Positional Regulation and thus enhance the motion smoothness. Such filtering is available directly in CAM software, where the principle is to replace collinear lines with one line and tangent arcs as can be seen in Fig. 8. (Brecher, 2021)

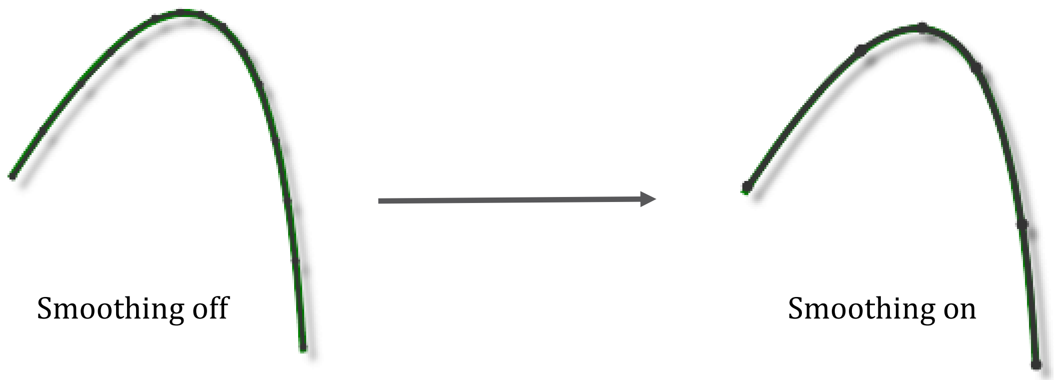


Figure 8 - Tool path filtering example (Autodesk, 2016)

Air Cuts Reduction

During various manufacturing strategies in CAM can happen that several toolpaths do not engage with the material. These are referred to as 'Air Cuts'.

In order to save machining time, ICAM presented their SmartCUT™ module which detects those areas and replaces the motions with rapid (G00 command) or with high-feed commands (G01 with high F value) as can be seen in the Fig. 9. (Icam, 2010)

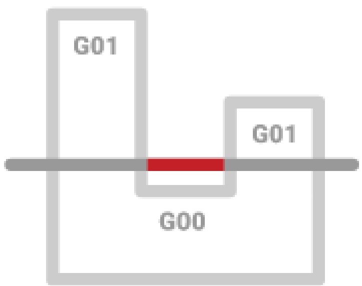


Figure 9 - Air Cuts reduction principle (Icam, 2010)

Cutting Force Calculation

CGTech presented their optimization module Vericut Force, which computes cutting forces during simulation of the NC code using well known equations e.g. Kienzle or Altintas Force Model.

Based on the calculated force values, the machining parameters can be changed to achieve better cutting conditions for given process and thus reduce machining times and increase tool life. (CGTech, 2010); (Altintas, 2012)

AICam

This software is being developed by the korean company NCBrain. Its first release was presented in 2017 and it uses AI algorithms to generate machining strategies and toolpaths without user interaction. The target market sector seems to be mold industry. The customer has to set up their own database with usable tools, tool holders, materials and machining parameters for each tool and material or use the built-in database which provides some general samples to correctly generate NC codes that can be later used on a production machine tool. The softwares' presented workflow is depicted in Fig. 10.

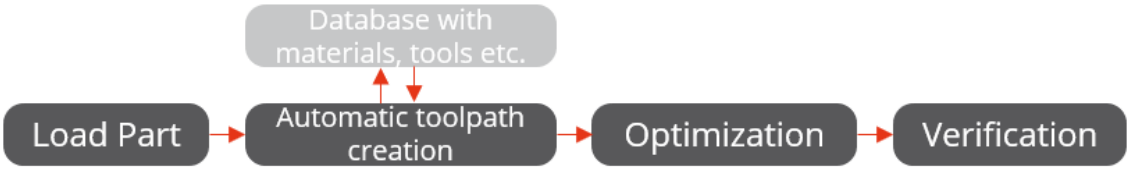


Figure 10 - NCBrain workflow (NCBrain, 2017)

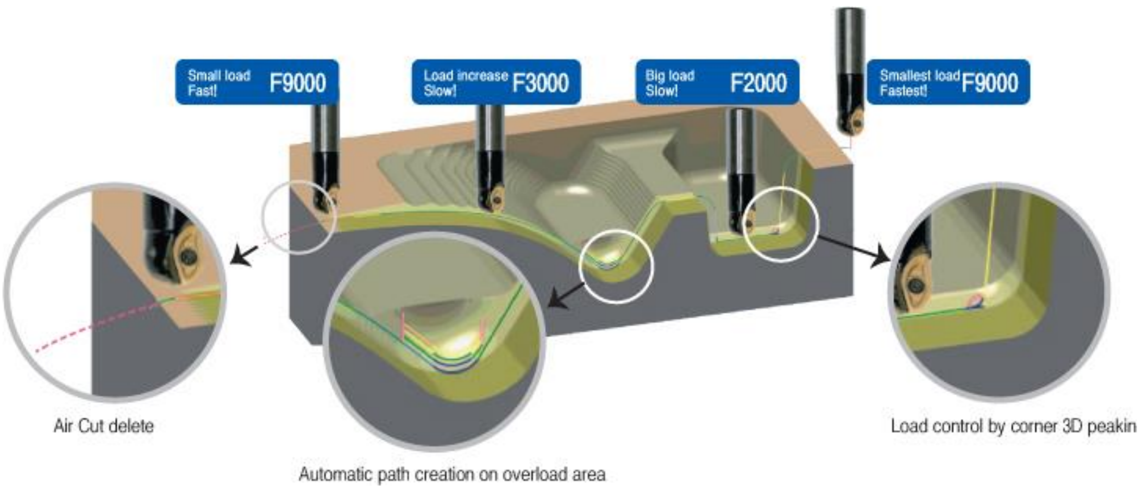


Figure 11 - NCBrain functionality (NCBrain, 2017)

The optimization part in Fig. 11 provides for example Dynamical Feed Rate Control, Air Cuts Reduction and many other functions.

Machine DNA

This Plug-in for CAM software PowerMill from Autodesk is a handy tool which helps with creation of a digital copy to existing machine on a shop floor with its current

condition. Not every machine tool has the same travel limits, kinematics, drive parameters and with that related axis dynamics and many more characteristics, which are generally not saved in CAM software. This digital copy can be used for more realistic Material Removal Simulations in CAM. (Technickytydenik, 2013)
Firstly, the user has to conduct a measurement of the condition of the milling machine using already pre-generated diagnostic file, which can be run on the machine tool. All the data during axis movement are collected from machine tool's sensors and those data can be later evaluated directly in PowerMill. As a result, PowerMill is able to automatically set up ideal parameters for each machining strategy such as Minimal Trochoidal Toolpath Radius, Toolpath Point Distribution, Corner Feed Rates etc. (Autodesk, 2020)

SmartPath

The company ICAM also presented their SmartPATH package, which aims to ease the moving of an NC program to another machine with different kinematics, travel and size limits. Regular process would be to re-program the toolpaths in CAM but with SmartPATH the user only has to re-postprocess the strategies in order to be able to use them on a different machine. (Icam, 2010)

NC Program Verification

When users do not want to test their already postprocessed NC codes directly on the machine tool either because of costs ineffectiveness or because of the high risk of machine tool crash potential then this verifiation (Fig. 12) is a next step after postprocessing CAM strategies into a NC code. It takes the generated NC code and runs real time virtual simulation of the machining process directly on the specified machine. It can also validate machine macros (parametrically written NC programms) and machine collisions. (NCSimul, 2012)

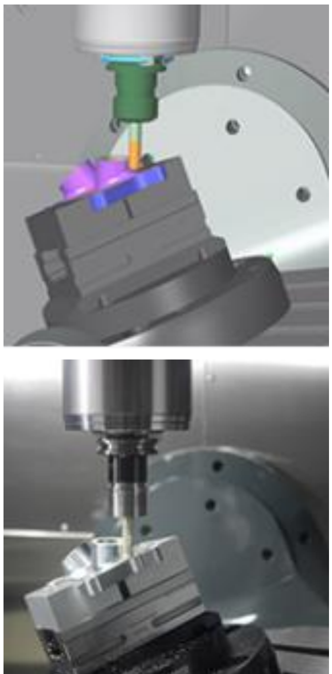


Figure 12 - NCSimul versus real machining (NCSimul, 2012)

This verification is available for example in CAD/CAM software Siemens NX, and CAM software extensions such as NCSimul, ICAM, Vericut and Machine Works.

2.2 Machine Control System Optimizations

The next option, how to optimize machining processes are algorithms which are directly implemented in the control system of a production machine. Those optimizations are mostly developed by the companies selling the control system itself. The main companies in this market sector are Fanuc, Siemens and Heidenhain.

Adaptive Feed Control

This optimization function (AFC) considers the spindle load during machining, which directly correlates with cutting forces. If the spindle load gets constantly higher, it could be a sign of a worn-out tool. AFC monitors this and lowers the feed rate so the cutting force stays constant. When the feed rate drops under pre-set value, the control system will stop the axis movements and return error message on the operating panel that the tool must be changed for a new one. This function could be combined along with trochoidal toolpaths to achieve reliable machining process and reduce machining time as can be seen in Fig. 13. (Heidenhain, 2013)

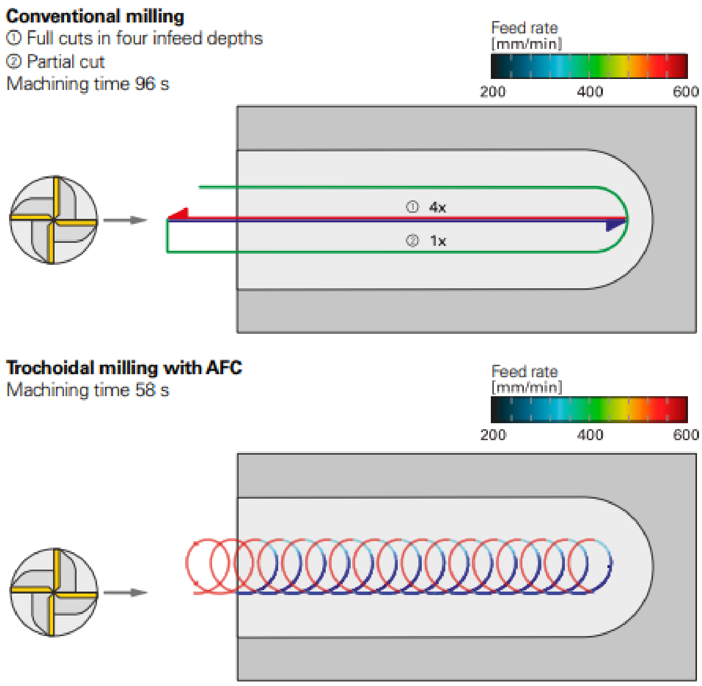


Figure 13 - Adaptive feed control with trochoidal milling (Heidenhain, 2013)

Adaptive Chatter Control

As chatter could be a significant limiting factor in some machining processes among others such as thermal and mechanical stability of the tool or available spindle power, the company Heidenhain developed an optimization function (ACC) to counter this phenomenon as well. The working principle is to get signals from the machine tool drives and thus evaluate whether the process is stable or unstable.

With the help of the drives itself, the vibration energy could be withdrawn from the process (Fig. 14). However, using the drives to counter the vibrations has a drawback of a frequency range limit up to 100Hz, which can be still effectively damped. (Heidenhain, 2013)

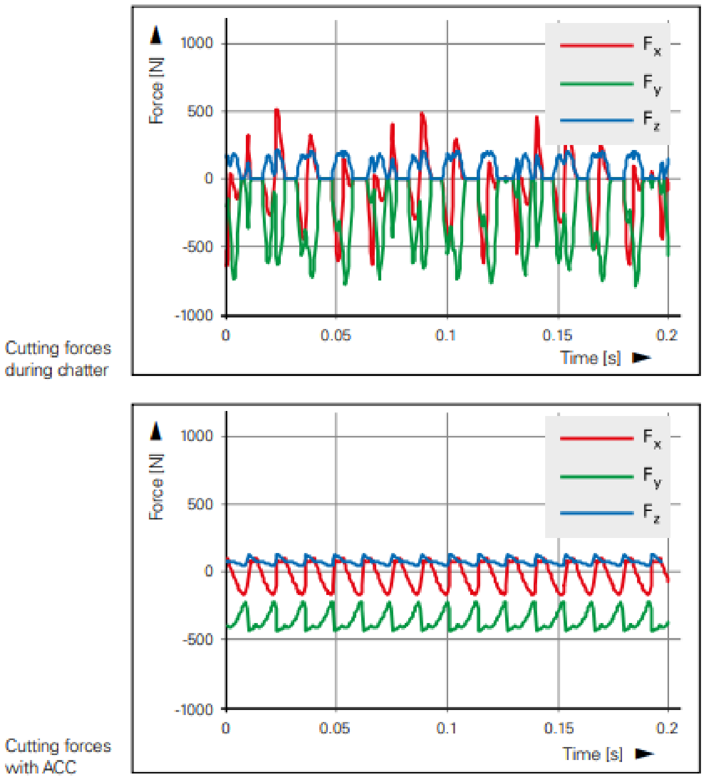


Figure 14 - Adaptive chatter control example (Heidenhain, 2013)

Optimized Contour Milling

Another product from the Heidenhain company is the Optimized Contour Milling (OCM). This software option offers functions for efficient roughing, finishing and deburring of pockets using adaptive tool paths, which maintain constant engagement parameters during the whole operation. The programmed feed rates are also adjusted for the finishing operation based on the angular velocities of the tool path (Fig. 15). The contours can then be either defined using cycle parameters or directly with CAD import into the control system. (Heidenhain, 2020)

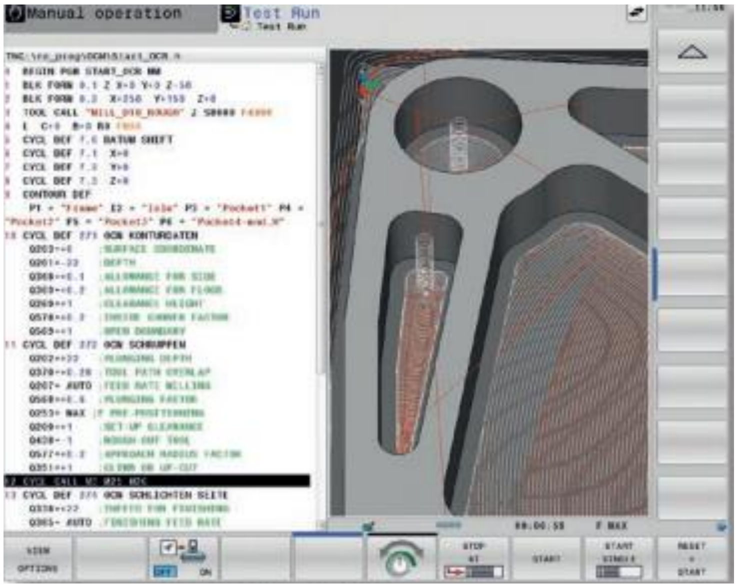


Figure 15 - Optimized Contour Milling setup (Heidenhain, 2020)

Spindle Speed Variation

Another optimization directly implemented in control system is Spindle Speed Variation (SSV). SSV is used to avoid chatter during machining by continuously changing the spindle rpm. This function is available on all lathes produces by the company Haas. The function SSV changes the rpm continuously within user-specified range with a constant duty cycle (Fig. 16). This rpm change is periodical and in this case the function is Sinus. The SSV is turned on and off directly in NC code with a M35/36 command and its setting can be adjusted in the control system as well as can be seen in Fig. 17 - parameters 165 and 166. Parameter 165 specifies the amplitude of the Sinus function and parameter 166 the cycle time. (Haas, 2015)

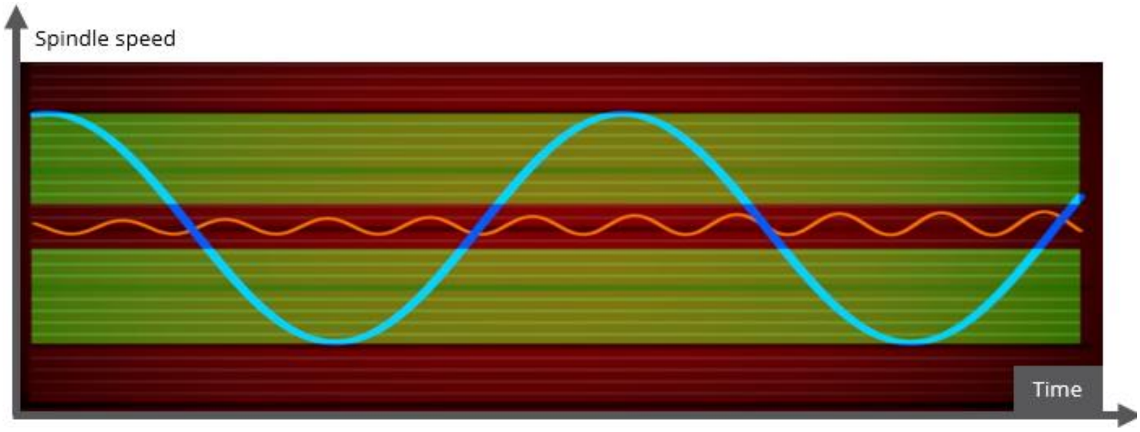


Figure 16 - Spindle rpm with SSV turned on (Haas, 2015)

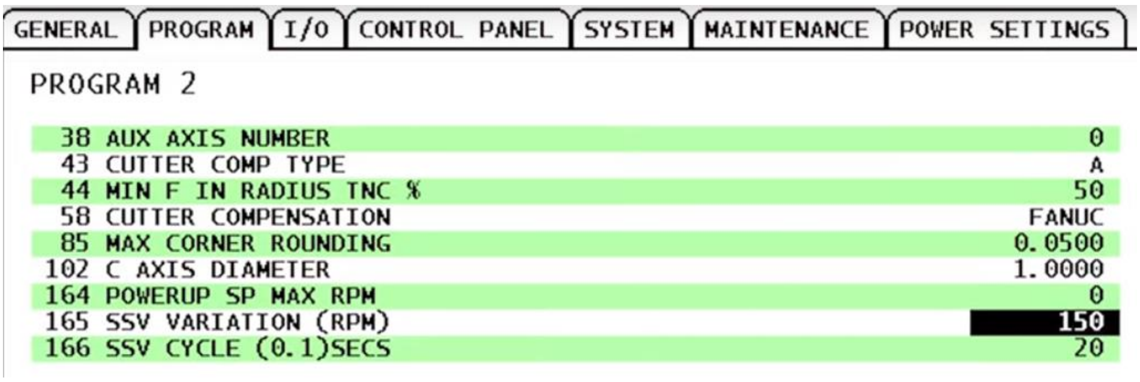


Figure 17 - SSV setting on the control panel (Haas, 2015)

3 Research Papers

This chapter focuses on a summary of all related research papers, which have already been published in the area of spindle speed and feed rate optimizations. Summarized principles in this chapter are usually not commercially available. The optimization principles have mostly been proven on selected parts but are yet to be used in the industry.

3.1 Scientific Publications

Investigation of the Dynamic Behavior of Machine Tools During Cutting by Operational Modal Analysis

Berthold et al. investigated the influence of a spindle speed change on the dynamic behavior of a milling machine tool. Furthermore, he compared the results of the Experiment Modal Analysis (EMA) with the Operational Modal Analysis (OMA) in order to investigate the differences between cutting and standstill in terms of a modal behavior. EMA was used for a machine tool in standstill, where some phenomena that occur only during the cutting process are neglected. For a machine tool under loads was then OMA the selected method. Those two methods show similarities in the natural frequencies but differ significantly in the damping values. (Berthold, et al., 2018)

The selected excitation process for the OMA was a face milling with three different spindle speed settings. The first setting was a constant spindle speed of 800 rpm, the second was a spindle speed in the range 1000 – 3000 rpm and the third was a sine function spindle speed alteration. Presented results show that the sine spindle speed alteration is not only the best one for the OMA but also avoids the critical frequencies during milling. (Berthold, et al., 2018)

On the Stability of High-Speed Milling with Spindle Speed Variation

Seguy et al. explored the spindle speed variation possibilities for high-speed milling during regenerative chatter effect. He altered the spindle's rpm with a given frequency and amplitude. The investigated variation functions were Sine, Triangular and Square wave. The Triangular function showed the best results for the testing, because the spindle dynamics was the limiting factor and with this function, they could use bigger frequency and amplitude interval. Then they proved that the spindle is indeed capable of such rpm changes (Fig. 18) with an error of 0,5% to programmed value. (Seguy, et al., 2009)

Later, they proved the matematical model for the regenerative effect for the given workpiece and then they have chosen rpms and depth of cut (DoC) in an unstable domain. It was proved that spindle speed variantion had a positive effect on the chatter suppression and a bigger DoC can be achieved while using spindle speed variation. (Seguy, et al., 2009)

At the end they declared that spindle dynamics play a significant role in the optimization process and the better the dynamic characteristics are, the better the optimization itself. (Seguy, et al., 2009)

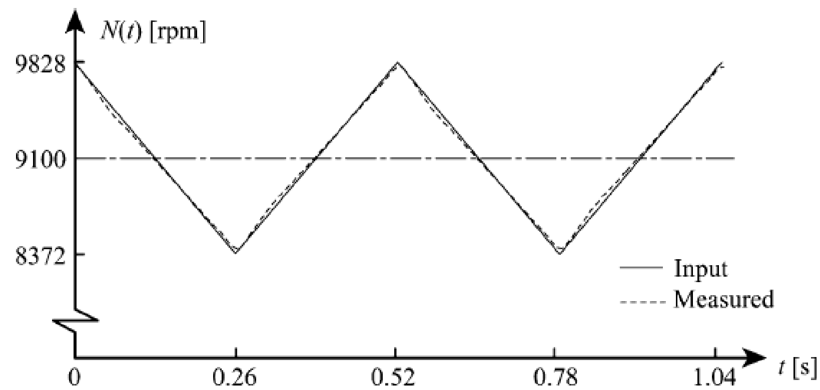


Figure 18 - Spindle rpm change using triangular function with measured error (Seguy, et al., 2009)

Reducing Machining Time by Pre-Process Control of Spindle Speed and Feed Rate in Milling Strategies

Vavruška et al. presented a cutting speed optimization function for milling free-form shapes with a ball and radius end mills on point milling strategies. As the effective tool diameter may not stay constant during machining, a way how to recalculate the spindle speed to compensate for the changing effective diameter is presented and the altered spindle speeds are automatically outputted for each NC block into an NC code. They also tested the optimization during milling of a part and measured the spindle speed values and motion axes velocities. The results indicated an increase in productivity by approx. 43%. (Vavruska, et al., 2018)

CAD/CAM Interfaced Algorithm Reduces Cutting Force, Roughness and Machining Time in Free-Form Milling

Käsemodel et al. demonstrated a very similar cutting speed optimization function for milling free-form shapes with a ball end mill as Vavruska et al. During machining of a free-formed shape the effective tool diameter does not stay constant, and this phenomenon will be discussed later in the next chapter, but the calculation principle used in this optimization is the same. (Käsemodel, et al., 2020)

The function takes several parameters as an input, such as Normal Vector, Feed per Tooth, programmed cutting Speed, max rpm and already generated NC code.

Käsemodel et al. developed a custom software using Visual Basic and Grasshopper to recalculate the cutting speed for each machine position and when the rpm difference due to the effective tool diameter change was bigger than 50 compared to the programmed value, they altered the NC code block with a command to set a new spindle rpm and recalculated the feed rate for that block. They also mention that the spindle speed alteration does not affect the spindle life. The presented results show up to 30% machining time reduction along with better surface quality and cutting force reduction. (Käsemodel, et al., 2020)

The Analysis of Tool Life and Wear Mechanisms in Spindle Speed Variation Machining

Albertelli et al. investigated the sinusoidal spindle speed variation (SSV) on a lathe. The variation is used for a chatter suppression during machining. They measured the temperatures between tool and part during conventional machining and spindle speed variation and presented an increase in temperature during ssv. This increase in temperature had a negative effect on a tool life.

They furthermore proved that there are new wear mechanisms while using ssv. (Albertelli, et al., 2014)

Optimization of Tool Axis Orientations in Multi-Axis Toolpaths to Increase Surface Quality and Productivity

Stejskal et al. presented an optimization of a tool lead angle, where he shows the impact of the tool lead angle change from 0-14 degrees on the surface quality. According to the Fig. 19 the minimal lead angle should be at least four degrees, in order to achieve a suitable surface quality. (Stejskal, et al., 2020)

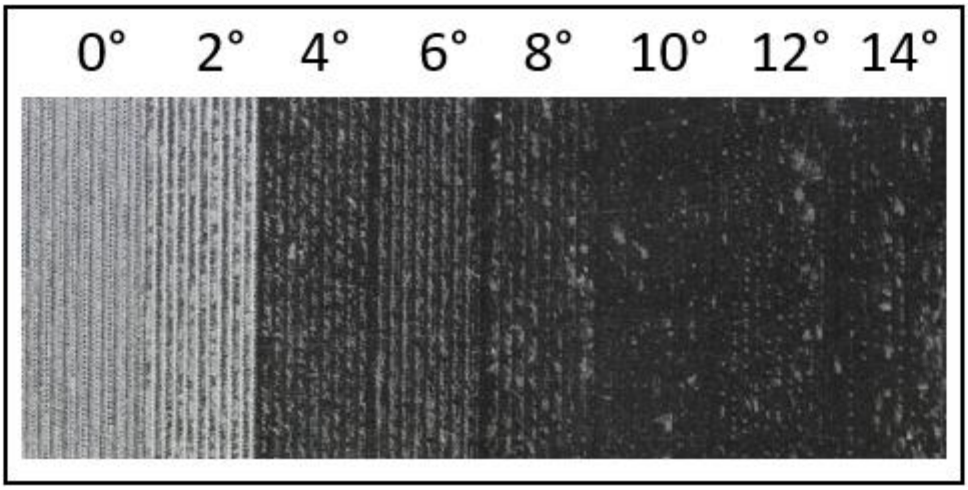


Figure 19 - Influence of the tool lead angle of the surface quality (Stejskal, et al., 2020)

3.2 Optimization of cutting conditions by CTU in detail

This sub-chapter presents and further explains the optimization functions that have been developed at CTU in Prague, Faculty of Mechanical Engineering, Department of Production Machines and Equipment in the field of cutting speed and feed rate optimizations during 3 axis and 5 axis milling. This thesis builds on those principles.

Cutting Speed Optimization

Stejskal proposed and later tested an algorithm for effective diameter calculation during free-form milling using ball and corner radius endmills, which has also been published as a journal paper under the name 'Reducing Machining Time by Pre-Process Control of Spindle Speed and Feed Rate in Milling Strategies' by Vavruska et al. The algorithm is written directly in postprocessor (gray tile in the Fig. 20) and is used for continuous spindle speed alteration during milling. The algorithm is used to further enhance the quality of finishing operations during machining process,

Figure 20 - Location of the Cutting Speed Optimization in the CAD/CAM workflow

where the part tolerances, surface quality and surface integrity are the most important parameters. (Stejskal, 2017)

The calculation of the effective diameter (radius) for 3 axis and 5 axis milling is depicted on the Fig. 21. As an input it is necessary to know the Contact Point – CP, the normal vector to the part surface $\vec{n}$, the Tool Tip Point – TT and the Tool Axis Vector $\vec{e}$. For five axis machining the Tilt Angle β is further required. Using those

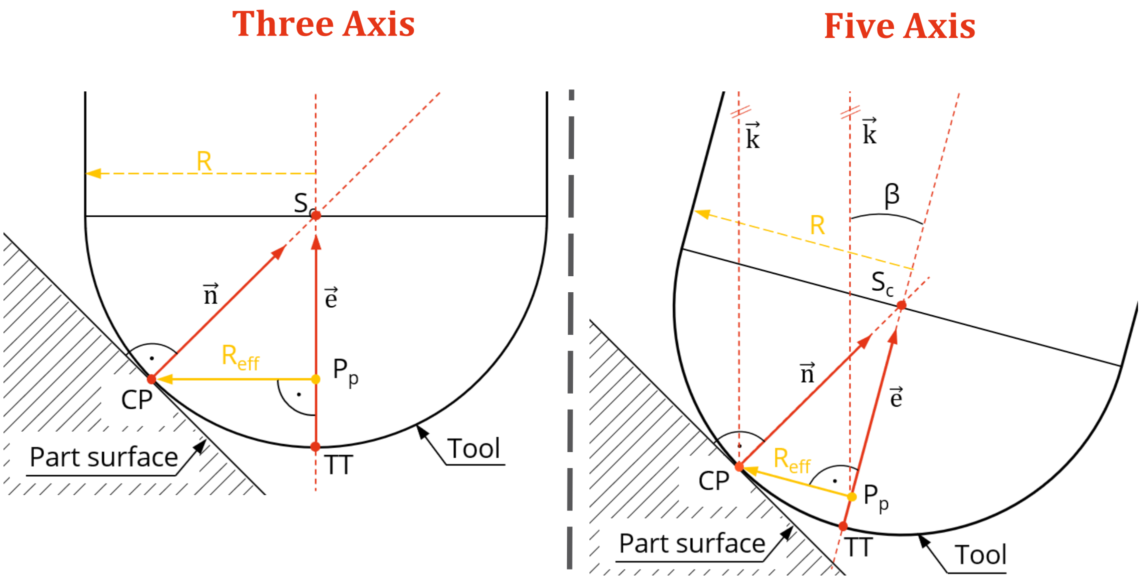


Figure 21 - Effective diameter calculation principle (Stejskal, 2017)

data, the Tool Center Point - S_c can be calculated and later the Effective Radius R_{eff}.

In order to get the required inputs especially the Contact Point, there is for example an option in CAD/CAM software Siemens NX under Path Settings – Non Cutting – More – Output Contact Data to turn this feature on as can be seen on the Fig. 22. If the feature is on, those data are then accessible inside the postprocessor, where the optimization takes place. The spindle speed values *n* are being overwritten directly

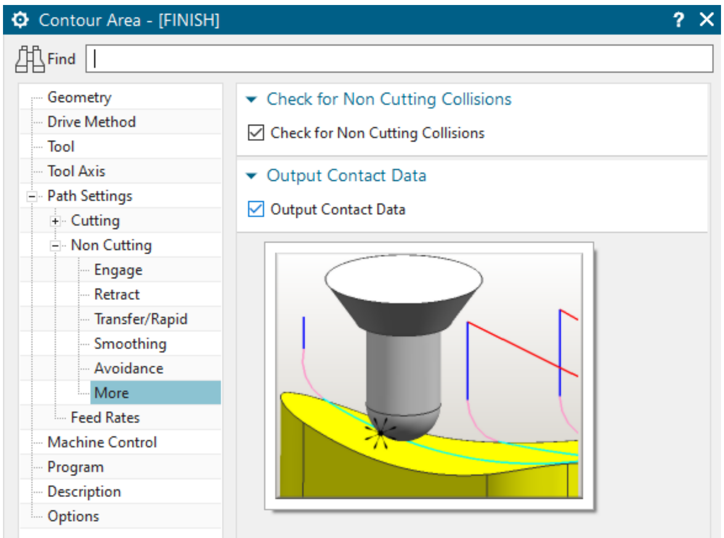


Figure 22 - Output Contact Data settings in NX

during postprocessing based on the effective cutting diameter calculated using the equation (1), where v_c is the programmed cutting speed and d the tool diameter:

$$n = \frac{v_c \cdot 1000}{\pi \cdot d} \; [rpm] \quad (1)$$

In order to keep the same cutting conditions and tool load, the feed rate f_n is also recalculated using the equation (2),

$$f_n = f_z \cdot z \cdot n \; [mm/min] \quad (2)$$

where f_z is the feed per tooth and z the number of teeth. (Stejskal, 2017)

This model works purely on the computer side and produces spindle speed and feed rate values, which might not be achievable directly during milling on the machine tool because the spindle and linear axes might not have the required dynamics, mainly acceleration. (Bartoš, 2019)

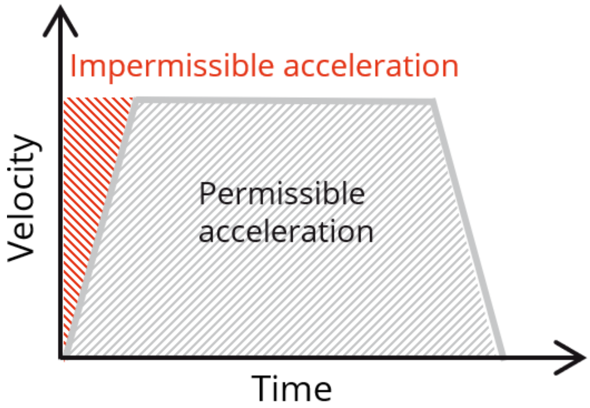


Figure 23 - Permissible and impermissible spindle acceleration values

As a follow up I have further enhanced the optimization algorithm with a spindle dynamics, where three milling machine tools with different spindles have been measured and the course of the acceleration and deceleration has been observed. (Bartoš, 2019)

The characteristic was the same for all three machine tools and can be seen in Fig. 23. However, the acceleration value has differed significantly from machine to machine. The higher the acceleration value, the higher spindle speeds can be achieved in shorter period as can be seen in the Fig. 23.

Given this fact it was necessary to include this calculation in the optimization algorithm. As a first step the distance s on the Fig. 24 between the current and the next position must be calculated using equation 3. As a next step the time t required

$$s = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2 + (Z_B - Z_A)^2} \; [mm] \quad (3)$$

for the movement between point A and point B is calculated using the already optimized value for feed rate with the equation 4. The time value is then used to

$$t = \frac{s}{f_n} \; [min] \quad (4)$$

calculate the maximal achievable spindle speed n_{lim} using equation 5, which is used for acceleration and where a is the measured spindle acceleration and n_{prev} the previous spindle rpm. The same equation only with minus sign is used for deceleration.

$$n_{lim} = a \cdot t + n_{prev} \tag{5}$$

If the calculated spindle rpm is bigger than the n_{lim}, the limit value is set as the target rpm.

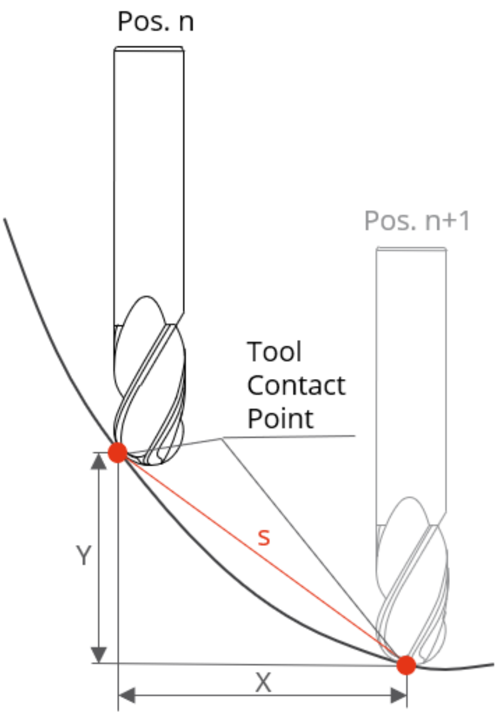


Figure 24 - Limit value for spindle rpm change calculation principle

Furthermore, the time spent above the nominal spindle speed is being monitored and user can set a maximum time value in minutes after which the maximal spindle rpm is set as the nominal value to keep the spindle properly cooled as referred to in the spindle manufacturer's manual.

To make all those settings user-friendly a special control window has been programmed directly in Siemens NX (Fig. 25), which sends all specified values to the postprocessor, so the user does not have to go directly into the postprocessor to make custom changes.

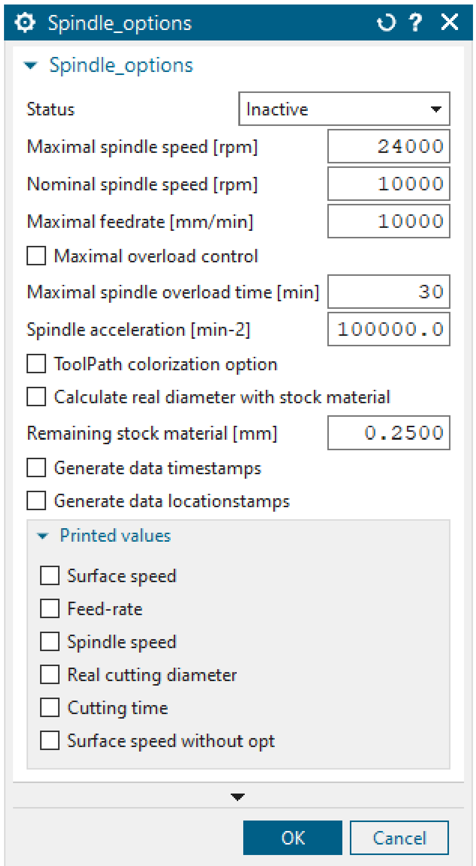


Figure 25 - Cutting Speed Optimization control window

Feed rate Optimization

Górecki developed a computational algorithm, which aims to compensate feed rate value during machining. Nowadays, almost all the manufacturing tasks are done using CAM software where the machining parameters such as Spindle Speed, Feed per Tooth etc. are specified. However, it can happen that during machining those predefined values may not be the same as we defined e.g., previous chapter about Cutting Speed Optimization.

The same phenomenon happens with feed rates as well. To be exact, there is a distortion during circular moves either in 2D or 3D machining because of the radius difference between part surface and tool path as shown in the Fig. 26 (the ratio between r_2 and r_1). Given the fact that the programmed feed rate is on the tool path and not on the part surface there will be different cutting conditions, which could lead to worse surface quality and sometimes even tool breakage. (Górecki, 2020)

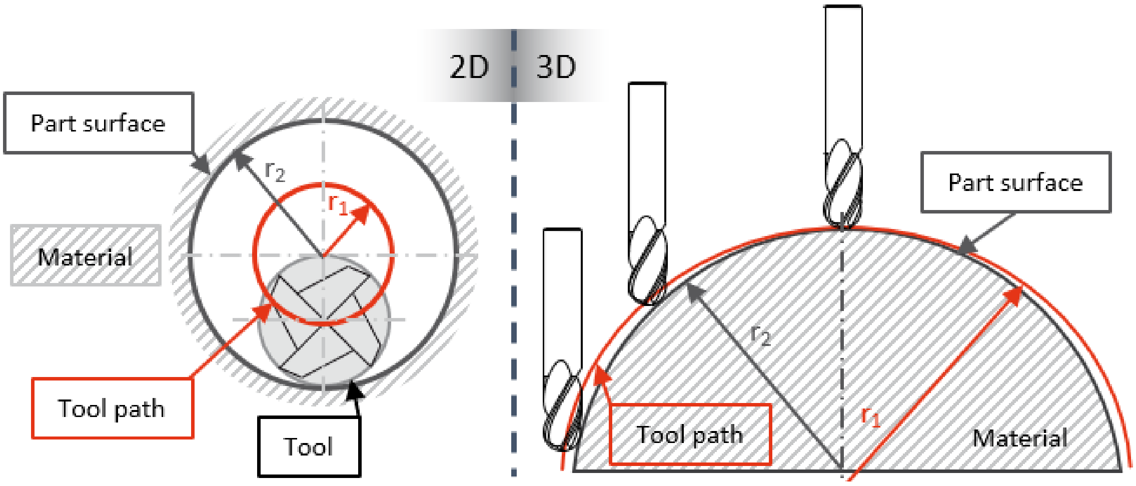


Figure 26 - Feed rate values inaccuracy during circular moves

While machining features in 2D, the toolpath is usually generated in the form of circular interpolation (G2/G3 command) and the directly programmed circle radius can be used. However, during 3D machining, the toolpath is mostly generated in the form of linear interpolation (G1 command) and thus there is no radius that could be used for calculations as can be seen in the Fig. 27.

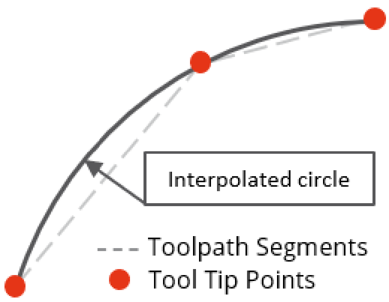


Figure 27 - Tool path as a linear motion during 3D interpolations

Because of this, the algorithm calculates a circle radius from three consecutive tool tip points, which are defined by the tool path as shown in the Fig. 27. Later, the computed circle radius is compared to the distance between surface contact point and the ratio between the two values is calculated. The radius ratio tells us whether the part surface is convex or concave (Fig. 28) and according to this the programmed feed rate f_{opt} is either lowered or increased using the equation (6),

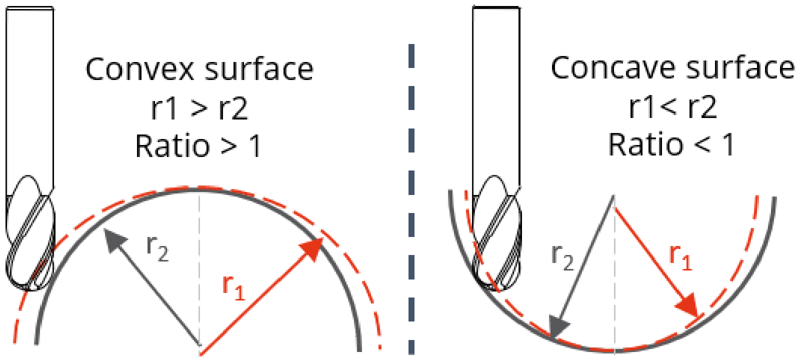


Figure 28 - Convex and Concave radius cases

$$f_{opt} = R_{ratio} \cdot f_n \; [mm/min] \tag{6}$$

where R_{ratio} is calculated according to the equation 7.

$$R_{ratio} = \frac{r_1}{r_2} \; [-] \tag{7}$$

4 Machine Tool Properties Influencing the cutting conditions

A production machine tool is a complex mechatronic system, which must comply with different load conditions and still perform within very strict tolerances, which are sometimes in the micrometer range. Given this fact, all the mechanical components, electrical parts, controls, software, but also loads and environment have an influence on the final performance of the machine tool. In the following chapters, the different aspects of the machine construction and their influence on the process optimizations are discussed.

4.1 Spindle Construction and Drive Mechanism

The spindle is a main part of a machine tool, which rotates around its axis and thus produce the main cutting motion. On a lathe, the spindle takes care of the rotation of the workpiece and on a milling machine, boring machine or surface grinder of the rotation of the tool. The rotational movement must be precise in order to guarantee the tolerances of the final workpiece.

Regarding the Cutting Speed Optimization described in the previous chapter, the spindle must also be very dynamical and should be able to produce high rpm, in order to fully utilize the optimization potential. Given this fact, only some of the spindle designs are suitable.

Gear-Driven Spindle

These spindles use gears or a transmission (Fig. 29), to transmit the torque from the motor onto the spindle. The advantages of this drive type are high transmissible torques, which are required during roughing operations, but also on heavy-duty machine tools. Using transmission, the relatively high rpm spectrum can also be achieved. The main disadvantages are the weight of the system, lower maximal spindle speeds and noise. This spindle drive is used in general applications. (Nohál, 2008)

Figure 29 - Gear driven spindle construction (Colonial Tool Group, 2016)

Belt-Driven Spindle

The next drive option is using a belt or a toothed belt (Fig. 30), which is in comparison to a gear drive lighter, but still capable of transmitting high powers. Flat belt drives are mostly used in surface grinders, because they do not produce vibrations and do not transfer the vibrations from the drive onto the spindle. Regular belt driven spindles are used very often on lathes. (Nohál, 2008)

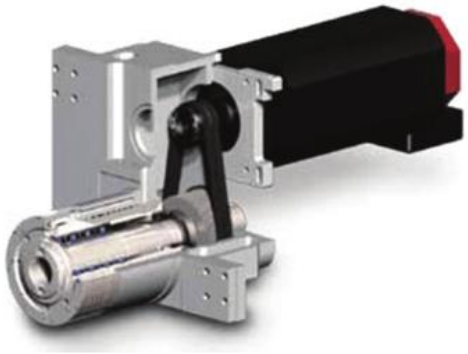


Figure 30 - Belt driven spindle construction (Goman, 2020)

Direct Drive Spindle

In the field of high-speed machining, the use of the direct drive spindles is the most common, given the fact that the system is very dynamically stable. The servomotor is connected together with the spindle shaft to create the so-called electric spindle. However, there are also different construction options. (Marek, 2014)

The electric spindle is made of the rotor, which is pressed on the spindle. In the outer tube, there is the stator with either the air- or water-cooling system. This configuration is capable of a dynamical rpm change. The construction example is shown on the Fig. 31. (Nohál, 2008)

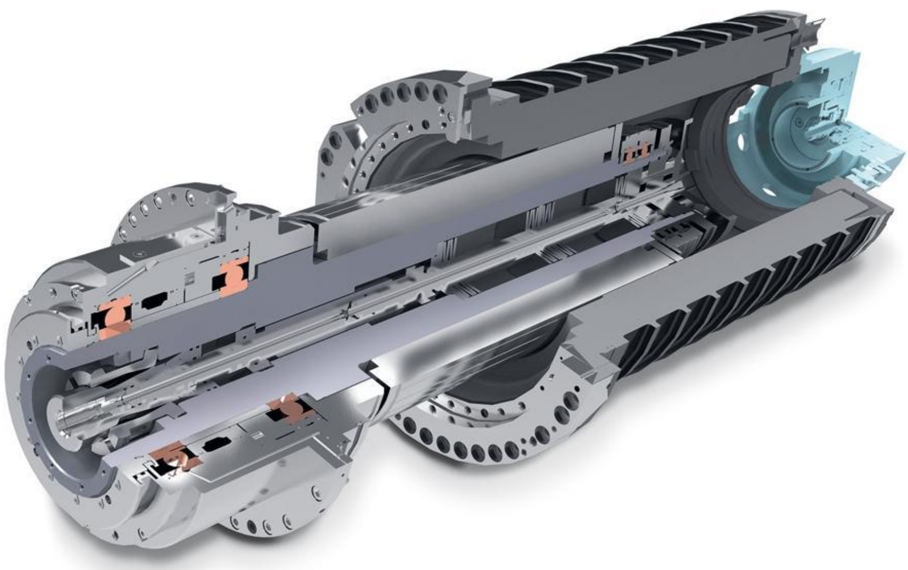


Figure 31 - Electric spindle construction (Zerspanungstechnik, 2018)

Spindle Bearings and Interface Type

Other parameters that have a direct influence on the performance of the spindle and thus on the suitability for the Cutting Speed Optimization are spindle bearings and spindle interface type.

Every kind of a bearing construction, such as ball, cylinder, angular contact ball bearing has a different permissible maximal rpm and thus specifies the maximal spindle speed. The spindle interfaces must go with the maximal rpm hand in hand, because they have a specified maximal rpm as well. The overview of the spindle interfaces is shown in the Table 1.

Table 1 - Permissible spindle speeds for different spindle interface types

Interface type	Spindle speed [rpm]	Label
ISO (SK)	10 000	ISO 40
HSK	24 000	HSK 63
Coromant Capto	11 000	C6
BIG Plus	11 000	BIG Plus 50

Spindle Control System

The spindle control system has got in the case of a direct drive spindle a velocity and a current feedback loop. This control system is usually simpler than a control system for a motion axis as its only goal is to reach the commanded spindle speed as fast as possible even with slight overshooting of the target value. Given this fact, the jerk value for the velocity feedback loop is set as an unlimited value. This has a direct influence on the spindle velocity characteristics as acceleration is an integration of the jerk (acceleration is a constant value when jerk is unlimited) and velocity is an integration of the acceleration, which is then a linear function. (Brecher, 2021)

4.2 Motion Axes Construction

Every machine type has got different parameters in terms of the maximal axes travel distance, permissible loads, maximal workpiece dimensions, maximal feeds etc.
The maximal axes travel distance directly influences the maximal workpiece dimensions and indirectly influences the used components for the motion axes, which could for example be direct drives or mechanical drives, both of which has got different dynamics, permissible loads etc.

Direct Drive

The direct drive consists of a linear motor, which works on the same principle as a servomotor with the difference, that the permanent magnets do not rotate. They are instead fixed along the motion axis travels which can either be a linear or a rotational axis. The construction of such a direct drive as an example of use in the case of a linear axis is shown in the Fig. 32. These drives excel in terms of dynamics in comparison to its mechanical counterpart. With the acceleration and velocity being up to 4x higher than with a mechanical drive. The Linear Motors are also better in terms of positional error-proneness as there are no mechanical components between the drive and the driven object (machine table). (Brecher, 2021)

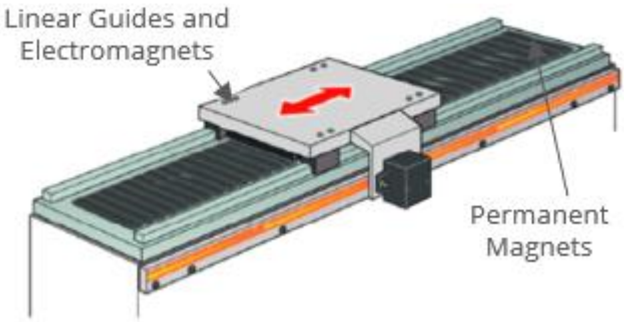


Figure 32 - Direct Drive construction (Brecher, 2021)

Mechanical Drive

The other option for motion axes construction is a mechanical drive. This widely used option in machine tool industry is cheaper and easier to build in comparison to the direct drives. It also does not produce that much heat, which could lead to positional errors. However, the number of mechanical components placed in between the Servomotor and the driven object must be taken into account as can be seen on the Fig. 33. The typical mechanical drives are Ball Screw, Rack and Pinion and Toothed Belt. (Marek, 2014)

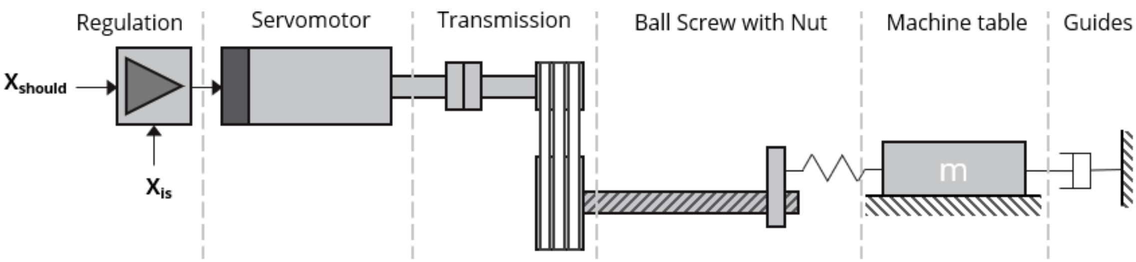


Figure 33 - Mechanical drive construction

4.3 Control System and its Settings

A control system is a group of components, which command or regulate another system. In machine tool industry there is in a vast majority of applications used Cascade Control System for motion axis with three feedback loops – Current, Velocity and Position as shown in the Fig. 34. This system is easy to design, fulfills all the demanding tasks and is relatively cheap. However, unique tuning of the

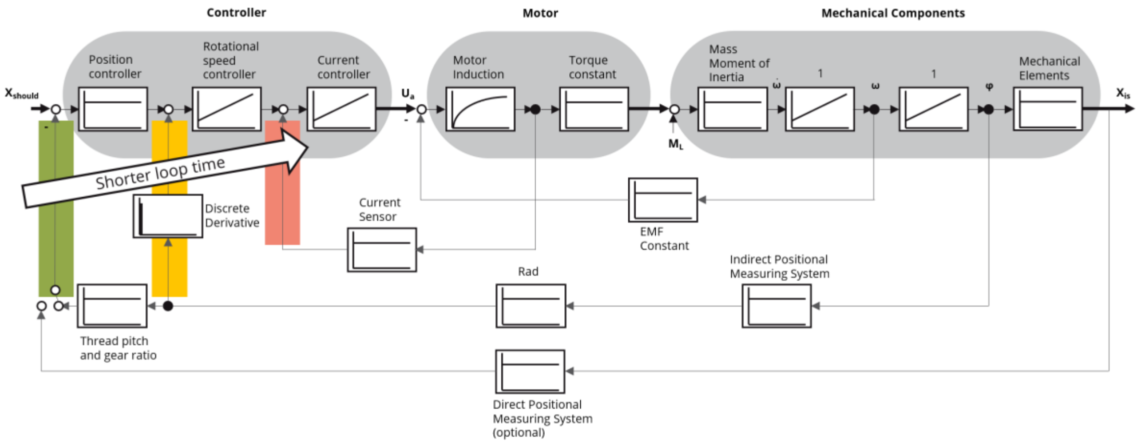


Figure 34 - Control System of a motion axis for a milling machine (Brecher, 2021)

control system parameters should be done for every machine tool as each one has got different travel distances, components weight and expected loads (light-duty machining, heavy-duty machining). Usually, this involves a compromise between processing time and position precision. The main influence on the optimization functions is the so-called Positional Error. According to the equation 8, the Positional error $\triangle s$ is directly influenced by the speed gain K_v but too high gain values could lead to overshooting of the target position (Fig. 35). (Brecher, 2021)

$$\triangle s = \frac{v_{is}}{K_v} \; [mm] \tag{8}$$

Where K_v is the speed gain and v_{is} the current velocity.

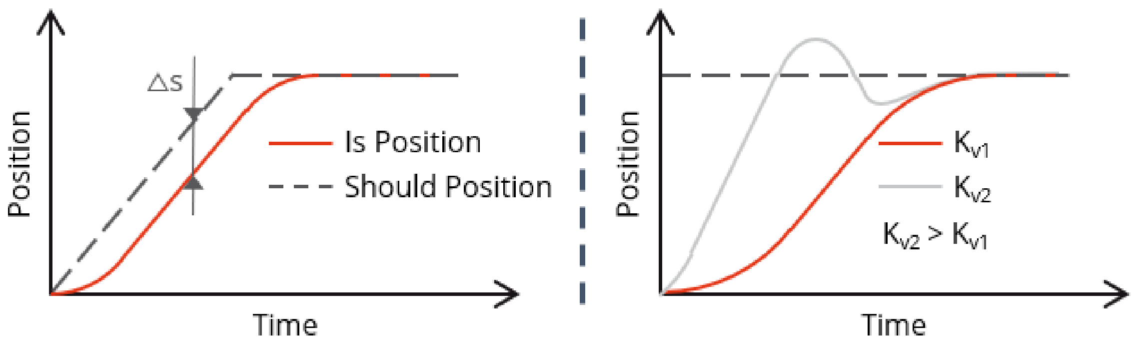


Figure 35 - Speed gain K_v and its influence on the positional error (Brecher, 2021)

4.4 Machine Tool Control

The control system is the brain of the machine tool and its main task is to control and monitor all the axes and sensors, based on the provided NC program. This process involves many steps, before the input is sent into the regulation and thus slightly altering the postprocessed NC code, in order to assure smooth and collision-free machining. These steps are discussed in the following paragraphs.

Tool Offsets

The first task is to ensure that every tool will move on the same programmed path despite length, diameter and shape differences as can be seen on Fig. 36. In order to be able to do so, the user must correctly fill in the Offsets Table directly on the control panel of the machine. This already has an influence on the optimization functions, because in the CAM software the used values for diameter are nominal (e.g., 16 mm) but the real dimension has certain tolerances and could be for example 15,98 mm. (Denkena, et al., 2011)

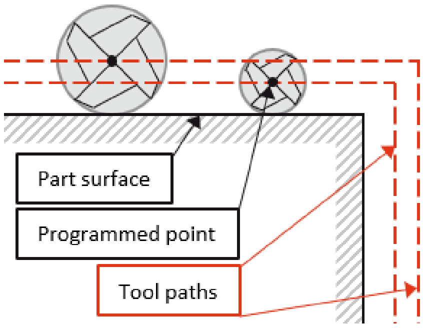


Figure 36 - Tool diameter influence on the programmed tool path

Kinematic Transformation

The machine tool knows only the position of one point (Machine Coordinate System Origin - MCS), which is specified by the manufacturer of the machine. This point is somewhere inside the machine travel boundaries, but from a practical perspective, the user does not work with this point, but rather with a user-defined one, also called User-Defined Coordinate System, or Workpiece Coordinate System - WCS. The distances for x, y, z axes from MCS to WCS are again specified in a table directly on the control panel. (Brecher, 2021)

The control system receives as an input the postprocessed NC code, which is in WCS. Furthermore, every machine can have different kinematical structure and different number of driven axes. Given this fact, the control must internally do a kinematic transformation from WCS to MCS to be able to correctly follow the programmed toolpath (Fig. 37). (Brecher, 2021)

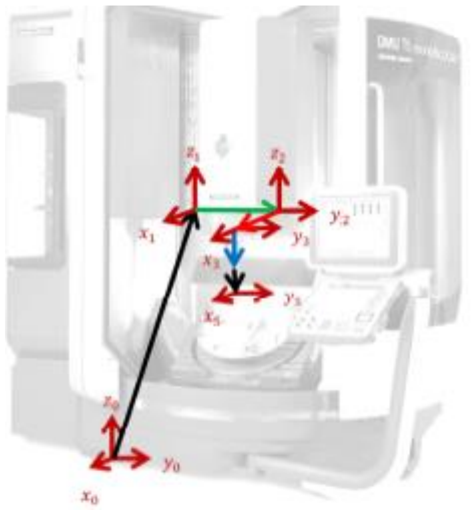


Figure 37 - Kinematic transformation for the linear axes of a 5-axis milling machine

Interpolation

The control system further interpolates the line or circle movement specified in the NC code with extra points in order to produce a smooth movement, because the motion is a combination of more than one axis. It could cause a slight deviation from a perfect circle during the circle interpolation as can be seen in the Fig. 38. In the case of the line interpolation, there will be points, which have not been in the NC code or in the CAM software and thus the cutting conditions could also vary slightly.

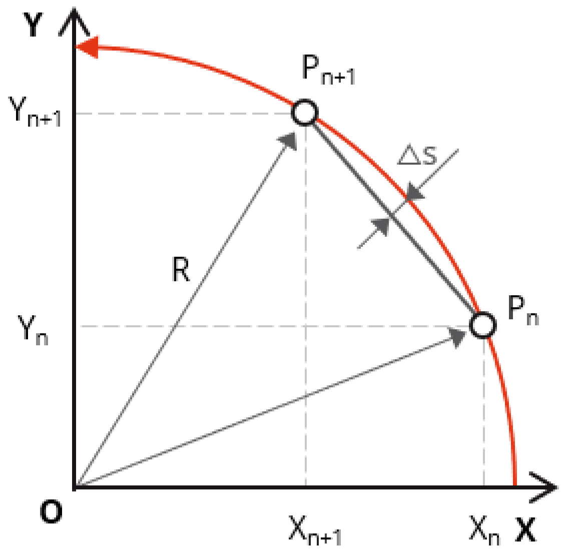


Figure 38 - Circular interpolation tool path deviation

Velocity Control

Another task of the control system is to assign every axis its velocity with consideration of the dynamical boundaries and minimal processing time, based on the programmed feed rate in the NC code. The dynamical boundaries are usually different for every axis, because of the servomotors and drive mechanism used and because of the different settings of the dynamical boundaries (Jerk values) as can be seen on the Fig. 39. (Brecher, 2021)

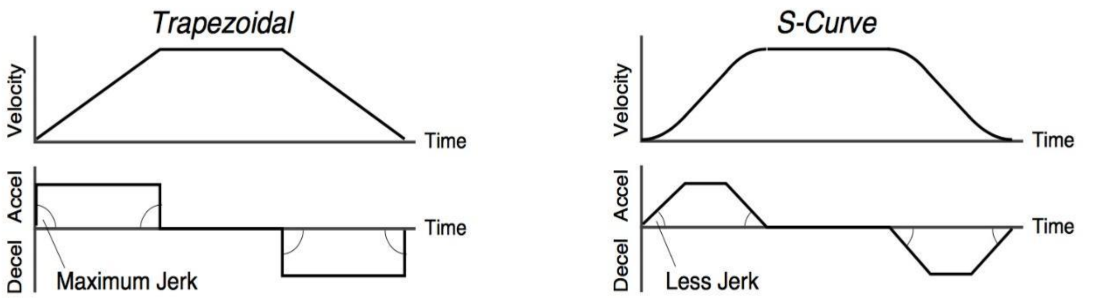


Figure 39 - Different Jerk settings for a velocity control curve (Collins, 2017)

Furthermore, there is an option to equip the control system with the function called Look-Ahead. This tool reads the next NC blocks (up to 10 000) and based on the future feed rate values adjusts current feed rate in order to produce as smooth motion without unnecessary accelerations and decelerations as possible (Fig. 40).

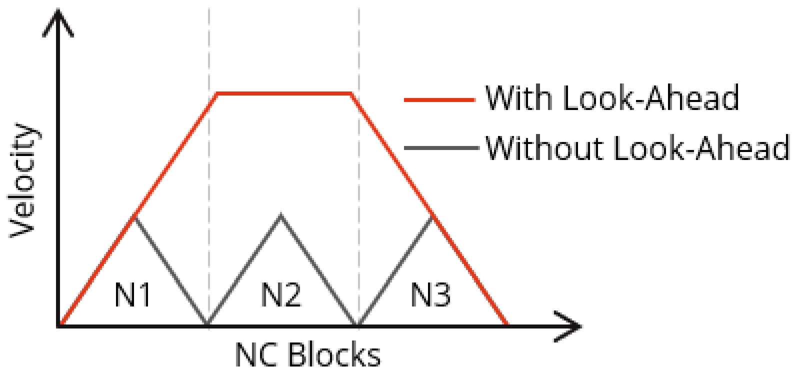


Figure 40 - Velocity curve with and without Look-Ahead

In addition to that, during multi-axis machining the programmed feed rate is not achieved on the tool tip point because the control system assigns to each motion axis different velocity so that the programmed endpoint is achieved by every axis simultaneously. There are optional functions that the customer can buy with the machine (for example TCPM for Heindenhain or Traori for Siemens controls), which take care of the kinematic transformations between rotary and linear axes automatically and the user can program only in x, y and z coordinates. This should also ensure the correct feed rate value on the tool tip point. However, as Vavruška described in his dissertation thesis, this is not the case. The control system tries to achieve the constant feed rate value on the tool tip point and because of that it could accelerate some axis above its programmed feed rate. (Vavruška, 2013)

Moreover, there are two modes in which the machine moves to programmed points.
The first one is so called Exact-Stop and the second one is Exact-Stop Window or Normal Cutting Mode. The programmed point is either achieved inside of a tolerance window (Fig. 41), which produces smooth moves, but does not achieve exactly the programmed position, or the point coordinates are achieved exactly and that could cause disrupted motion.

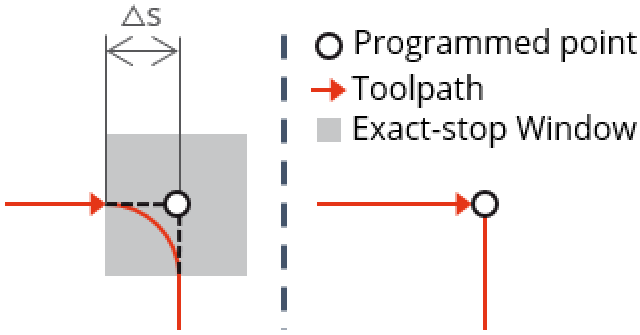


Figure 41 - Comparison of Exact-stop and Exact-stop window machine mode

5 Suitable Programming Languages

The standard language for the postprocessors in Siemens NX is the Tool Command Language – TCL. It is a lightweight language, which covers all the basic operations and tasks available directly in postprocessor. However, in order to be able to program and implement advanced mathematical operations and custom optimization functions, different programming language must be used. Given the fact that the custom optimization functions require some advanced mathematics, it is necessary to program them in a different language and then establish a connection between the postprocessing and the custom script, in order to be able to directly calculate all the optimized feeds and speeds in postprocessor and output them in the NC code. All of this should preferably happen without extra user interaction.

There are several ways how to connect running postprocessing operations with custom user scripts while using Siemens NX as a CAM software and NXOpen API as the connection between the postprocessing and custom user script.

In the next subchapters it is discussed how to establish the connection between postprocessor and the user script as well as what the advantages and disadvantages of each programming language are.

5.1 TCL with Direct Python Connection

This is the simplest method that can be used to establish the connection, which utilizes the TCL command 'exec' <file path to .py file> <passed variables> (Fig. 42). This way, variables are being passed to a specified python script, where it is possible to further work with them e.g., using command 'eval' (Fig. 43). Once the python script is processed, the variable '*mom_python_tcl_output*' will store the result in TCL. Python is nowadays a very progressive open-source language, where the user can download and use a vast number of libraries. Some of them are for example NumPy or SciPy, both of which are powerful mathematical libraries used for

```
global mom_python_tcl_output
set mom_python_tcl_output [exec python z:\\Pycharm_Projects\\NXOpen_37\\python_tcl.py $mom_mcs_goto(0)]
MOM_output_literal "MCS_GOTO: $mom_python_tcl_output"
```

Figure 42 - TCL code to call custom python script

advanced operations and calculations. However, the main disadvantage of Python is the execution time. An example python code for direct TCL connection can be seen in the Fig. 43.

```
import sys

def func1(printed_item):
    print("Called Function 1: " + str(printed_item))

def func2():
    print("Called Function 2")

if __name__ == '__main__':
    for indx, item in enumerate(sys.argv):
        if indx != 0: # 0 is file name
            eval('func1(item)') # must be passed as a string
```

Figure 43 - Python code for direct TCL connection

5.2 NXOpen Python

Python can be furthermore used as a programming language, while using NXOpen API along with 'MOM_run_user_function' command in TCL. In comparison to the TCL – Python direct connection this has the advantage that the user can access Siemens NX CAM variables which are directly available during postprocessing and use all the API methods provided by Siemens as can be seen in the Fig. 44. The main disadvantage is again the execution time compared to either below mentioned C – language options or to the direct TCL – Python connection.

```
import NXOpen
import NXOpen.UF

def main(*args):
    theUFSession = NXOpen.UF.UFSession.GetUFSession()
    theMomObject = theUFSession.Mom
    theUiSession = theUFSession.Ui

    mom_id = theUFSession.Mom.AskMom(theMomObject)
    testing_string = theUFSession.Mom.AskString(mom_id, "string")
    try:
        # Write string from postprocessor to Listing window
        theUiSession.OpenListingWindow()
        theUiSession.WriteListingWindow(testing_string)
    except NXOpen.NXException as e:
        # Exception handling code
        pass


if __name__ == '__main__':
    main(None)
```

Figure 44 - Example of a NXOpen Python code

5.3 NXOpen C#

A next option is the C# programming language, which offers very similar code structure as its python counterpart, while having faster code execution and overall better support from Siemens NX team. There are also templates for Visual Studio directly provided with Siemens NX installation, which help the user to accelerate the development process significantly. The C# code can be exported as a .dll file, which is a very convenient option for external implementation into another application.

5.4 NXOpen C++

The fastest option regarding code execution is C++ programming language. While there are templates prepared for Visual Studio within the NX installation folder in the same manner as for C#, the complexity of the language is generally not suitable for small scripting applications and the speed benefit in comparison to C# is only very small. The C++ code can also be exported as a .dll file like C# providing the same convenience. There is also an indirect option how to write C++ code with the use of Matlab software. The code written in Matlab can be exported as a C++ code with a built-in script directly in Matlab. However, not every custom method in Matlab can be exported into C++, which makes this option very inconvenient.

The C++ and C# languages may be up to 10x faster than Python language, which makes them particularly suitable for advanced mathematical calculations as the computational time is drastically reduced and thus the postprocessing itself as well.

6 State of the Art Summary

There are already established and well tested optimization principles for roughing and finishing milling operations both on the CAM side of the workflow such as Constant Tool Engagement, Maximal Cutting Length and Dynamical Feed Rate Control as well as on the Control system side, such as Optimized Contour Milling. Both categories are commercially available and, in most cases, seamlessly implementable into an already established workflow of a company. However, none of those principles work with a continuous spindle speed variation or a feed rate compensation during 3D interpolation milling operations.

Nevertheless, a development has been established in the last years described in the research papers regarding the topic of a continuous spindle speed variation as listed in the chapter 3.1 - Scientific Publications. The authors of (Käsemodel, et al., 2020) article were able to program and indeed continuously change the spindle rpm but lacked the consideration of the spindle dynamics. They also mentioned that such a frequent spindle rpm change does not have a negative effect on the spindle lifetime. However, there are no papers regarding the feed rate compensation because of the changing angular velocity during 3D interpolation milling operations.

Furthermore, there has been proven a negative effect on a surface quality and tool life during milling with a tip of a ball end mill as described in (Stejskal, et al., 2020).

Moreover, there are plenty of factors that have a certain degree of influence on the performance and reliability of the optimization functions as described in the chapter 4 - Machine Tool Properties Influencing the cutting conditions and before using such an optimization, the user should be aware of all the factors to achieve the best results.

Among the programming languages, which are usable for Siemens NX connection with the postprocessor, the C# and C++ options are the most suitable for such optimization functions because of the short execution time, while having all the mathematical tools that are necessary.

7 Custom Solution Proposals

According to the state of the art there are still no viable solutions for feed rate compensations during 3D interpolated milling operations except the (Górecki, 2020) thesis, which this thesis builds on and the goal is to further improve the calculation precision and reliability of the optimization.

The Cutting Speed Optimization with continuous spindle speed variation during milling operations proved to be the right direction to pursue as there are already a few papers that have been recently published focusing on this topic. Although the computational principle is the same as the one in (Stejskal, 2017) and (Bartoš, 2019), it still does not include all the critical factors that influence the calculation of required spindle speed. A next option how to calculate the effective cutting diameter of the tool, which considers the remaining stock material on the part surface will be added. This option further enhances the computational precision and give the opportunity to the user to make the choice between computational method in relation to the specific machined part.

The last step is to join the Cutting Speed Optimization and Feed Rate Optimization together, to produce a complete solution for the compensation of the phenomena that occur during milling of a complex parts.

7.1 Feed Rate Optimization

This optimization is currently based on an analytical calculation of a circle center and radius from the tool tip points that are generated by the CAM software and the circle radius ($r_{tcp1,2}$) is then compared to the distance between computed circle center and a contact point between the tool and the workpiece ($r_{contact1,2}$) as can be seen in the Fig. 45. This method has many drawbacks in terms of the computational accuracy of the compensated feed rate. One of the drawbacks is the neglection of the contact point position. If the contact point does not lie on the same

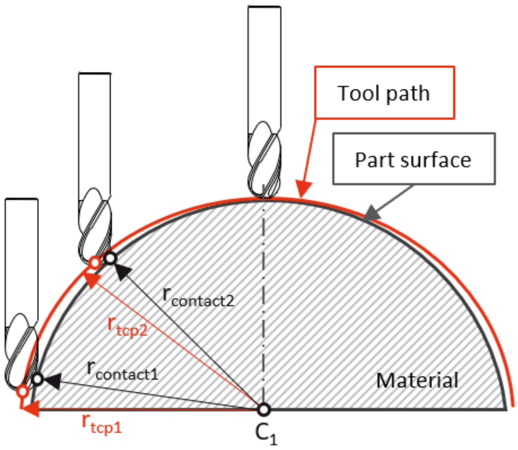


Figure 45 - Current feed rate compensation calculation

plane as the computed circle, the distance from the circle center to the contact point is distorted as depicted in the Fig. 46. Another drawback is the analytical circle calculation based on the three points. The tool tip and contact points are being generated with a certain tolerance from the ideal part surface and thus are heavily influencing the calculated circle center and radius if only three points are used for the calculation. The proposed solutions to those issues are discussed in the following subchapters.

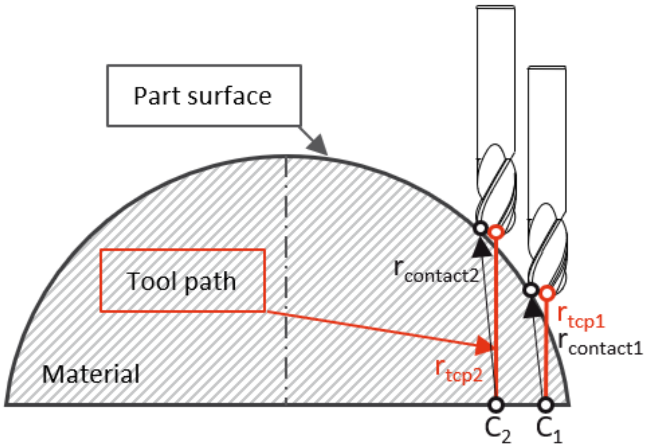


Figure 46 - Distance from contact point to calculated circle center distortion

7.1.1 Three Points Circle Calculation with Projected Contact Point

The first proposal is to take the existing algorithm as it is and correct the distance distortion between calculated circle center and contact point by projecting the contact point to the same plane as the tool tip points as depicted in the Fig. 47. This solution would be easy to implement and would improve the accuracy of the calculated compensated feed rate, however, the input in form of the three points would still pose difficulties in producing a smooth feed rate curve because of the toolpath tolerance specifically set for every operation, which has a significant influence on the precision of the generated points as can be seen in Fig. 48.

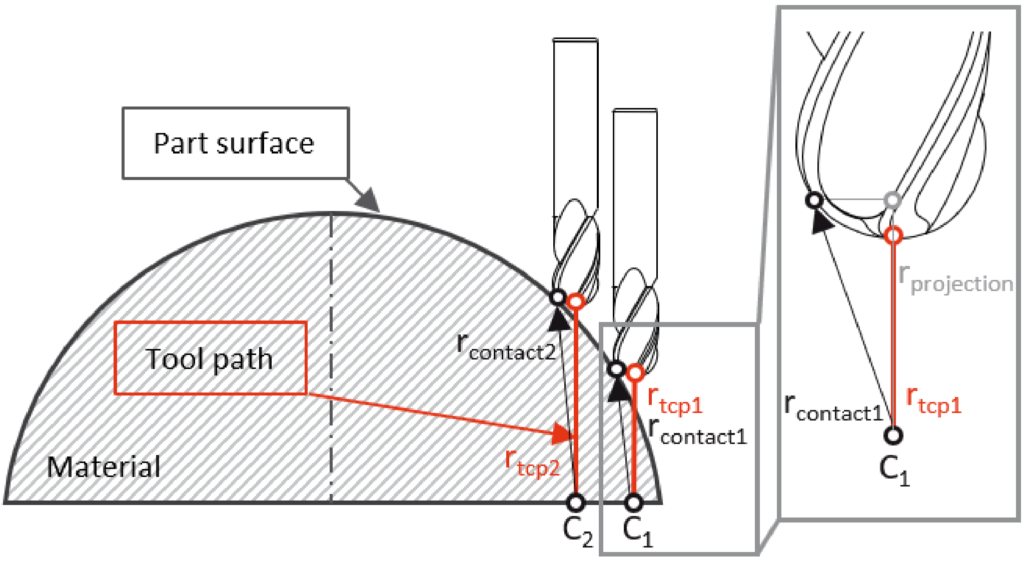


Figure 47 - Contact point projection to the plane of the tool tip points

7.1.2 Three Points Circle Calculation with Projected Contact Point and Feed Rate Values Smoothing

This option would work on the same principle as the previous one with one extra step to smoothen out the outputted feed rate values to counter the inaccuracies which are caused by the three-point circle calculation. The implementation would have to be a two-step postprocessor in order to have all the optimized feed rate values at disposal. The smoothening could be done for example using the Simple Moving Average algorithm or similar Exponential Moving Average and Weighted Moving Average, where the time period would have to be empirically evaluated to produce a satisfactory results. This option would be rather hard to implement and would not solve the inaccuracy problems

7.1.3 Best Fit Circle Calculation with Projected Contact Point

The next proposed solution would reduce the influence of the positional error of the generated points from CAM as depicted in the Fig. 48 by adding more points to the circle calculation algorithm. The algorithm would have to be fundamentally changed in this case, because a circle can be exactly calculated using only three points, thus the resulting circle would be a best fit approximate to the provided set of points. The best fit would be calculated using the Least-Squares algorithm, which is a good compromise between computational efficiency and accuracy. The output from the algorithm would be again the circle center coordinates and its radius, which could be compared to the distance between the projected contact point to the tool tip points plane and the circle center.

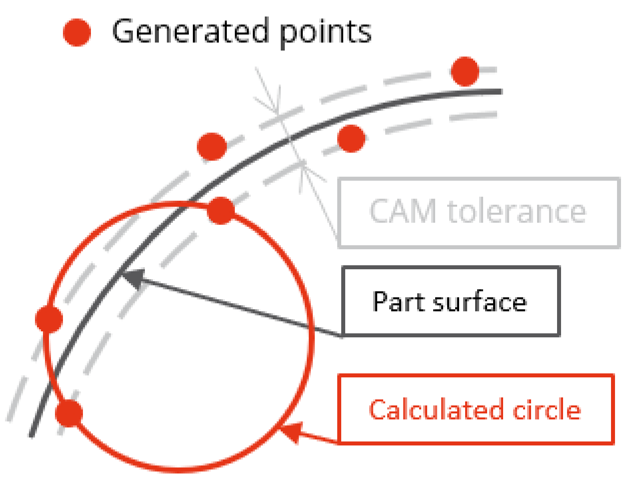


Figure 48 - Computational error in case of the three-points circle calculation

7.1.4 Best Fit Spline Interpolation with Projected Contact Point

The last proposed solution is an approximative solution as well but this time instead of trying to fit a circle through the set of points the algorithm would try to fit a spline. As the geometrical shapes and surfaces are also internally represented by a spline with a given number of control points in CAD and CAM software this should exactly reconstruct the part surface. However, we would need to implement additional logic for the toolpath reconstruction as it also consists of a non-cutting and repositioning movements, which would have a negative effect on the accuracy of the spline curve. The algorithm could not also reconstruct exactly the part surface because of the positional error that the generated points from the CAM have as seen in the Fig. 48.

7.2 Selection of the Custom Solution

The above presented solutions were compared based on the following criteria: Computational Accuracy, Computational Efficiency, Implementation Difficulty, User Friendliness and Scalability. The ranking scale is from zero to three, with three being the best and zero the worst. The awarded points are then summed and the proposed solution with the highest number of points in total is the selected solution, which is the Best Fit Circle with Projected Contact Point as can be seen in the Table 2.

Table 2 - Custom solution selection

Criteria / Proposal	Computational Accuracy	Computational Efficiency	Implementation Difficulty	User Friendliness	Scalability	Ranking
Three-Point Circle Calculation	0	2	3	2	1	**3.** ∑=8
Three-Point Circle Calculation with Feed Rate Postprocessing	1	0	1	1	1	**4.** ∑=4
Best Fit Circle with Projected CP	2	3	1	3	3	**1.** ∑=12
Best Fit Spline with Projected CP	3	2	0	2	3	**2.** ∑=10

8 Existing Algorithms Enhancement

This chapter presents the development and ideas behind the realization of the selected solution for the Feed Rate Optimization and shows the optional effective diameter calculation, which will be added to the Cutting Speed Optimization.
Both optimization algorithms are then joined together into one package to be easily distributable and to give the user the opportunity to choose what exactly needs to be optimized.

As discussed in the Chapter 3.2 - Optimization of cutting conditions by CTU in detail, the following modifications or improvements will be done based on the already existing solutions to the Cutting Speed and Feed Rate Optimizations created at the Department of Production Machines and Equipment and described in (Stejskal, 2017), (Bartoš, 2019) and (Górecki, 2020), where the benefits of the optimizations are well documented and compared to conventional milling strategies along with time studies.

8.1 Software Specification

In order to design the testing parts and create the tool paths, the CAD/CAM software Siemens NX – 1926 version was used. For the adjustments of the postprocessor, the software Postbuilder was used, which is also being developed by Siemens and thus ensures smooth communication and connection with the generated Cutter Location Data by NX. Programming of the algorithms enhancements was done in the Visual Studio 2019 developed by Microsoft. The following subchapters describe the used software in more detail.

Siemens NX

The reason why Siemens NX was chosen as the CAD/CAM software is the already existing algorithms, which were build and programmed on the Siemens NX interface. The main used modules in NX are CAD, where the user can parametrically design parts and assemblies and the CAM module, where the tool paths creation and NC code generation take place.

The CAM module can be used for several manufacturing technologies as can be seen in the Fig. 49. Every category focuses on different technology (e.g. turning, 3-axis milling, 5-axis milling) and thus contains different strategies for tool path generation. For this work, the categories Mill_Contour and Mill_Multi-Axis are the most important ones as the existing algorithms are used for milling operations.

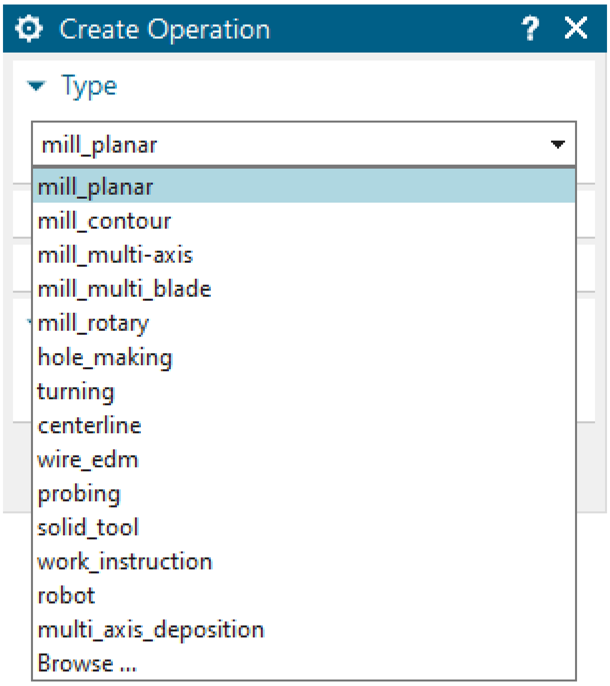


Figure 49 - NX CAM manufacturing technologies overview

While creating the tool paths (in our case for milling operations), the user has to follow the instructions displayed for every milling strategy and select required surfaces, parameters, used tool and cutting conditions as can be seen in the Fig. 50, to achieve the best possible results.

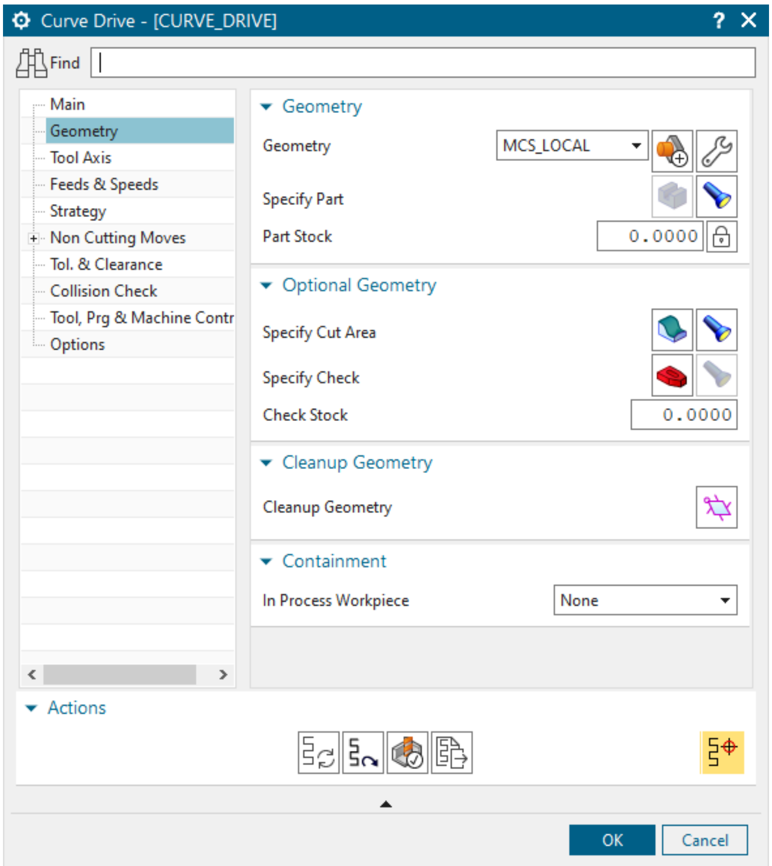


Figure 50 – Milling strategy parameters

If the user wants to verify the generated tool paths, there are two options available. The first one being simple display of the used tool and generated tool path (Fig. 51). The second one is a machining verification with a material removal, which can also show the possible collisions between the tool and the machined part or clamping objects.

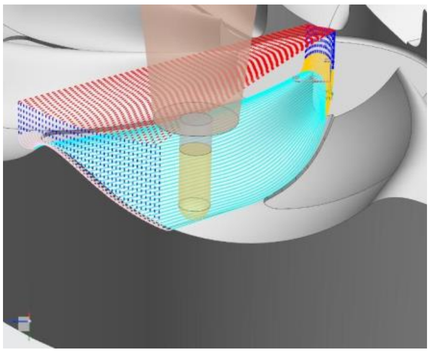


Figure 51 - Verification example of a generated tool path

The next step in the workflow after tool paths creation is postprocessing. NX sends the so-called Cutter Location Data – CL Data with various information to the postprocessor, which translates those data into a NC code. Given the fact, that the postprocessor reads these information during execution time, the user can create his own UDE (User Defined Event) to communicate and send certain information or variables to the

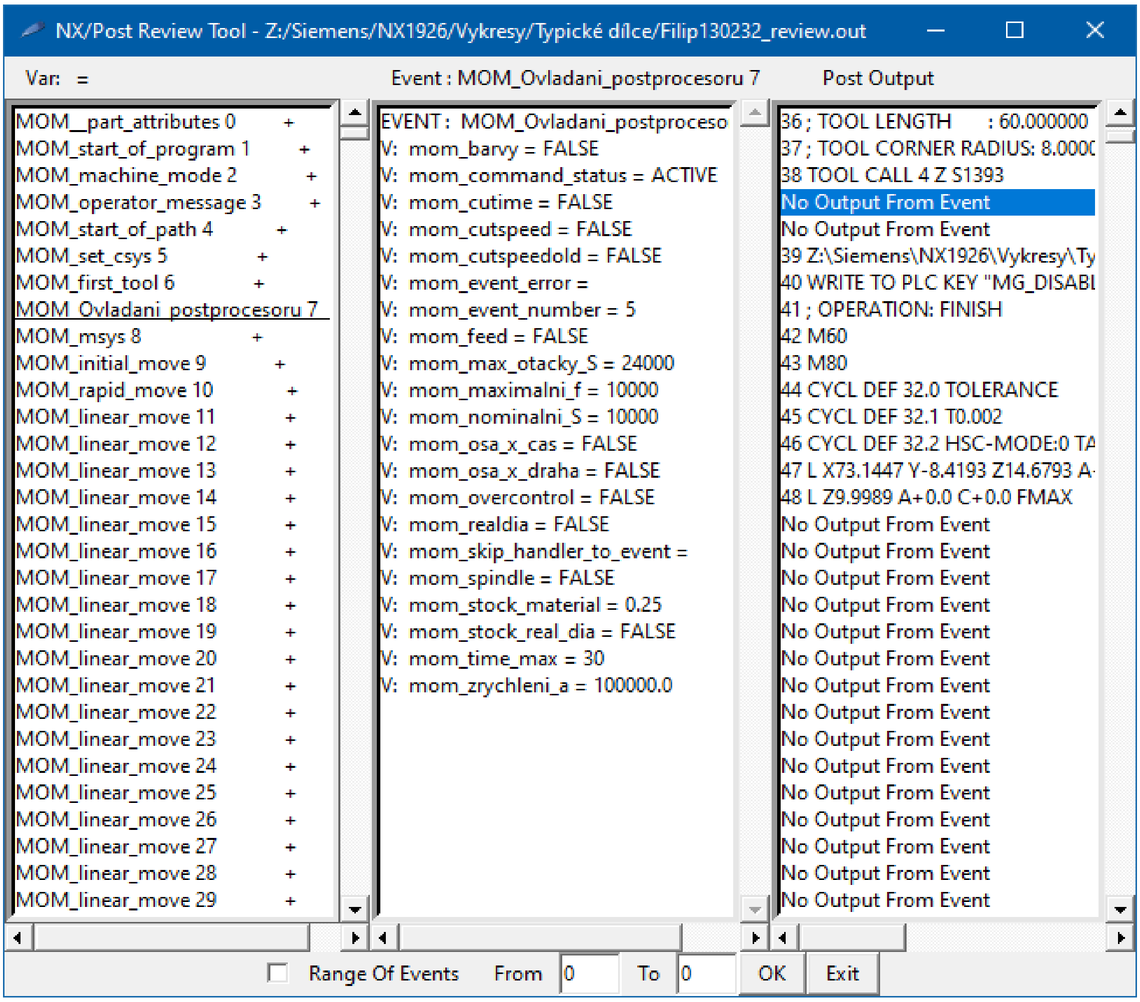


Figure 52 - CL data generated by NX with custom user-data

postprocessor. An example of such an UDE can be seen in the Fig. 52, where the middle column are the custom data, which were sent using an user-created UDE in NX directly to the postprocessor and are further used in the optimization algorithms.

Postbuilder

The Postbuilder software allows creation of a new postprocessor based on the provided templates or modification of an already existing postprocessor. The postprocessor is used as a translator of the CAM data into a CNC machine-friendly code as every machine tool and CAM are different and require or generate data in a different format.

When a postprocessor is opened in Postbuilder, there are five tabs located in the top bar. The most important one for this work is the Program & Tool Path tab, where the postprocessing sequence is displayed. The category tree can be found in the light-yellow area on the left side in the Fig. 53, where under Tool Path – Motion the optimization algorithms are being called with every iteration of the postprocessor as it goes through the whole tool path points generated by the CAM software.

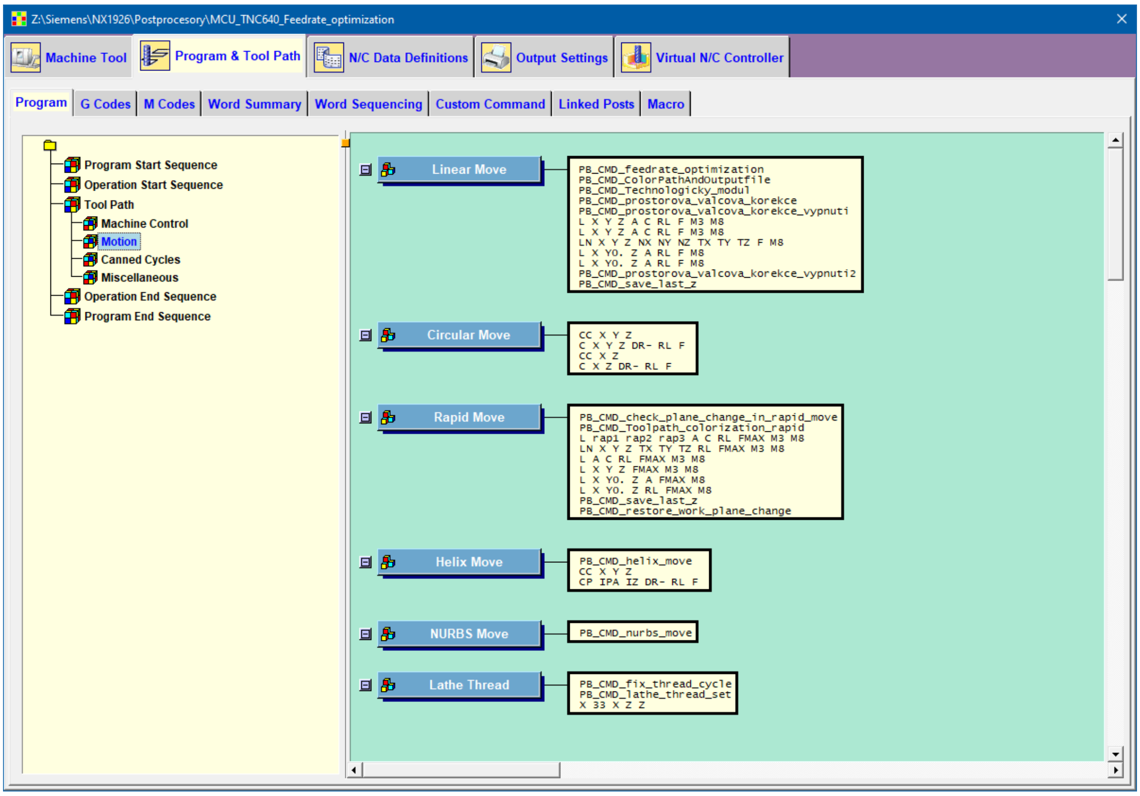


Figure 53 - User Interface of the Postbuilder software

The category Motion is then further divided into specific motion types. For the purpose of the optimization functions, the most suitable motion type is Linear and given this fact, the optimization blocks are located here (Fig. 54).

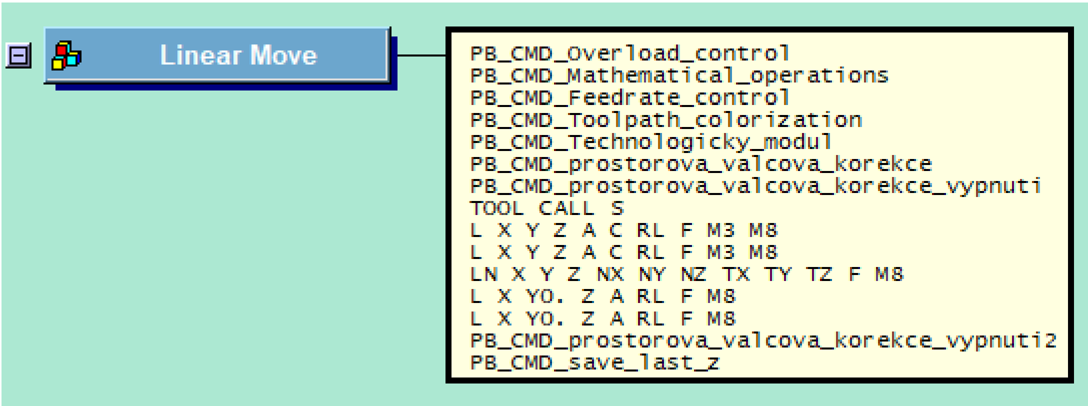


Figure 54 – Linear motion type in the Postbuilder software and its sub-modules

Visual Studio 2019

In order to effectively implement the optimization functions into a running postprocessing, the execution time is a key factor. Given this fact, the C++ and C# programming languages were selected as the most suitable ones. To ease the programming workload, already prepared templates from Siemens were used to create the skeleton of a future .dll file, which is called as a custom user function directly from the postprocessor. The template files are prepared for different distributions of the Visual Studio IDE from Microsoft based on the used Siemens NX version.

After importing the Template files from the Siemens NX installation folder into Visual Studio templates folder, there is an option to create a new NXOpen C# or C++ project as can be seen in the Fig. 55. The user then follows instructions given on the screen and a skeleton code is created.

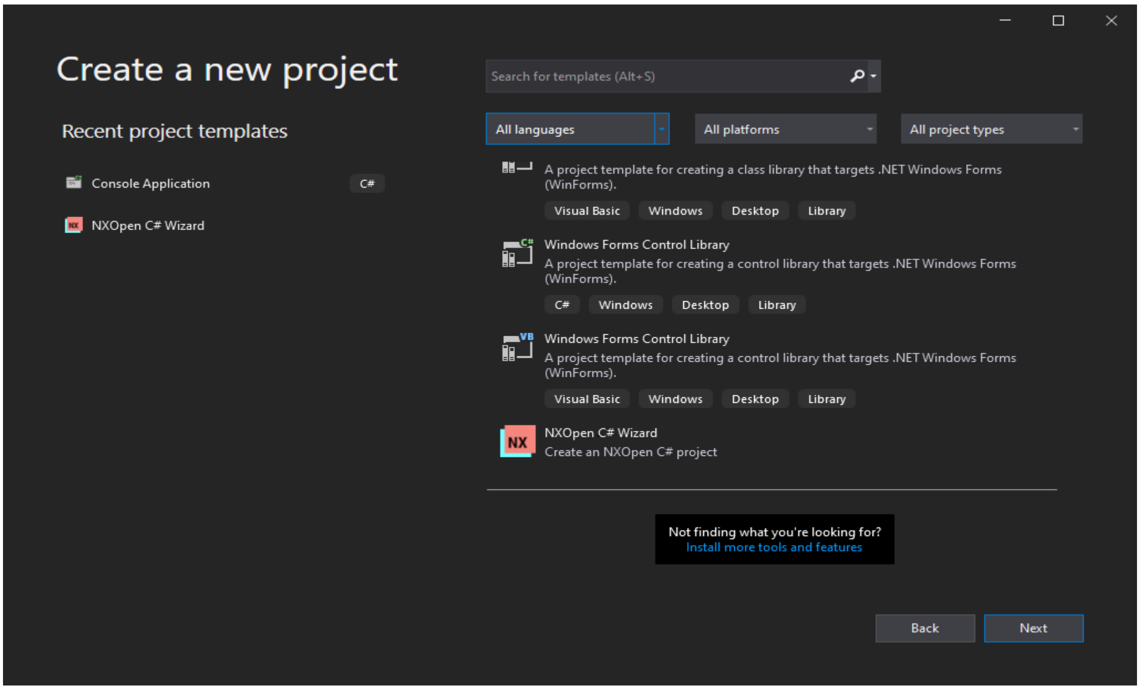


Figure 55 – Creation of a NXOpen project in Visual Studio 2019

8.2 Feed Rate Optimization Enhancements

To be able to program the best fit circle algorithm, it needs to be ensured that all the input points lie on the same plane as a circle is a 2D shape. However, the outputted points from the CAM software may not always lie on the same plane either because of the positional error caused by the toolpath tolerance as depicted in the Fig. 48, or simply because of the shape of the tool path / workpiece curvature. This needs to be solved in the first place before computing the best fit circle using best fit plane as a numerical approximation from the provided set of tool tip points. Once the equation of the best fit plane is found, the input set of tool tip points is projected into the best fit plane and using those points the best fit circle can be calculated. The best fit circle algorithm computes the circle center and its radius. The next step is to project one contact point into the best fit plane as well and compare the distance from the projected contact point to the circle center with the circle radius. The ratio between those two values is the compensation coefficient for the programmed feed rate value.

8.2.1 Best Fit Plane Algorithm

The goal is to find a plane that will lie as close as possible to a set of k 3-D points (P_1, …, P_k). The closeness is evaluated based on the square sum of the orthogonal distances between the plane and the points. Let the plane be described by a point C belonging to the plane and a unit vector $\vec{n}$, which is the normal to the plane. The orthogonal distance between a point P_i and the plane is then $(P_i - C)^T \cdot \vec{n}$. The plane can then be found by solving

$$\min_{C,||n||=1} \sum_{i=1}^{k} ((P_i - C)^T \cdot \vec{n})^2 \tag{9}$$

As a next step a 3 x k matrix can be written as

$$A = [P_1 - C, P_2 - C, \dots, P_k - C] \tag{10}$$

where C is according to the equation [9]

$$C = \frac{1}{k}\sum_{i=1}^{k} P_i \tag{11}$$

The unit plane normal vector $\vec{n}$ and plane base vectors $\vec{v}_{base1,2}$ are given by solving the singular decomposition A = USV^T, where U is a unitary 3x3 matrix. (Arun, et al., 1987)

$$\vec{v}_{base1} = U(:,1), \vec{v}_{base2} = U(:,2), n = U(:,3) \tag{12}$$

8.2.2 Points Projection to the Best Fit Plane

Once the plane equation is known, the input set of k 3-D points (P_1, …, P_k) must be projected into this plane. A directional vector $\vec{v}$ for each point P is calculated as

$$\vec{v}_i = P_i - C \tag{13}$$

The directional vector $\vec{v}$ is then transformed from 3-D into 2-D coordinates using a 2 x 3 transformation matrix T as

$$\vec{v}_{2D_i} = T\vec{v}_i \tag{14}$$

where T is

$$T = [\vec{v}_{base1}, \vec{v}_{base2}]^T \tag{15}$$

The transformed directional vector $\vec{v}_{2D}$ elements also represent the 2-D coordinates of the 3-D point in the best fit plane.

8.2.3 Best Fit Circle Algorithm

Given a finite set of points in $\mathbb{R}^2$, the algorithm tries to find a circle that best fits the points (in a least-squares sense). Given the general circle equation in $\mathbb{R}^2$ as

$$(x - x_0)^2 + (y - y_0)^2 = R^2 \quad (16)$$

it can also be written after a few mathematical operations as

$$2xx_0 + 2yy_0 + r_0 = x^2 + y^2 \quad (17)$$

where r_0 is

$$r_0 = R^2 - x_0^2 - y_0^2 \quad (18)$$

The equation [17] can also be written in a matrix form for multiple input points as

$$\begin{bmatrix} 2x_1 & 2y_1 & 1 \\ \vdots & \vdots & \vdots \\ 2x_n & 2y_n & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ r_0 \end{bmatrix} = \begin{bmatrix} x_1^2 + y_1^2 \\ \vdots \\ x_n^2 + y_n^2 \end{bmatrix} \quad (19)$$

The system [19] of linear equations $A\vec{x} = \vec{b}$ does not have an exact solution unless it is provided with exactly three points, which would be the exact solution. To find the approximate solution (in a least-squares sense), it tries to find $\vec{x}^*$ as

$$\min\|\vec{b} - A\vec{x}^*\| \quad (20)$$

where $\vec{x}^*$ can be calculated from

$$\mathrm{A}^{\mathrm{T}} A\vec{x}^* - A^T\vec{b} = 0 \quad (21)$$

The equation [21] can be for example solved by matrix inversion as

$$\vec{x}^* = (A^T A)^{-1}(A^T\vec{b}) \quad (22)$$

but to ensure a good computational reliability and computational efficiency the algorithm uses a Cholesky Decomposition to solve the system instead. The results of the linear system are the circle center coordinates x_0, y_0 and the value r_0 from which it then calculates the circle radius using the equation [18].

8.2.4 Implementation of the Feed Rate Compensation

The algorithms mentioned in the previous subchapters are programmed in C# and exported as a Dynamically Loaded Library (.dll), which is then loaded directly during postprocessing using the MOM_run_user_function <filename> command and called

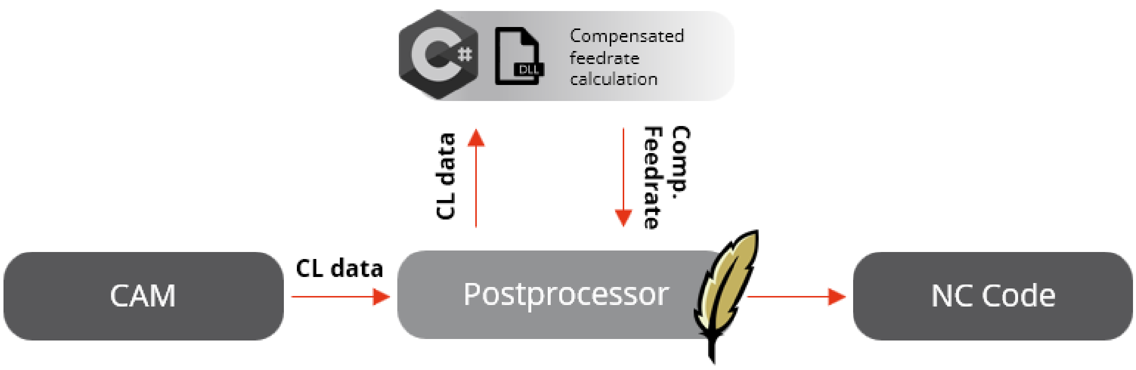


Figure 56 - Workflow of a postprocessing with implemented Feed Rate Optimization function

with every iteration of the postprocessor as depicted in the Fig. 56. The .dll file then returns the compensated feed rate, which is calculated as a ratio between the tool tip points circle radius and the distance from the circle center to the projected contact point (the contact point is projected into the plane by using the same algorithm as described in the chapter 8.2.2 - Points Projection to the Best Fit Plane but it is not used to compute the best fit circle). The algorithm logic is depicted in the Fig. 57. The equation for the compensated feed rate calculation is

$$f_{comp} = f_{orig} \cdot r_{ratio} \tag{23}$$

where r_{ratio} is calculated as

$$r_{ratio} = {}^{r_{tcp}}/_{r_{contact}} \tag{24}$$

where r_{tcp} is the circle radius computed by the best fit circle algorithm from provided tool tip points and $r_{contact}$ the distance to the contact point from the circle center. The whole logic for Feed Rate Algorithm is in the appendix 1.

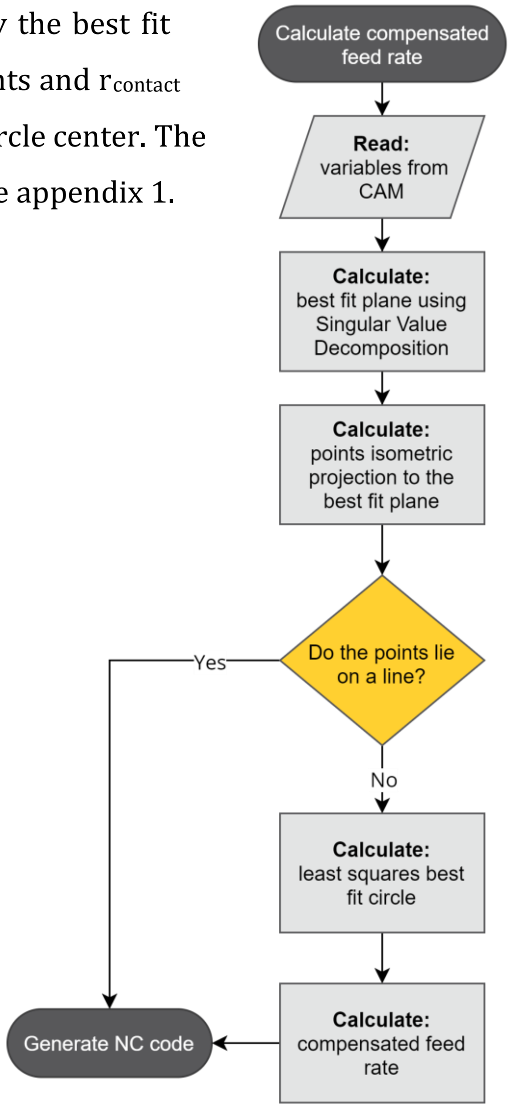


Figure 57 - Feed rate compensation algorithm flowchart

8.2.5 Comparison between the Analytical (old) and the Best Fit (new) Algorithm

As previously mentioned, the analytical circle calculation has its weakness in the computational accuracy because of the three input points as shown in the Fig. 48. A demonstrative geometry has been selected and it is shown in the Fig. 58 along with the feed rate values for the generated toolpath (cyan curve). The old algorithm (light gray curve) tends to oscillate around the correct values, whereas the new one has a smooth course (red curve).

The difference in values between the gray and red curve is caused by the computational inaccuracy of the analytical circle calculation (gray curve).

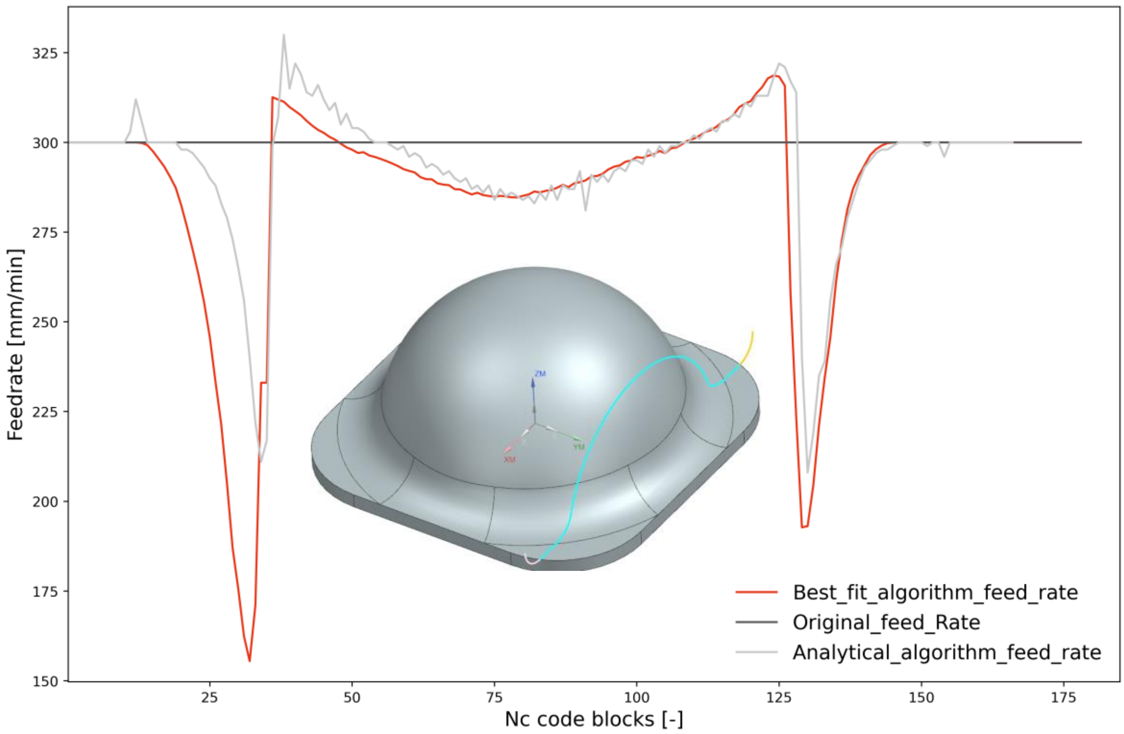


Figure 58 - Feedrate values comparison for the old and the new algorithm

8.3 Cutting Speed Optimization Improvements

The already existing solution principle to cutting speed optimization is described in the chapter 3.2 - Optimization of cutting conditions by CTU in detail and is based on the contact point. The publications from chapter 3.2 also describe how to enable the continuous change of the spindle rpm without linear axis stopping and what the benefits of using this optimization function are.

However, as this model considers only Contact Point it neglects the fact that there is still a thin layer of the stock material left for the finishing operation.

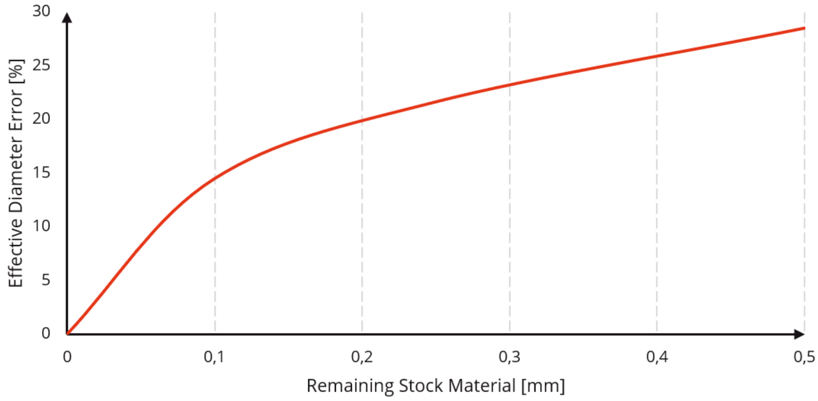


Figure 59 - Influence of a remaining stock material on a Contact Point algorithm precision

This causes computational errors while having thicker remaining stock material as can be seen in the Fig. 59, because the contact of the tool touches the material in a different place than where the contact point is as it is shown in the Fig. 60, where $A_{1,2}$ are the contacts of the tool with the material and CP is the generated Contact Point from CAM.

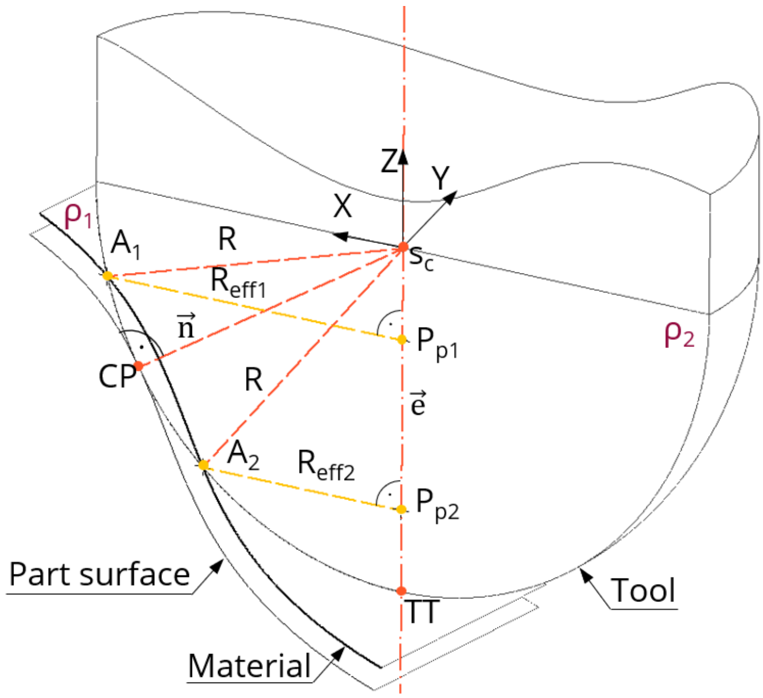


Figure 60 - Contact Point position with the consideration of a remaining stock material

8.3.1 Effective Diameter with Remaining Stock Material

The aim of this approach is to give the user a choice to calculate the effective cutting diameter with respect to the remaining stock material. When the remaining stock material crosses a certain thickness, the Contact Point Algorithm could prove contra productive because of the computational error of the effective diameter and thus the tool overloading and even breakage may occur.

As can be seen in the Fig. 61, there are now two potential contacts $A_{1,2}$ between the material and the tool. To calculate these points, we need to solve for an $A_{1,2}$ as an intersection of two planes and a sphere using formulas (25, 26, 27).

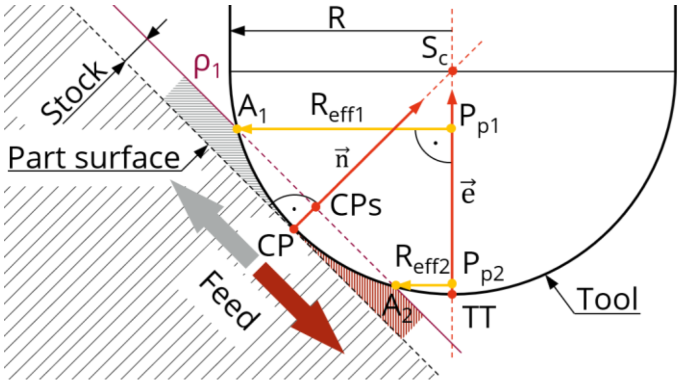


Figure 61 - 2D representation of the effective diameter calculation with remaining stock material

The $\boldsymbol{\rho_1}$ plane (25) is an offset along the surface normal $\vec{n}$ from a Contact Point by remaining stock material and $\boldsymbol{\rho_2}$ plane (26) consists of a tool axis vector $\vec{e}$ and a Contact Point CP. The $\boldsymbol{\rho_1}$ plane is a simplification of the real case, where the part surface is a complex spline as can be seen in the Fig. 60, however since generated contact points have very fine tolerances during finishing operations, this fact is being neglected.

$$\rho_1: n_1 A_{x_{1,2}} + n_2 A_{y_{1,2}} + n_3 A_{z_{1,2}} + d_1 = 0 \tag{25}$$

$$\rho_2: j_1 A_{x_{1,2}} + j_2 A_{y_{1,2}} + j_3 A_{z_{1,2}} + d_2 = 0 \tag{26}$$

$$\left(A_{x_{1,2}} - S_{c_x}\right)^2 + \left(A_{y_{1,2}} - S_{c_y}\right)^2 + \left(A_{z_{1,2}} - S_{c_z}\right)^2 = R^2 \tag{27}$$

In which

$A_{1,2}$ point of intersection [mm]

$\vec{n}$ surface normal vector [-]

$\vec{j}$ normal vector of ρ_2 [-]

S_c center of a ball-end mill [mm]

The effective cutting radius as a distance from the points $A_{1,2}$ to the tool axis $\vec{e}$ is calculated using formula (28) and the value is then multiplied by two to get the effective diameter.

$$R_{eff_{1,2}} = \left|\left\{S_c \cdot \left[\left(A_{1,2} - S_c\right) \cdot \vec{e}\right] \cdot \vec{e}\right\} - A_{1,2}\right| \tag{28}$$

where

S_c center of a ball-end mill [mm]

$A_{1,2}$ point of intersection [mm]

$\vec{e}$ tool axis vector [-]

As those calculations must be done for every iteration of the postprocessor similarly as the Feed Rate Optimization calculations, the algorithm was written in C# as well to ensure a fast execution time and is exported as a Dynamically Loaded Library. This .dll file is then loaded in run-time during postprocessing using the TCL command MOM_run_user_function as mentioned in the chapter 5 - Suitable Programming Languages. The flowchart of the algorithm is depicted in the Fig. 62 and the workflow of the postprocessor in the Fig. 63. The detailed flowchart for effective cutting diameter calculation is in the appendix 2.

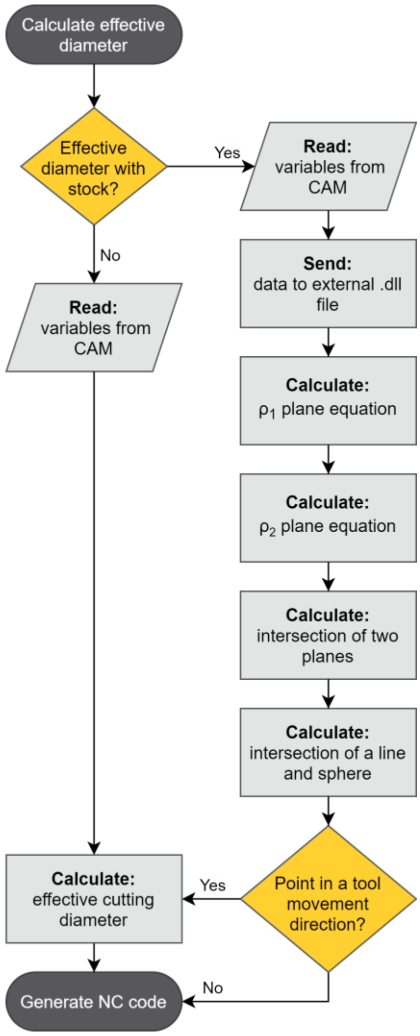


Figure 62 - Flowchart of an effective diameter calculation with remaining stock material

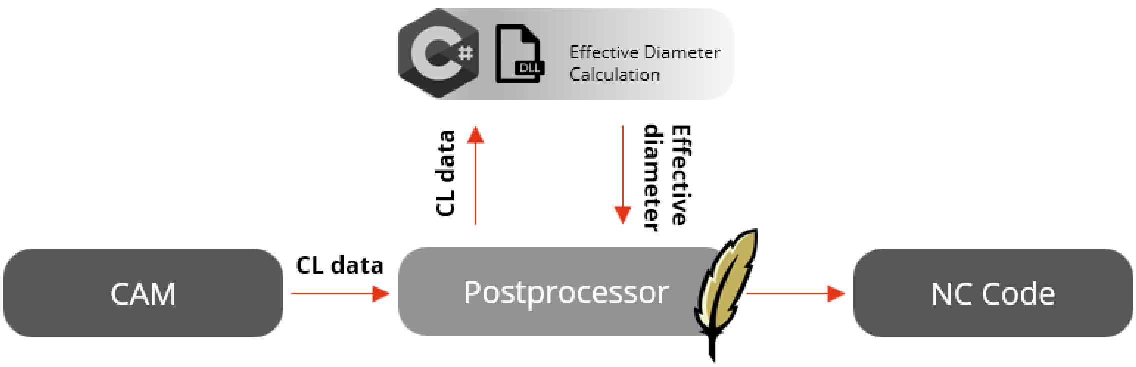


Figure 63 - Workflow of a postprocessing with implemented Cutting Speed Optimization function

8.3.2 Minimal Contact Angle Check

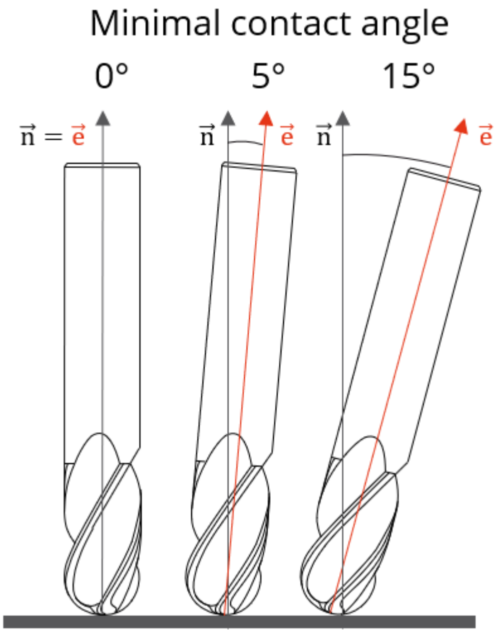


Figure 64 - Minimal contact angle

To avoid flat areas during milling where the effective diameter is close or equal to zero and the calculated spindle rpm would therefore be close or equal to infinity an extra check has been included into the algorithm. The minimal tilt angle check computes the angle between the surface normal $\vec{n}$ in z axis direction and the tool axis $\vec{e}$ in z axis direction. The user can specify the minimal allowed tilt angle (Fig. 64) in the custom created UDE in Siemens NX as can be seen in the Fig. 64. If the angle is smaller than the value specified by the user, the spindle rpm will keep the same value as in the previous NC block and the effective diameter will not be calculated in this case.

8.3.3 Comparison between the Computational Options

To provide an insight what exactly it means to choose the Contact Point or the Remaining Material effective diameter calculation a demonstrative geometry has been selected and it is shown in the Fig. 67 along with the programmed tool path and the cutting and spindle speed graphs for the given tool path. As can be seen in the graph, the Contact Point algorithm achieves the highest spindle rpm, whereas the Remaining Material option tends to be more conservative as it includes the stock left for finishing in the calculation, which basically offsets the effective diameter to higher values as thus the required spindle rpm to keep the cutting speed constant are lower. As the Contact Point option achieves higher spindle rpm, it can also happen that it sometimes overshoots the programmed cutting speed, because the spindle acceleration may not be high enough to react that quickly to surface changes and the spindle rpm start to ‘drag’ behind the wanted rpm value, resulting into the overshooting. This, on the other hand is very well suppressed while using the Remaining Material option, because the spindle rpm do not achieve as high values as with the Contact Point algorithm.

It is also worth mentioning, that the cutting speed in the middle area on the part is only theoretical, as there is no contact between the part and the tool, because of the small cavity and that the cutting and spindle speed values in this area are also affected by the Minimal Tilt Angle Check mentioned in the previous chapter, which does not allow the spindle rpm to get close to infinity as the effective diameter in this area is very close to zero.

8.4 Optimization Modules Concatenation

To make the whole optimization more scalable and user-friendly, the stand-alone optimizations are concatenated together, as they have been two separate modules so far. This not only makes the source code cleaner and easier to maintain, but also unlocks the possibility to combine the optimizations together, where the user can exactly choose what should and what should not be optimized. The concatenated optimization structure is shown in the Fig. 65.

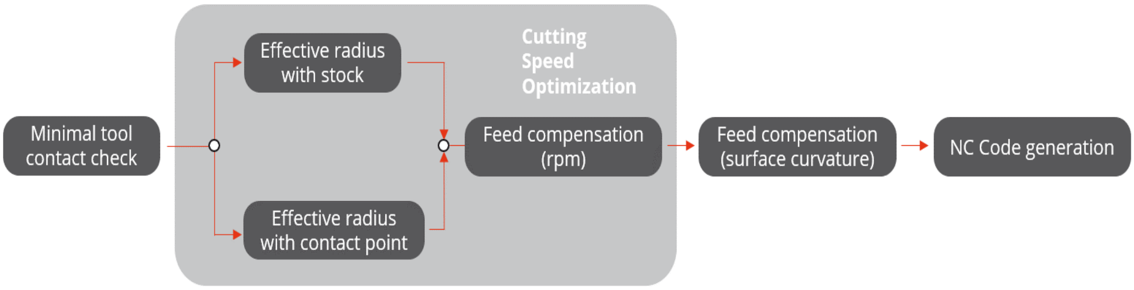


Figure 65 – Concatenated optimization structure

A special window with which the user can control the logic of the postprocessing (optimization) has also been created directly in Siemens NX and it is shown in the Fig. 66. This makes the postprocessor with included optimization flexible, as it can either be used without any optimization, or partially with either Cutting Speed Optimization or Feed Rate Optimization, or with both optimizations at the same time. The control logic of the postprocessor is depicted in the Fig. 68 and in the appendix 3.

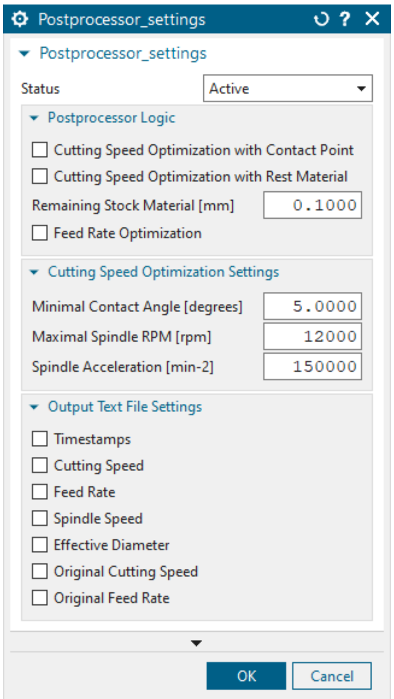


Figure 66 - Custom Siemens NX UDE

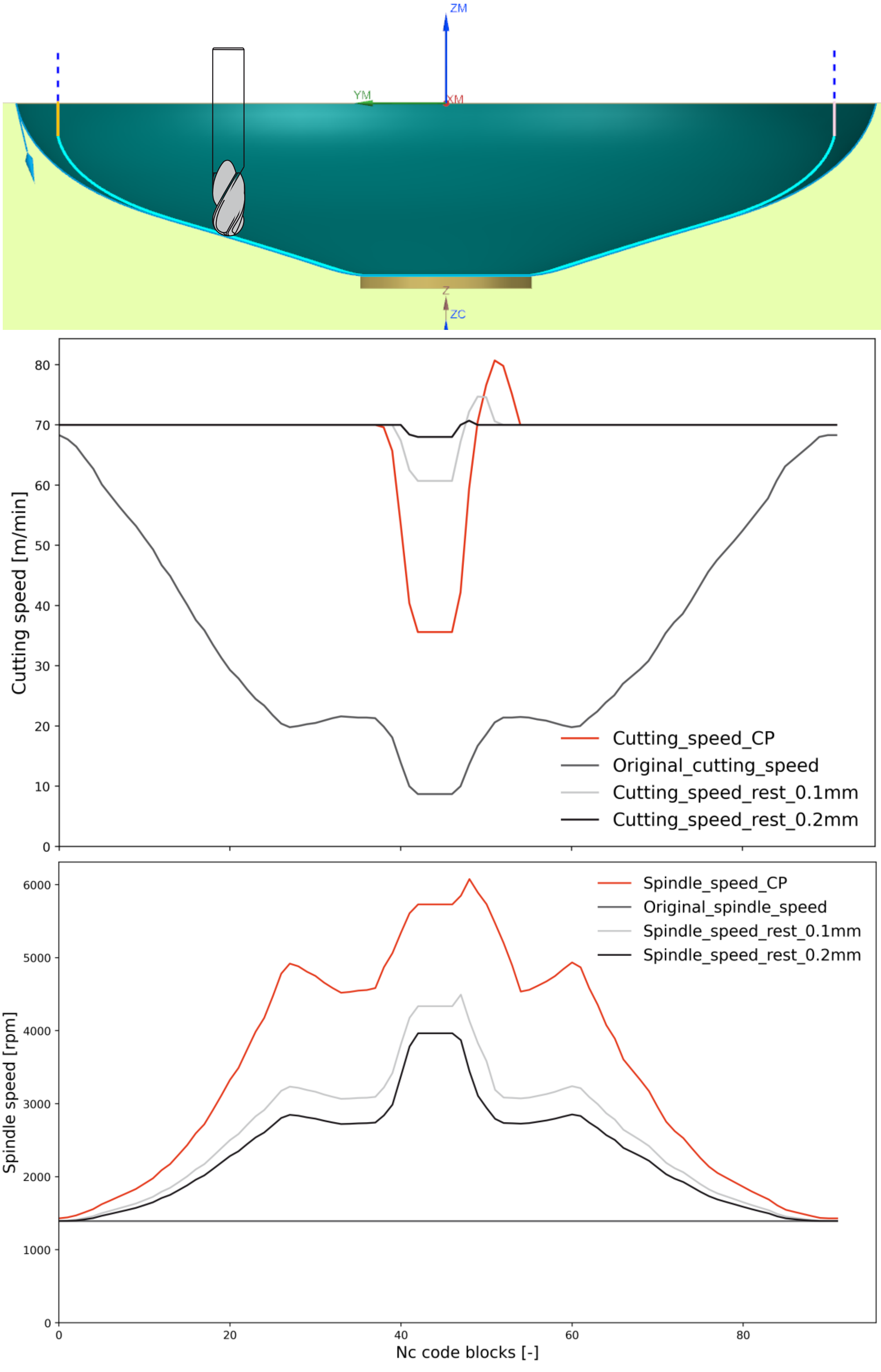


Figure 67 - Demonstrative geometry with cutting and spindle speeds comparison

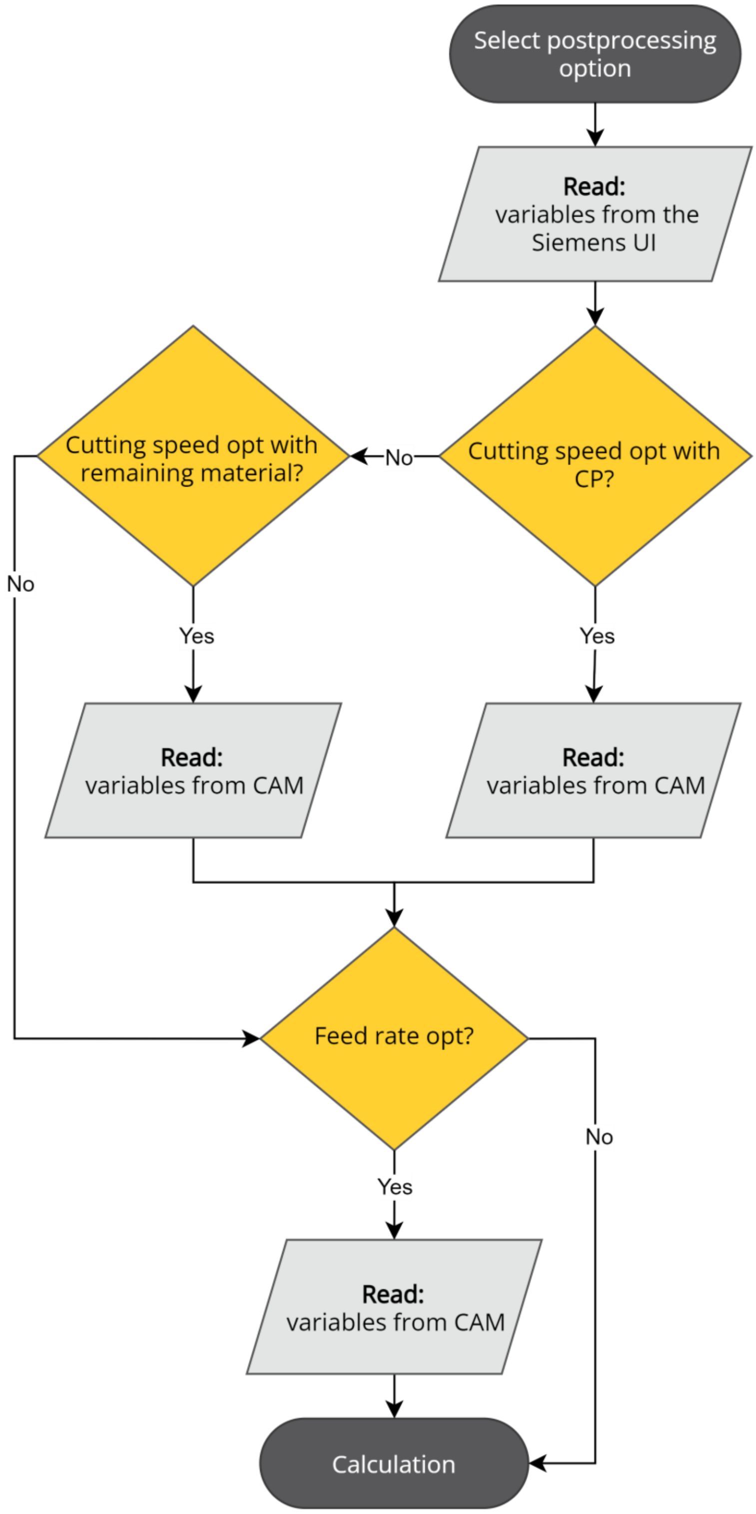


Figure 68 – Control logic of the postprocessor

9 Verification of the Optimization Functions on a Milling Machine Tools

As both optimization functions work with the data from CAM, it is necessary to verify, if the generated values for cutting speed and feed rate compensations are also achieved directly during milling on the machine tool.

9.1 Used Machine Tools and Test Part

The graphs shown in the Fig. 58 and Fig. 67 for both optimization principles are just mathematically calculated values with the consideration of the spindle kinematic parameters but it does not consider all the variables that influence the machining process. Given this fact it is necessary to test the optimizations during machining and compare the results with the calculations from the postprocessor. To further emphasize the influence of the used machine tool, several machine tools have been used to carry out the same optimized NC code and the machining times from the CAM software and after machining have been compared.

As a demonstrative geometry, a bottom part of a bowl mold has been selected and it is shown in the Fig. 69. The material is a duplex stainless steel (DIN 1.4462), which has been roughed and semifinished to leave an almost even 0,25 mm remaining stock material on the surface for the finishing operation. The tool used for the finishing operation was a 16 mm 2teeth ball end mill, the cutting speed value set in CAM

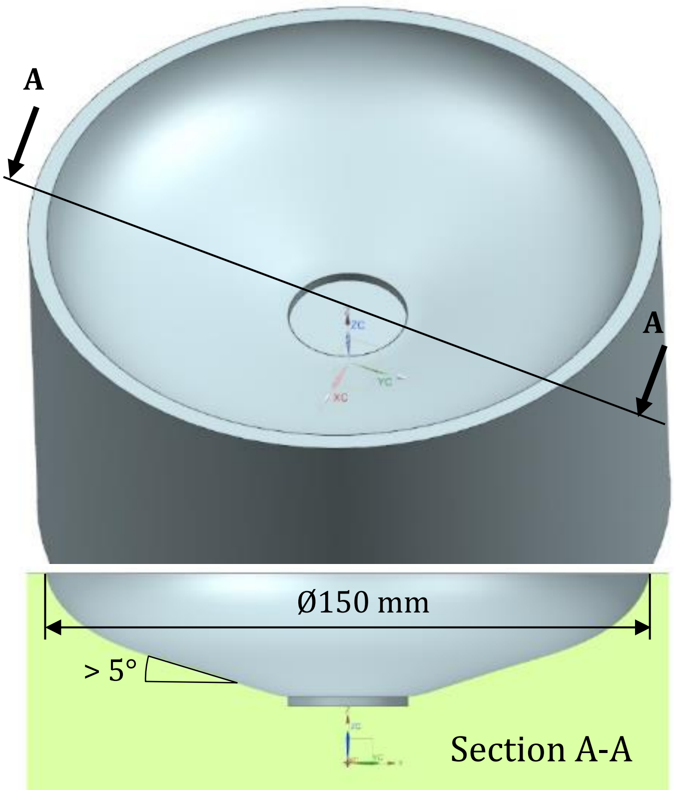


Figure 69 - Demonstrative geometry

was 70 m/min and the feed per tooth was 0,1 mm.

The finishing strategy tool paths are shown in the Fig. 70. This strategy was used on all the machine tools listed in the Table 3 and the machining times without any optimizations and then either with the Feed Rate Optimization or with the combination of a Cutting Speed and the Feed Rate Optimization were compared. The spindle accelerations listed in the Table 3 were measured directly on the machine tools during a ramp up of the spindle to the wanted spindle rpm value and the accelerations were then used during the tool path preparations in CAM for each individual machine tool. A selected region from the finishing geometry is displayed in the Fig. 71 along with the optimized spindle speeds and feed rates for the combination of both optimization functions. The whole graph for the finishing operation is then shown in appendices 4 and 5.

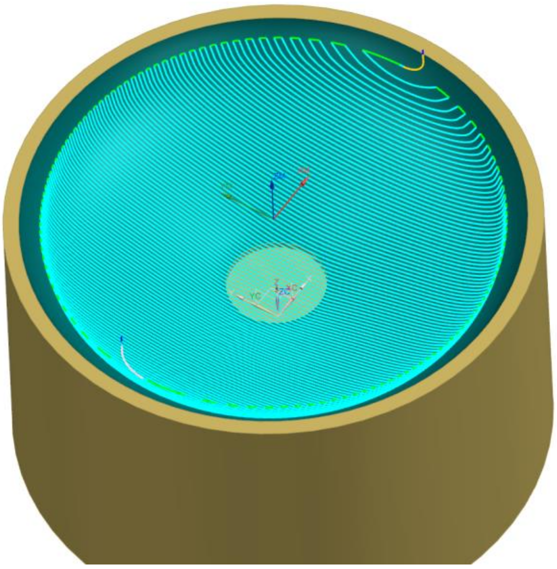
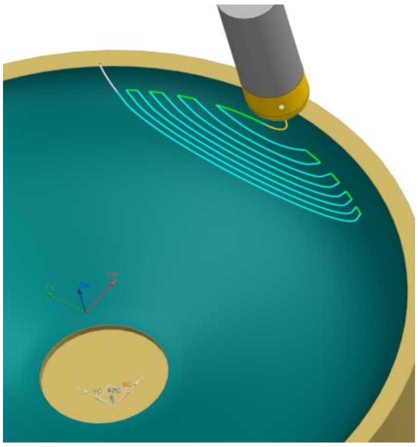


Figure 70 - Demonstrative geometry finishing tool path



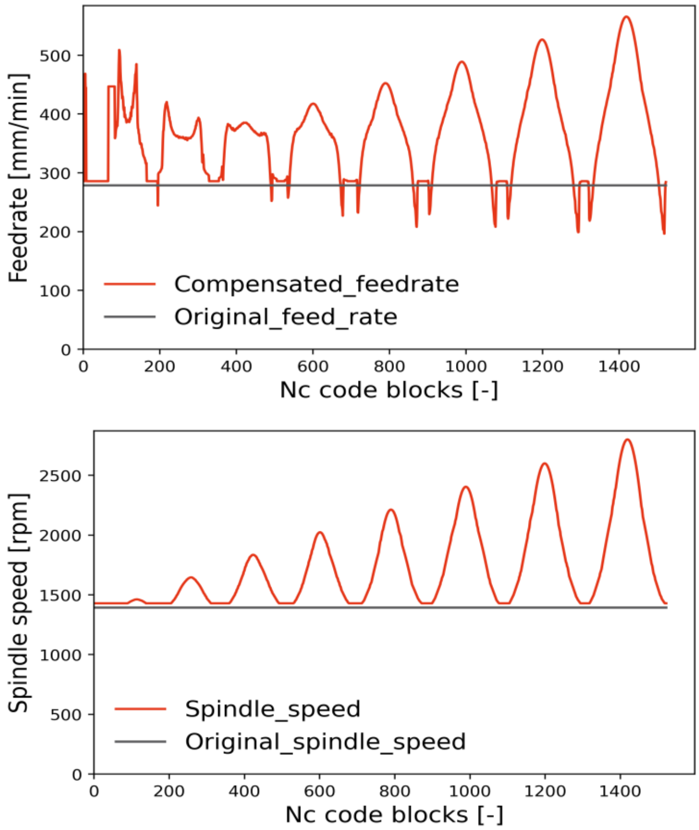
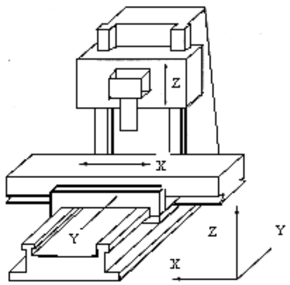
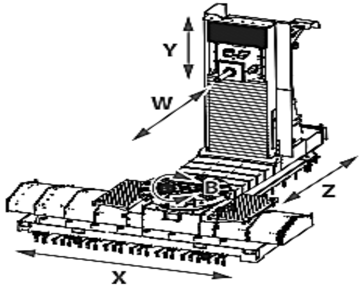
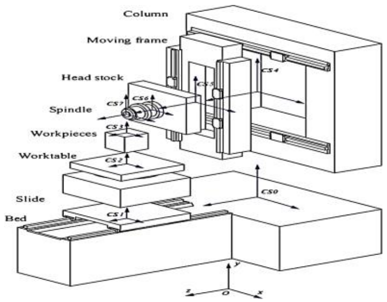


Figure 71 - Optimized spindle speed and feed rate for a section from the finishing operation

Table 3 - Overview of the machine tools used for the verification of the optimization functions

Machine	Machine type	Machine tool parameters				Used optimization	Mach. time without optimization [min]	Mach. time with optimization [min]
		Max. spindle rpm	Travels [mm]	Spindle acceleration [min^{-2}]	Motion axes drive			
Tajmac ZPS MCFV5050LN		15 000	X: 500 Y: 400 Z: 400	150 000	Linear drive	Cutting speed + Feed rate	53,5	27,5
TOS Varnsdorf WHT110C		4 000	X: 3000 Y: 1250 Z: 1500 B: 360° W: 650	60 000	Ball screw	Feed rate	53,5	55,1
Tajmac ZPS MCH630i		18 000	X: 750 Y: 700 Z: 770 B: 360°	280 000	Ball screw	Cutting speed + Feed rate	53,3	27,5

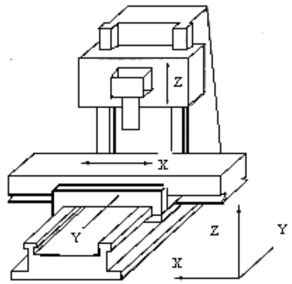
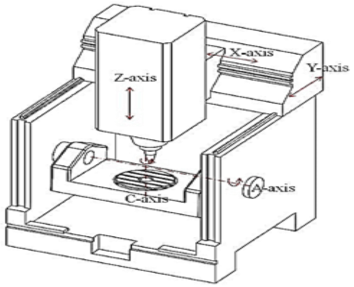
Haas TM1[1]		4 000	X: 762 Y: 406 Z: 406	126 000	Ball screw	Cutting speed + Feed rate	53,5	28,7
Kovosvit MAS MCU700		24 000	X: 700 Y: 820 Z: 550 A: ± 120° C: 360°	280 000	Ball screw	Cutting speed + Feed rate	53,3	27,5

Image sources top to bottom: (Khan, et al., 2008); (TOS Varnsdorf); (Zhang, et al., 2019); (Chris, 2021)

[1] The continuos spindle rpm change is not yet possible for this machine. The displayed values are results from the machine without continuos spindle rpm change.

9.2 Feed Rate Optimization Verification

To further verify and discover the pros and cons of the optimization functions, the demonstrative geometry was machined on the Tajmac ZPS MCFV5050LN milling machine tool and not only the machining time, but also the tool wear and surface roughness were measured. After the whole finishing operation (Fig. 70), the tool was examined under a microscope, the tool cutting edges were photographed, and the tool wear measured. Firstly, the conventional finishing operation was tested and the whole process was repeated until the tool wear reached 200 μm. The same process was then carried out twice for the same finishing operation with the Feed Rate Optimization turned on. The results are summarized in the Table 4. The process parameters were set as previously mentioned: cutting speed – 70 m/min, feed per tooth – 0,1 mm, 16 mm ball end mill with two teeth and machined material was a DIN 1.4462 duplex steel. Mahr LD130 was used as a roughness measuring device and Keyence VHX-7000 with RRR lens as a Microscope.

Table 4 - Results of the Feed Rate Optimization verification

Optimization	Tool wear	Machining time [min]	Measured surface roughness R_a [μm]
Conventional		53,5	0,3
Feed rate		55	0,3

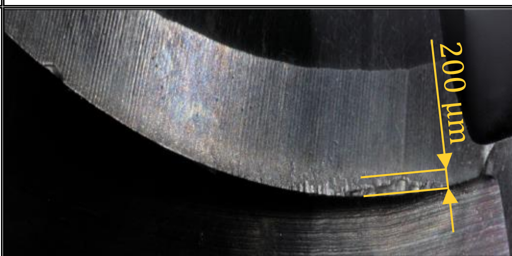
The **photographs** of the tool show the tool wear **after** manufacturing **eight workpieces**. The tool wear seems to be more evenly distributed along the tool blade while using the Feed Rate Optimization. However, the number of manufactured workpieces with one tool was the same as when using the conventional method (eight). The surface roughness measurement did not show any difference between the conventional and optimized tool paths. Finally, the machining time for the optimized NC code was 1,5 min longer than the original NC code, which is an increase of 2,8 %. The feed rate graph for this verification is shown in the appendix 6.

It is also worth mentioning, that the increase of the machining time directly correlates with the shape of the part as is discussed in the chapter 3.2 - Optimization of cutting conditions by CTU in detail and can be seen in the Fig. 28. If the **part had a convex shape** instead of a concave shape, the optimization would have **decreased** the **machining time**.

9.3 Cutting Speed Optimization Verification

The exact same tests were carried out for the Cutting Speed Optimizations as well. Firstly, the option that calculates the effective diameter from the contact point was measured and after that the option considering the remaining stock material. Process parameters, tool, machine tool, and machined material were the same as during the previously mentioned verifications. Both optimization options were verified twice. The results are summarized in the Table 5.

Table 5 - Results of the Cutting Speed Optimization verification

Optimization	Tool wear	Machining time [min]	Measured surface roughness R_a [µm]
Conventional		53,5	0,30
Contact Point		26	0,29
Remaining Material		36	0,29

The **photographs of the tool** show the tool wear **after** manufacturing **eight workpieces.** The tool wear improves significantly while using either the Contact Point or Remaining Material Cutting Speed Optimization as can be deducted based on the images in Table 5. The tool with conventional NC code was able to manufacture eight workpieces before reaching the 200 μm tool wear threshold, while the optimized NC codes machined 15 (Remaining Material) and 23 (Contact Point) workpieces. This result was surprising, because the Remaining Material calculation should be more precise in the terms of optimized cutting conditions and should therefore show better results in tool wear than the Contact Point option. However, the cutting speed and feed per tooth originally set in CAM based on the recommendations from the tool manufacturer for the conventional strategy might have been underestimated. Given the fact, that the Contact Point option is more aggressive than the Remaining Material option in terms of achieved spindle speeds and feeds as can be seen in the appendices 7 and 8, it might have improved the overall cutting conditions more than the Remaining Material option, because of the initial underestimation of the cutting conditions and therefore improved the tool wear results.

The machining time also improved significantly for both effective diameter calculation principles, while the Contact Point option saved more machining time, the Remaining material option is more conservative in terms of required machine tool dynamics as it does not reach that high spindle rpm. The surface roughness did not change much while using the optimized NC codes. The spindle speed and feed rate graphs for those verifications are shown in the appendix 7.

9.4 Concatenated Optimizations Verification

At last, the exact same tests were carried out for the combination of Cutting Speed Optimization with Contact Point and Feed Rate Optimization. Process parameters, tool, machine tool, and machined material were the same as during the previously mentioned verifications. This optimization option was also verified twice. The results are summarized in the Table 6.

Table 6 - Results of the concatenated optimizations verification

Optimization	Tool wear	Machining time [min]	Measured surface roughness R_a [μm]
Conventional		53,5	0,3
Combination of both cutting speed and feed rate		27,5	0,29

The **photographs** of the tool show the tool wear **after** manufacturing **eight workpieces.** As can be seen on those photographs, the tool wear is barely visible after manufacturing eight workpieces with the combination of Feed Rate and Cutting Speed Optimizations. The total number of workpieces manufactured with one tool was 25. The machining time is 1,5 min longer than the pure Cutting Speed Optimization with contact point, which corresponds with the results from the Feed Rate Optimization verification. The measured surface roughness did not change in comparison to the conventional or optimized NC code. The spindle speed and feed rate graphs for this optimization are shown in the appendices 4 and 5.

9.5 Verification Results

The measured machining times and tool wear depends heavily on the machined part geometry. The presented results were all measured on the same demonstrative geometry shown in Fig. 69.

The Feed Rate Optimization has more or less the same results as machining the part conventionally without any optimization at all. The tool wear is more evenly distributed along the tool blade, however, **both** options **machined** in total **eight parts** with one tool. The **machining time** was **longer by 2,8 %** in comparison to non-optimized NC code and **surface roughness did not change at all**. The machining time change depends on whether the part surface is a convex or concave shape. If the **part surface** were **concave**, the machining time **would decrease** in comparison to the non-optimized NC code.

The Cutting Speed Optimization proved to have significant benefits on machining time and tool wear. The Contact Point effective diameter calculation had bigger benefits than the Remaining Material calculation, however the machine tool must be very dynamical. Using the **Remaining Material option, 15 parts** were manufactures and while using **the Contact Point option it was 23 parts**. The machining **time saving** were **32,7 %** for the Remaining Material option and **51,4 %** for the Contact Point option in comparison to non-optimized NC code. **The surface roughness did not change at all**.

The combination of both optimizations confirmed the results acquired from the verifications of the single optimizations. The tool wear has improved slightly as the **tool machined 25 parts** in total. The **machining time rose by 1,5 min** in comparison to the Cutting Speed Optimization with contact point, which is exactly the difference between non-optimized NC code and NC code with Feed Rate Optimization. **The surface roughness did not change at all again.**

10 Summary and Outlook

It was proven, that during milling of geometrically complex surfaces, the feed rate and cutting speed in the contact point between part and tool are not always the programmed values. Given this fact, it is necessary to recalculate the spindle rpm and feed rates to keep the cutting speed and feed per tooth constant as is originally intended.

Several options of how to compensate the feed rate values are presented in this thesis. The most suitable one was 'Best Fit Circle Calculation with Projected Contact Point' because of the computational accuracy. Furthermore, the optimization of spindle speed control was enhanced by calculation of the effective cutting diameter considering the remaining stock material.
The enhanced algorithms for cutting speed and feed rate compensations were both programmed in C# and exported as a .dll, which is loaded in run-time during postprocessing directly from the postprocessor. Given this fact, the user can exactly choose which optimization or combination of optimizations will be used.

Both optimizations were then verified on several milling machine tools to see the influence of a particular machine tool on the optimizations. Moreover, a demonstrative geometry has been selected, machined, and the tool wear, machining times and surface roughness values were observed and measured. Using both optimizations, the tool was able to machine more than three times more parts (8 conventionally, 25 with optimizations) and the **machining time was reduced by 26 minutes** from 53,5 minutes to 27,5 minutes which is equal to **51,4 %** time savings, while achieving the same surface quality in comparison to conventional machining.

According to the syllabus of the thesis the state of the art of the machining process optimizations was conducted in the fields of CAM software and machine tool control systems. Furthermore, the influence of the machine tool and its mechanical and electrical components on the optimization functions was summarized. The optimizations utilizing cutting speed and feed rate values were either improved or completely reworked to provide more accurate results. Moreover, the optimizations were concatenated together into one .dll file which is implemented in a postprocessor. Verification of optimizations has been performed using real machine tools. It can be summarized, that the desired goals were achieved.

References

Albertelli P., Mussi V. and Monno M. The analysis of tool life and wear mechanisms in spindle speed variation machining [Journal]. - London : Springer, 2014.

Altintas Yusuf Manufacturing Automation [Book]. - 2012. - ISBN 9781107001480.

Arun K.S., Huang T.S. and Blostein S.D. Least-Squares Fitting of Two 3-D Point Sets [Article] // Transactions on Pattern Analysis and Machine Intelligence. - USA : IEEE, 1987.

Autodesk Autodesk Help [Online] // Fusion 360 Help. - Autodesk, 2016. - 10 2021. - https://help.autodesk.com/view/fusion360/ENU/?guid=GUIDA73542E9-ED9C-4BD9-A87D-3A0ECA8BEB41.

Autodesk Creating MachineDNA [Online] // Autodesk Knowledge. - Autodesk, 2020. - 10 2021. - https://knowledge.autodesk.com/support/powermill/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/PWRM-ReferenceHelp/files/GUID-E1694D20-D4FD-4E19-B6A3-8D3953B2206C-htm.html.

Autodesk Fusion 360 Help [Online] // Adaptive Clearing Dialog. - Autodesk, 2016. - 10 2021. - https://help.autodesk.com/view/fusion360/ENU/?guid=GUID09E44604-DAD8-47D6-ADC6-C100869DE724.

Bartoš Filip Application of Dynamic Control of Cutting Conditions on CNC Milling Machine Tools // Bachelor's thesis. - Prague : CTU, 2019.

Berthold J. [et al.] Identification of modal parameters of machine tools during cutting by operational modal analysis [Journal]. - Chemnitz : Elsevier, 2018.

Brecher Christian Werkzeugmaschinen Fertigungssysteme 3 [Book]. - Berlin : Springer, 2021. - ISBN 978-3-662-46568-4.

cadcamlessons CadCamLessons [Online] // 2D Adaptive Clearing Fusion 360. - 2019. - 10 2021. - http://cadcamlessons.com/2d-adaptive-clearing-fusion-360/.

CGTech Axiomtech [Online] // OptiPath. - Axiom Tech, 2010. - 10 2021. - https://www.axiomtech.cz/24888-optimalizace-optipath.

CGTech Axiomtech [Online] // Force. - Axiom Tech, 2010. - 10 2021. - https://www.axiomtech.cz/24894-force.

Collins Danielle Linear Motion Tips [Online] // How to reduce jerk in linear motion systems. - 5 2017. - 10 2021. - https://www.linearmotiontips.com/how-to-reduce-jerk-in-linear-motion-systems/.

Colonial Tool Group Gear driven spindles [Online]. - 2016. - 10 2021. - http://www.colonialtool.com/gear-driven-spindles/.

Denkena Berend and Tönshoff Hans Kurt Spanen [Book]. - 2011. - ISBN 978-3-642-19772-7.

Fritz Alfred Herbert Fertigungstechnik [Book]. - Berlin : Springer, 2015. - ISBN 978-3-662-46555-4.

Goman V.V. Prospects of Upgrade of the Main Drives of CNC Machine Tools Using Mechatronic Modules [Journal]. - Yekaterinburg : Knowledge E, 2020.

Górecki Jan Postprocessor with Feed-Rate Correction for Angular Velocity for CNC Machines // Master's thesis. - Prague : CTU, 2020.

Haas Haascnc [Online] // M38 / M39 Spindle Speed Variation. - Haas, 2015. - 10 2021. - https://www.haascnc.com/video/tips/dxr2wddffm4.html.

Heidenhain Dynamic Efficiency - OCM [Online] // Heidenhain. - 2020. - 10 2021. - https://news.heidenhain.com/fileadmin/semiconductor/Flyer_OCM_en.pdf.

Heidenhain Heidenhaingb [Online] // Dynamic Efficiency. - Heindenhain, 2013. - 10 2021. - https://www.heidenhaingb.com/wp-content/uploads/pdf-files/Dynamic_Efficiency_en.pdf.

Chris 5 Axis CNC [Online] // Pinterest. - 2021. - 12 2021. - https://www.pinterest.com/charpr0205/5-axis-cnc/.

Icam Icam [Online] // SmartPack. - Icam, 2010. - 10 2021. - https://icam.com/tool-path-optimization-software/.

Käsemodel Rodrigo Berretta [et al.] CAD/CAM interfaced algorithm reduces cutting force, roughness, and machining time in free-form milling [Journal]. - London : Springer, 2020.

Khan Abdul Wahid and Chen Wuyi Calibration of CNC milling machine by direct method [Journal]. - Beijing : Spie, 2008.

Marek Jiří Konstrukce CNC obráběcích strojů III [Book]. - Prague : MM Publishing, 2014. - ISBN 978-802-6067-801.

NCBrain NCBrain [Online] // Product Solutions. - AICam, 2017. - 10 2021. - http://www.ncbrain.com/ab-1337.

NCSimul Hexagon [Online] // NCSimul. - 2012. - 10 2021. - https://www.ncsimul.com/ncsimul-machine.

Nohál Libor Deskripce vřeten obráběcích strojů // Bachelor's thesis. - Brno : BUT, 2008.

Seguy Sébastien, Insperger Tamás and Arnaud Lionel On the stability of high-speed milling with spindle speed variation [Journal]. - London : Springer, 2009.

Stejskal M. [et al.] Optimization of Tool Axis Orientations in Multi-Axis Toolpaths to Increase Surface Quality and Productivity [Journal]. - Prague : Elsevier, 2020.

Stejskal Michal Increasing the Efficiency of Machining on CNC Machine Tools // Master's thesis. - Prague : CTU, 2017.

Technickytydenik Technickytydenik [Online] // Technicke Publikace. - 2013. - 10 2021. - https://www.technickytydenik.cz/rubriky/archiv/vyuzijte-potencial-vaseho-frezovaciho-centra-naplno-s-technologiemi-delcam-vortex-a-machine-dna_20620.html.

TOS Varnsdorf tosvarnsdorf [Online] // Výrobní program. - 12 2021. - https://www.tosvarnsdorf.cz/files/machines/tos-varnsdorf-vyrobni-program-cz.pdf.

Vavruska P., Zeman P. and Stejskal M. Reducing Machining Time by Pre-Process Control of Spindle Speed and Feed Rate in Milling Strategies [Journal]. - Prague : Elsevier, 2018.

Vavruška Petr Technological Extensions of Postprocessors for Multi-Axis Machine Tools // Dissertation thesis. - Prague : CTU, 2013.

Zerspanungstechnik Duoblock premieren von Dmg Mori [Online]. - 2018. - 12 2018. - http://www.zerspanungstechnik.de/2015/03/03/duoblock-premieren-von-dmg-mori/.

Zhang Li and Lianjie Ma A novel integrated process–machine approach for designing customized milling machines [Journal]. - Qinhuangdao : Springer, 2019.

List of Figures

List of Tables

List of Appendices

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF MECHANICAL ENGINEERING

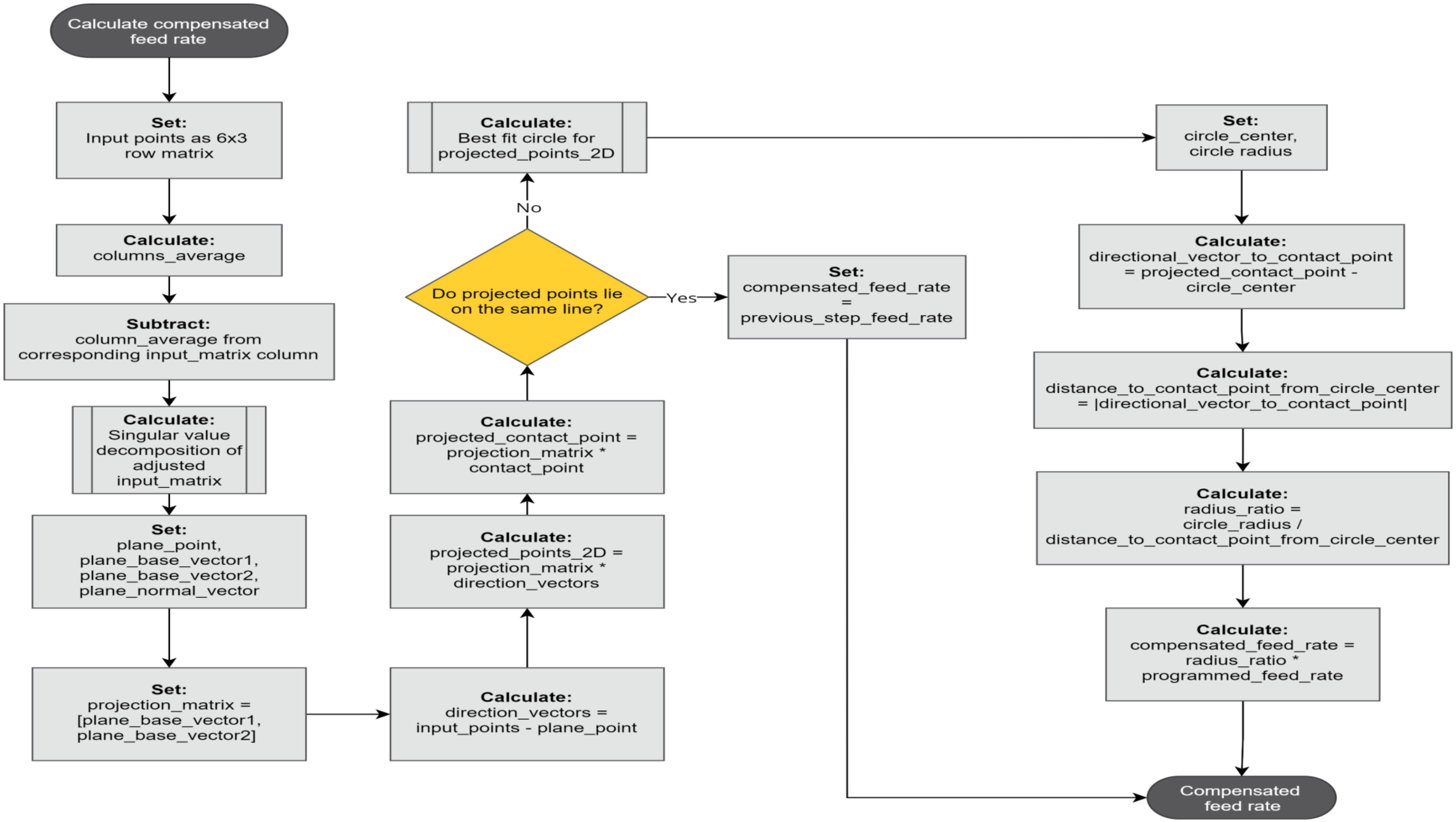
DEPARTMENT OF PRODUCTION MACHINES AND EQUIPMENT

MASTER’S THESIS

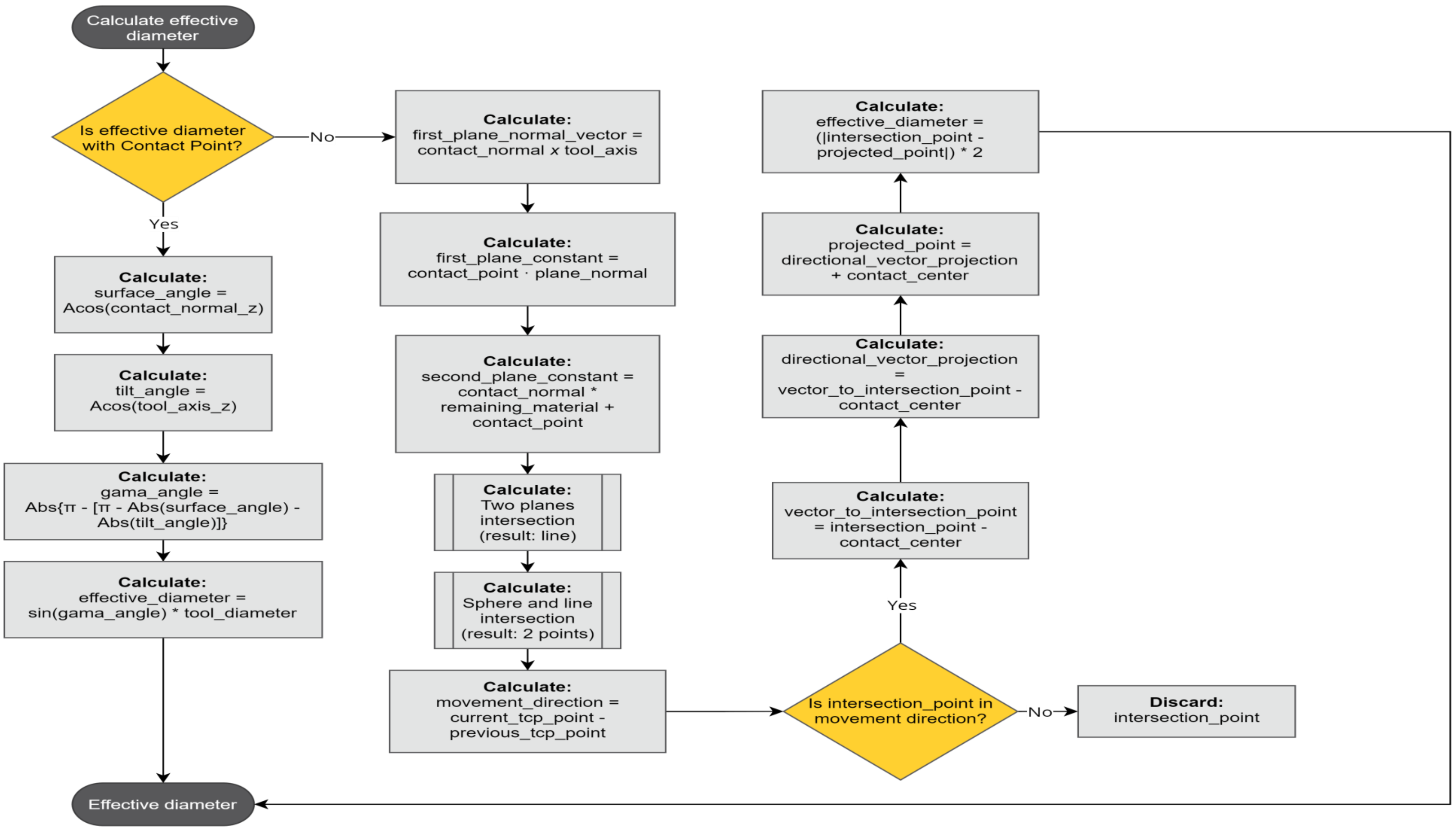
TEXT APPENDICES

2022 **Filip Bartoš**

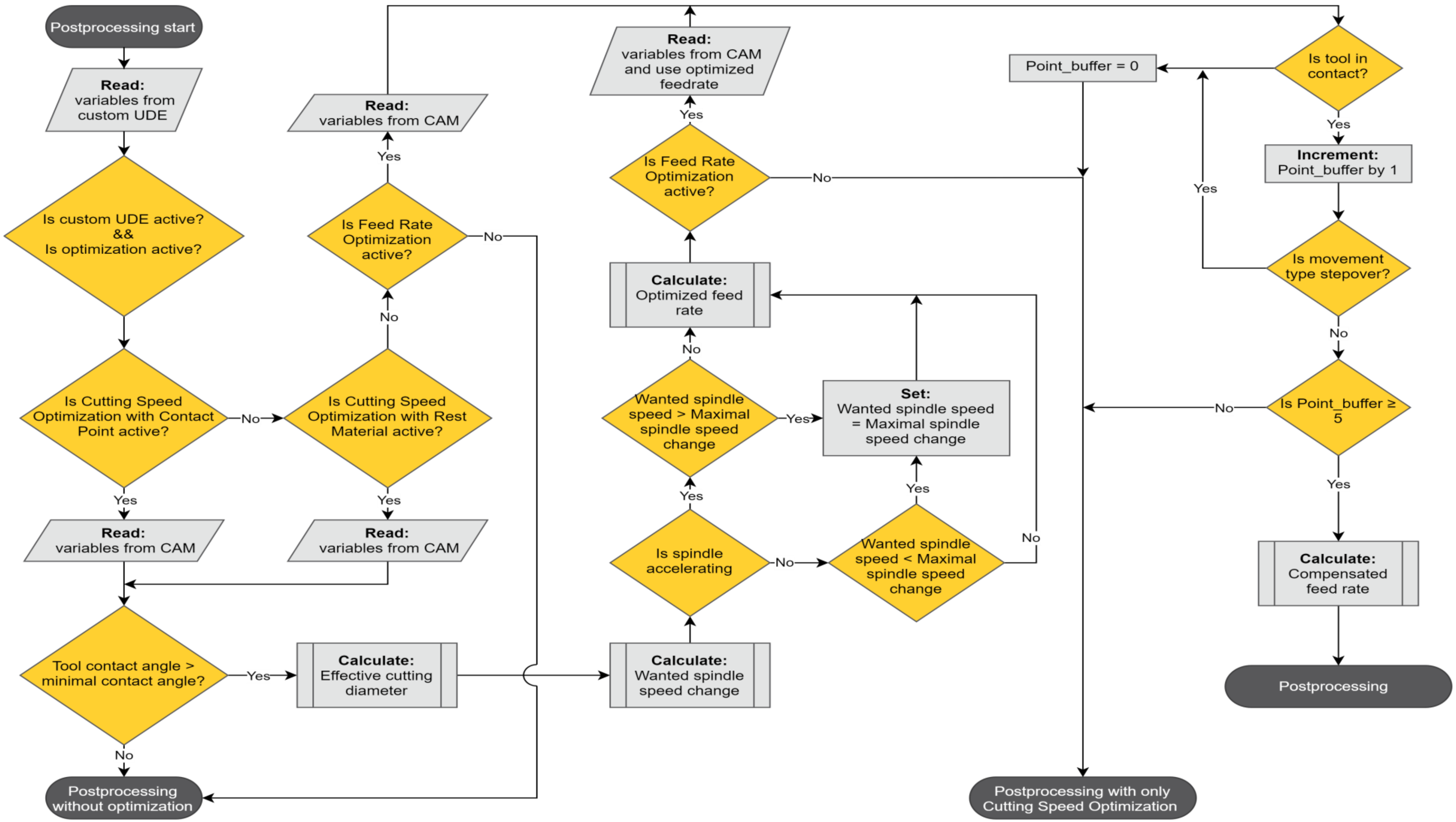
Appendix 1: Detailed logic for Feed Rate Optimization algorithm



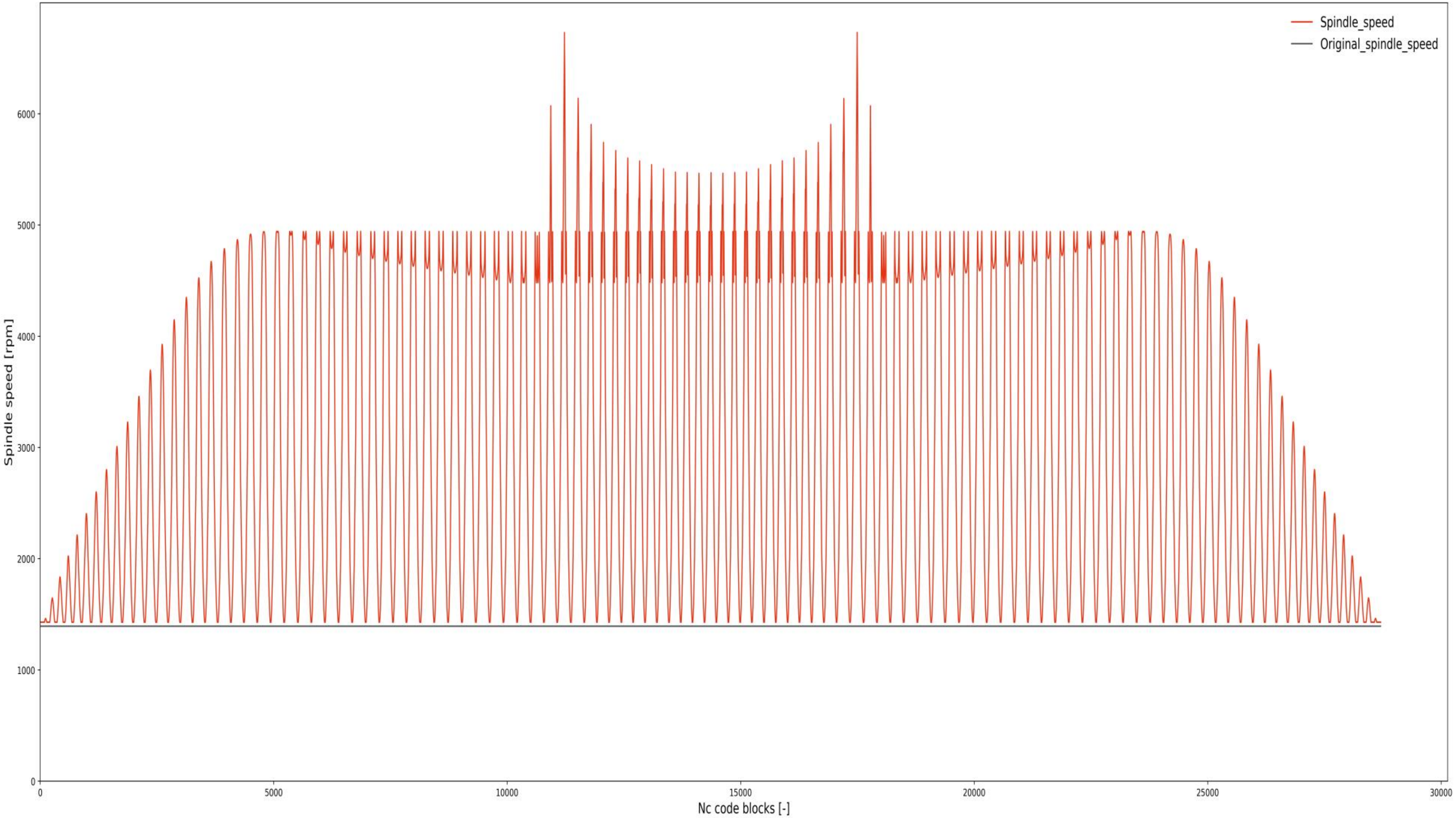
Appendix 2: Detailed logic for effective cutting diameter calculation



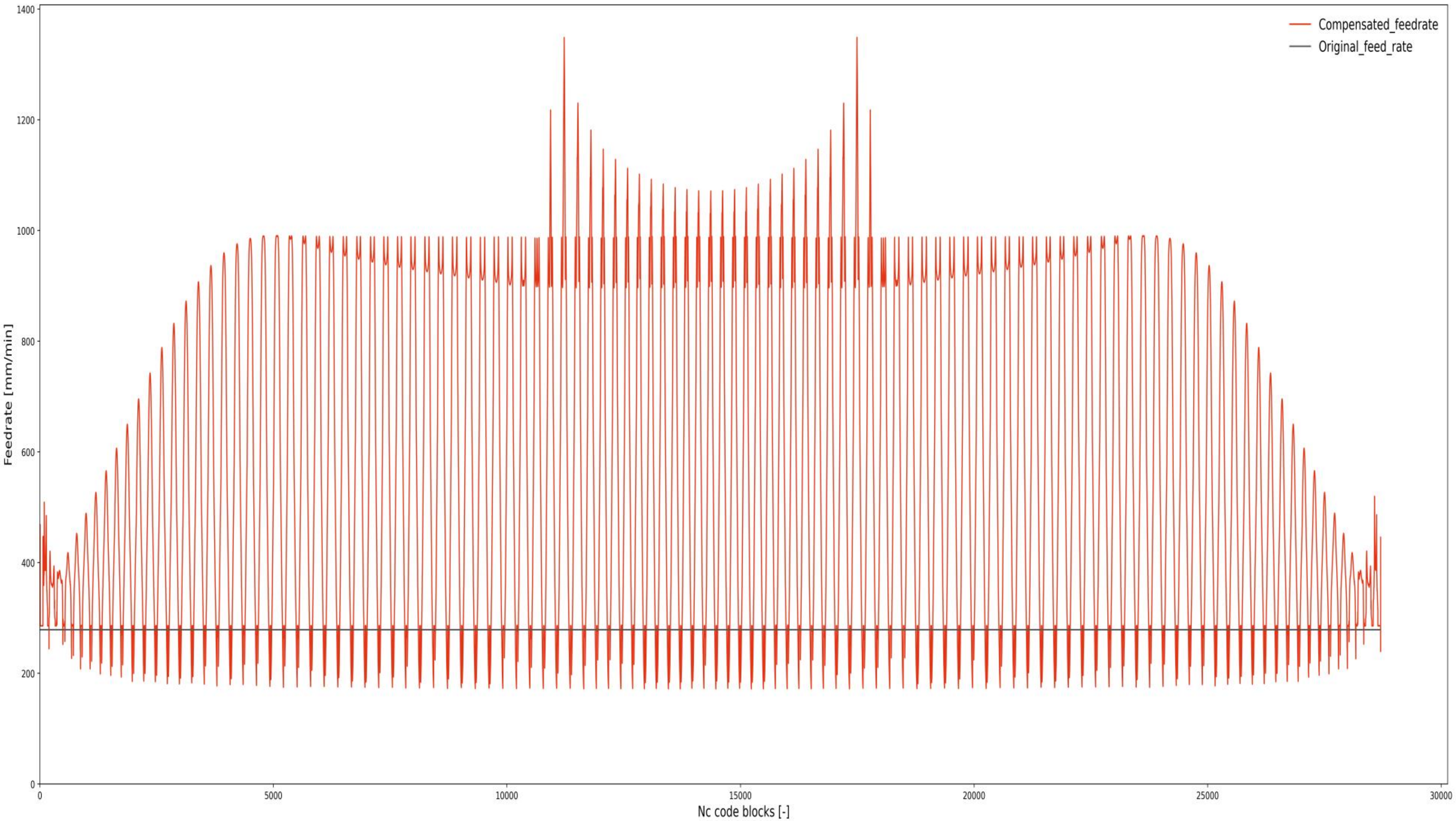
Appendix 3: Control logic for postprocessing with custom UDE and .dll file



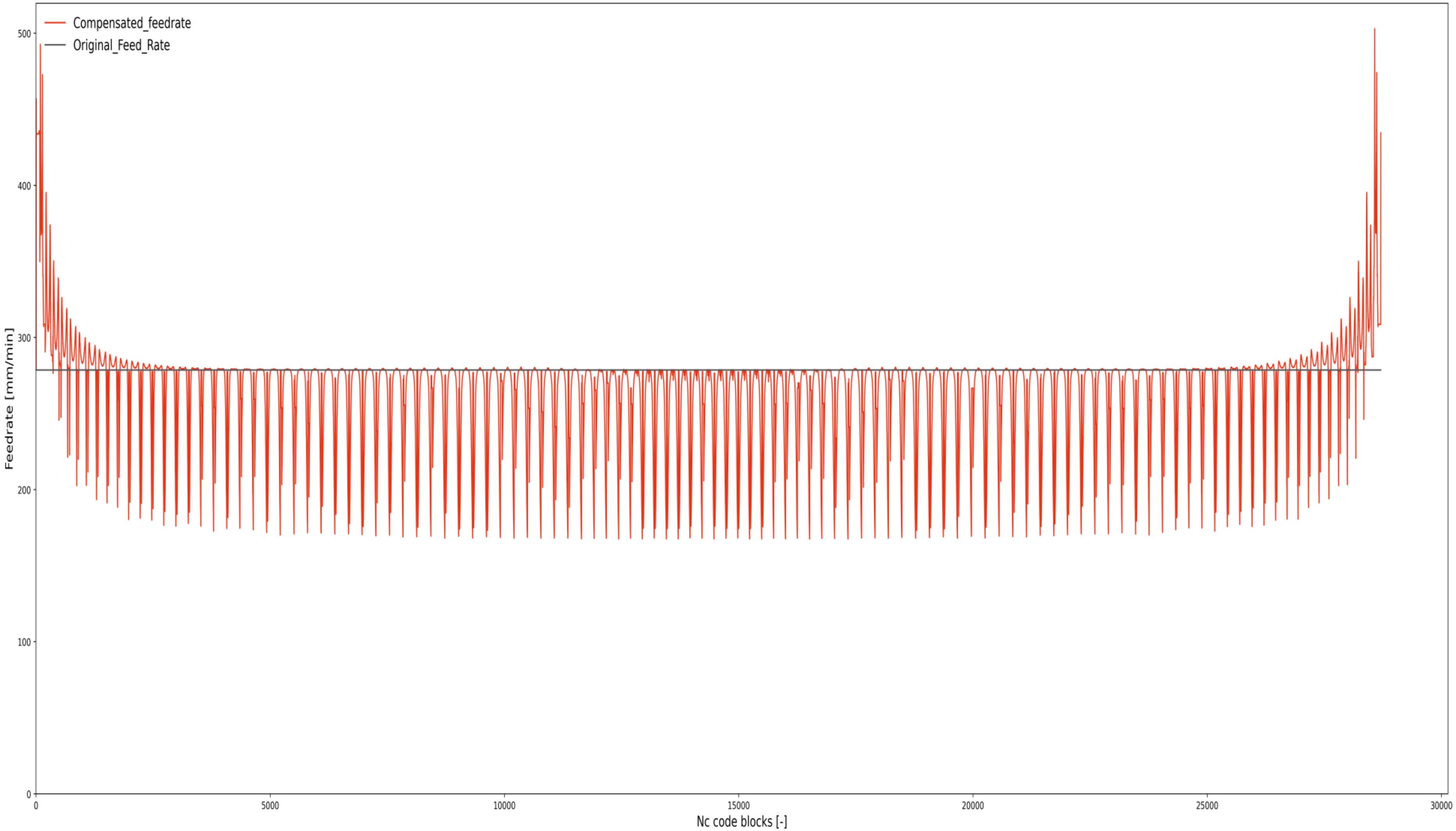
Appendix 4: Spindle speed graph for the finishing operation on the demonstrative geometry using combination of both optimizations



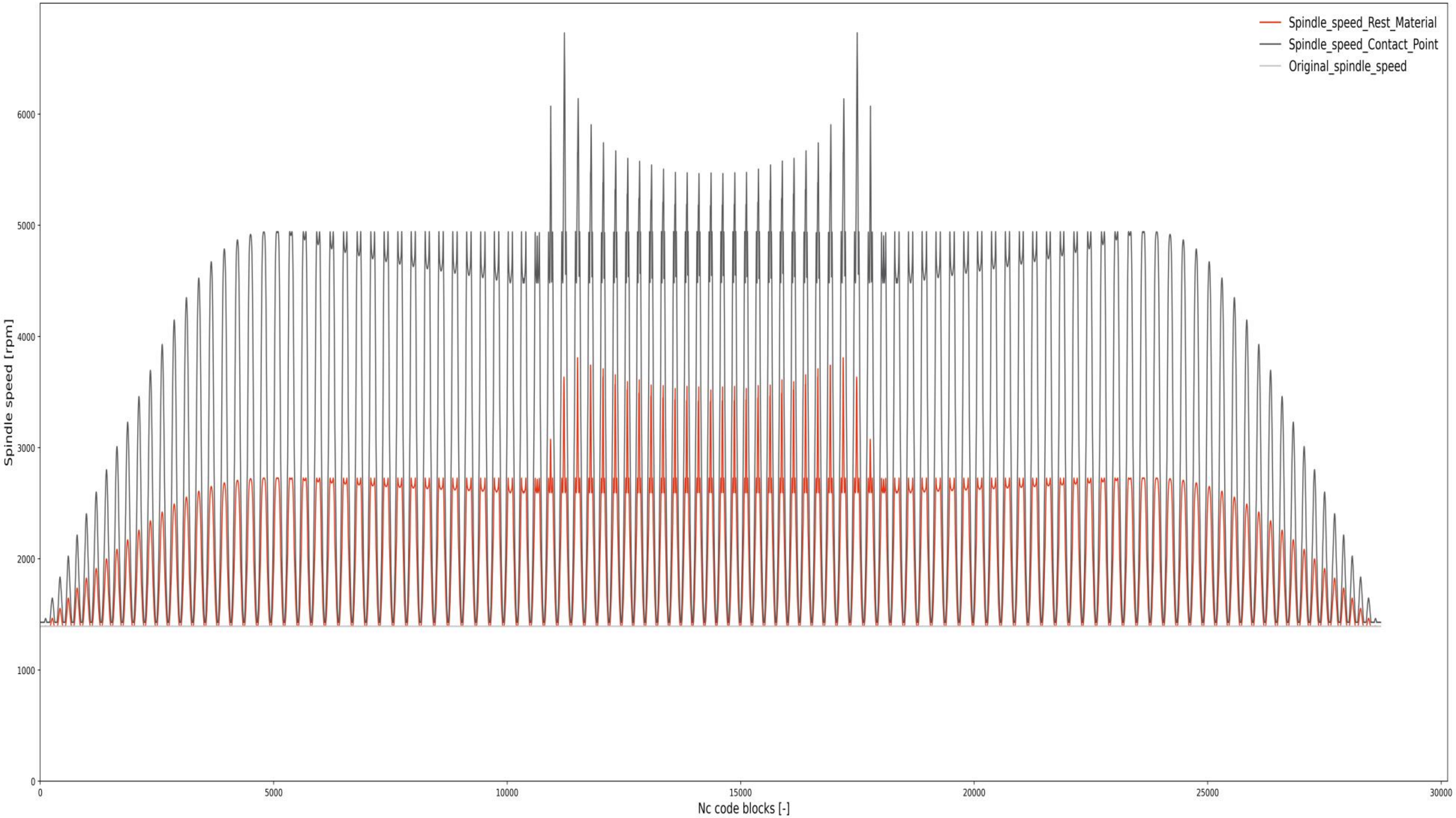
Appendix 5: Feed rate graph for the finishing operation on the demonstrative geometry using combination of both optimizations



Appendix 6: Feed rate graph for the Feed Rate Optimization verification



Appendix 7: Spindle speed graph for Cutting Speed Optimization verification



Appendix 8: Feed rate graph for Cutting Speed Optimization verification