CIRP Annals - Manufacturing Technology 70 (2021) 341–344

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: https://www.editorialmanager.com/CIRP/default.aspx

Online adaption of milling parameters for a stable and productive process

Benjamin Bergmann*, Svenja Reimer

Institute of Production Engineering and Machine Tools (IFW), Leibniz Universität Hannover, Germany

Submitted by Tojiro Aoyama (1), Tokyo, Japan

ARTICLE INFO

Article history:
Available online 19 May 2021

Keywords:
Machine tool
Chatter
Machine Learning

ABSTRACT

On the way to fully autonomous machine tools it is essential to independently select suitable process parameters and adapt them on-the-fly to the appropriate process conditions in a self-controlled manner. Such systems require complex physical process models and are usually limited to feed and spindle speed adaption during the milling process. This paper introduces a new approach enabling machines during the milling process to learn which parameters lead to a stable process with maximum productivity and to adjust them autonomously. It is shown that this approach enables the machine tool to independently find stable process parameters with maximum productivity.

1. Introduction

Self-excited chatter vibrations are one of the main limitations regarding productivity in metal cutting [1,2]. The choice of suitable process parameters for a stable and productive process is a major challenge due to the nonlinear correlation between process parameters (depth of cut a_p, width of cut a_e and spindle speed n) and the occurrence of chatter. To select appropriate process parameters, costly experiments and time-consuming simulations to obtain stability lobe diagrams (SLD) have to be carried out for each process setup [3–5]. Additionally, simplified model assumptions lead to uncertainties [6]. Time variant changes in the stability limit, e.g. due to tool wear can only be considered to a limited extent [7]. To overcome these challenges, an autonomous online optimization of the process parameters is required.

In the context of Industry 4.0, Self-optimizing machining systems (SOMS) are becoming increasingly relevant. SOMS are able to independently evaluate their current process conditions and autonomously adapt process parameters to increase productivity [8,9]. Previous works on SOMS for online chatter avoidance address the variation of cutting speed v_c to increase the process stability [10–12]. For this purpose, the dominant chatter frequency is identified by the spectrum of sensor signals, e.g. from a microphone. The spindle speed is adapted such that the chatter frequency is a multiple of the tooth passing frequency. In combination with an iterative increase of the width or depth of cut increased productivity was achieved. However, the method is limited to processes with a single chatter frequency and simple tool geometries with equal flute spacing. Further, the method for adaption of v_c only considers process stability. As soon as stable parameters are found no further adaption of the cutting speed occurs and thus the productivity is kept constant. Knowledge from previous cuts is not used in this method. SOMS are able to learn the correlation between process parameters and process stability independently during the process. Hereby stable parameters with maximal productivity could be determined for different tool geometries and without prior knowledge of the process dynamics. However, SOMS with these capabilities and adapting v_c and the width or depth of cut during the process for increasing the productivity with regard to the process stability do not exist.

A basic requirement for such a SOMS is the process parallel determination of the correlation between process parameters and process stability. Approaches for this include the process parallel determination of frequency response functions [13] and data-driven modelling of the SLD utilizing machine learning (ML) [14–16]. The experimental effort to gain training data for the ML models is time-consuming. Surrogate-based optimization allows to simultaneously train ML models in relevant areas and to optimize parameters [17]. In this work, surrogate-based optimization is used for the first time for self-learning optimization of milling parameters during the process. Based on online learning SLD (LSLD), the machine tool learns to find process parameters that lead to a stable process with maximum material removal rate Q_w and adapt them during the process. For the first time, a SOMS is presented and realized that is able to autonomously adjust the tool path and the cutting speed during milling to maximize Q_w while considering the process stability. A prerequisite of the system is the online monitoring of the process stability. As shown in [18], static process forces can be measured by strain gauges applied to the spindle slide. In this paper, a new method for chatter detection based on semi-conductor strain gauges is presented.

2. Surrogate-based optimization of process parameters

Surrogate-based optimization is a global optimization strategy that does not require any analytical description of the objective

* Corresponding author.
E-mail address: bergmann@ifw.uni-hannover.de (B. Bergmann).

https://doi.org/10.1016/j.cirp.2021.04.086
0007-8506/

function. It is common for the optimization of expensive-to-evaluate functions [17]. Since the true objective function is not known, statistical models are used as surrogate models. Iteratively, optimal values are determined by optimizing the surrogate model. The real objective function is evaluated at these values and the obtained results are used to improve the surrogate model. In the following, the feed per tooth f_z is kept constant such that Q_w can be calculated depending on a_p, a_e and n.

2.1. Objective function

Assuming a function f_{SLD}, which has positive values if chatter occurs and negative values otherwise, the optimal operation point $param_{opt} = (a_{p,opt}, a_{e,opt}, n_{opt})$ with maximum productivity can be determined by solving the optimization problem Eq. 1.

$$\begin{aligned} param_{opt} = \quad & \operatorname{argmax}_{a_p,\ a_e, n} Q_w(a_p,\ a_e,\ n), \\ s.t. \quad & -f_{SLD}(a_p,\ a_e,\ n) \geq 0 \\ & v_c(n) - v_{c,\ min} \geq 0\ \wedge v_{c,max} - v_c(n) > 0 \end{aligned} \tag{1}$$

To avoid excessive tool wear, additional limitations for the cutting speed $v_{c,min}$ and $v_{c,max}$ are specified. Further technical limitations could be added equivalently. For solving Eq. 1, the target and the technical limitations are combined into a common objective function f_{target} as a weighted linear combination (Eq. 2). For a higher generality and a better comparability of the individual terms, Q_w is normalized by its theoretically maximum. To consider the restrictions to v_c, log-barrier functions $b_{v_c}(n)$ were used (Eq. 3), which are common for constrained optimization [19]. Thus, parameters are rated less suitable the further they deviate from the recommended parameters.

$$\begin{aligned} f_{target} = & \lambda_{target} \cdot (1 - Q_w^{norm}(a_p, a_e, n)) \\ & - \lambda_{SLD} \cdot (f_{SLD}(a_p,\ a_e,\ n) + f_{safe}(a_p,\ a_e,\ n)) \\ & + \ \lambda_{v_c} \cdot b_{v_c}(n) \end{aligned} \tag{2}$$

$$\begin{aligned} b_{v_c}(n) \ = & -Re\left(\log\left(\left| v_{c,min} - v_c(n) \right| \right) \right) \\ & - Re\left(\log\left(\left| -v_c(n) + v_{c,max} \right| \right) \right) \end{aligned} \tag{3}$$

2.2. Learning stability lobe diagram

Since f_{SLD} is unknown, a LSLD is used as a surrogate model. In [15], the suitability of different ML methods and hyperparameters has already been investigated. In this paper, Regularized Kernel Interpolation (RKI) is used for LSLD since it has the best convergence rate, a high classification quality, and short training times. The RKI model is a linear combination of kernel functions k_w, centered at the individual samples of the training data (Eq. 4). The kernel parameter ε determines the expansion of the individual terms. To determine the weights $\alpha_{i,\lambda}$, the linear system of equations Eq. 5 with the kernel matrix $\boldsymbol{K}$, the regularization parameter λ and the identity matrix $\boldsymbol{I}$ is solved.

$$\begin{aligned} & f_{SLD}(\boldsymbol{x}; \lambda, \epsilon) = \sum_{i=1}^{N} \alpha_{i,\lambda} k_w(\boldsymbol{x}, \boldsymbol{x}_i; \epsilon) \\ & k_w(\boldsymbol{x}, \boldsymbol{y}; \epsilon) = \ \max(0,\ \epsilon \cdot \boldsymbol{x} - \boldsymbol{y})^2,\ \boldsymbol{x}, \boldsymbol{y} = \ (a_e, n) \epsilon \mathbb{R}_+^2 \end{aligned} \tag{4}$$

$$\boldsymbol{\alpha}_\lambda = (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \cdot \boldsymbol{y} \tag{5}$$

For simplification, only the autonomous adaption of n and a_e is considered in this paper. However, the method can also be extended to the depth of cut. The individual terms of the objective function can be interpreted as a safety zone around the unstable operating points. Since operating points with the same n and higher a_e are also probably unstable, this safety zone is additionally extended upwards to higher values of a_e (Eq. 6).

$$f_{safe}((a_e, n)) = \sum_{(a_{e,j}, n_j)\ \in X_{chat}} \frac{\min(0, a_e - a_{e,j})}{1 + (n - n_j)^2} \tag{6}$$

2.3. Autonomous parameter adaption

For the parameter optimization, the LSLD is inserted as a surrogate model into f_{target}. A stochastic gradient method with multiple starting values is used to select optimal parameters. This reduces the probability of reaching only a local minimum and the optimization for the individual start values can be carried out in parallel, to reduce the computing time. The process stability is then determined for the parameters identified by the optimization. Via this surrogate-based optimization, productivity is maximized and an increasingly accurate LSLD is created. The LSLD selects its training data independently during the process. In the following, the method is investigated with a previously untrained LSLD. If data is already available, a pre-trained LSLD can also be used to accelerate the optimization.

3. Realization of the intelligent machine tool

As depicted in Fig. 1, the system consists of a machine tool, a process-parallel data acquisition and a parameter optimization. In the following, these components are described in more detail.

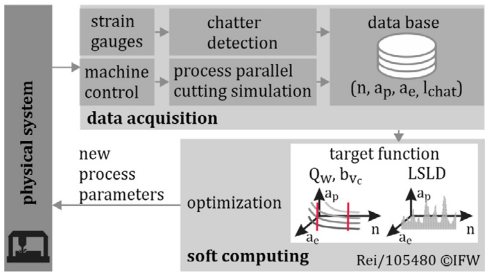

Fig. 1. System overview.

3.1. Physical system

The system was implemented on a 5-axis machine tool DMG Mori HSC 30. The machine is controlled by a Sinumerik 840D sl machine control. A Process Field Bus (Profibus) is used for communication with the machine control. Control data (axis positions, spindle speed, etc.) is written to the Profibus via the Siemens ADAS compile cycle in the interpolation cycle (4 ms) of the machine tool.

3.2. Data acquisition

To enable online learning, the process parameters and the process stability must be continuously recorded. While n can be read from the machine control, a_e and a_p must be calculated from the axis positions. For simple processes, as the face milling process examined in this paper, this can be realized in the NC code. For processes with a higher complexity, a process-parallel milling simulation based on the axis positions is used [20]. A semiconductor strain gauge on the spindle slide was used to evaluate the process stability. The sensor signals were recorded at a sampling rate of 20 kHz and filtered by an analogue low-pass filter with a cut-off frequency of 5 kHz to avoid aliasing. An accelerometer was used as a reference. The dominant chatter frequency was identified in the range of 1-4 kHz, therefore 5 kHz were considered sufficient for subsequent analysis. The once-per-revolution variance var_{SG} [21] was used for chatter detection, because it requires only a small computation time and is therefore well suited for a real-time system. To determine the statistical limits b_{chat}, experiments with four different feeds and cutting speeds were performed in Al7075 and AISI 1045. The courses of var_{SG} and b_{chat} for a ramp milling process are shown in Fig. 2.

In the investigations, tools with a maximum width of flank wear (VBmax = 15 μm) were used. Width and depth of cut were increased

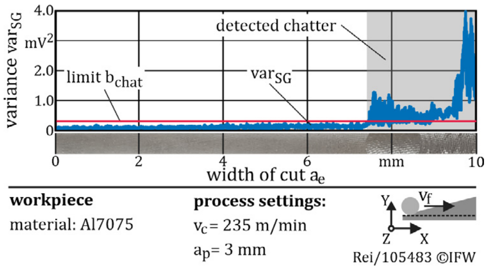

Fig. 2. Chatter detection with semi-conductor strain gauges.

in steps of 0.1 mm. Constant cutting parameters were used for each step to avoid difficult to classify transition areas, as in ramp milling. To increase the sensitivity, b_{Chat} was chosen as the maximum value of the processes without visible chatter marks instead of a 6σ-limit. Further experiments with a worn tool (VB = 140 μm) were performed in AISI 1045 to investigate the influence of tool wear. On average, var_{SG} was higher with the worn tool in stable areas and lower in unstable areas due to higher damping. Nevertheless, the stable and unstable processes could be clearly separated by b_{chat}. Chatter occurring for process parameters near the stability limit, respectively in the semi-stable area, is in general less pronounced than chatter occurring far outside the stable range. This is also reflected by the amplitude of var_{SG}. To address this constraint, the LSLD is trained with the chatter score l_{chat} from Eq. 7 as a regression task. Due to the negative value of l_{chat} at stable operating points, these parameters are preferred during optimization. To avoid a premature stagnation of the algorithm, the value of l_{chat} in the stable range has to be weighted lower than in the unstable range. In a series of experiments ($\epsilon_{\text{stab}} = 1, 0.5, 0.1$) ϵ_{chat} = 1 and ϵ_{stab} = 0.1 proved to be suitable.

$$l_{chat} = \begin{cases} \epsilon_{\text{stab}} \cdot (var_{norm} - 1), & var_{SG} < b_{chat} \\ \epsilon_{\text{chat}} \cdot (var_{norm} - 1), & var_{SG} \geq b_{chat} \end{cases}$$

$$var_{norm} = \frac{var_{SG}}{b_{chat}}$$

3.3. Soft computing

The number of computational operations for training the LSLD and for the optimization cannot be predicted deterministically. Therefore, these modules are implemented outside of the real-time system as a “soft computing” system with Python. As soon as new process information is available, the training of the LSLD is triggered. In contrast to offline learning where model parameters can be selected based on the available amount of data, in online learning they have to be adapted to the current data situation. To ensure continuous adaption of the models to the data density, the kernel parameter ϵ is reselected at each learning step by the fill distance [15]. If the unstable areas are insufficiently known, the optimization would select high values for a_e to maximize productivity, which can result in chatter. Thus, a limit to the maximum step size Δa_e is introduced.

4. Process parallel parameter adaption

NC programs can only be changed to a limited extent during the process. For a process parallel change of a_e and a_p, preparation of the NC code is necessary. In Fig. 3, the prepared NC code for a slot milling process is shown. While v_c can be changed on the entire tool path, a_e may only be adapted after a completely machined tool path. This is

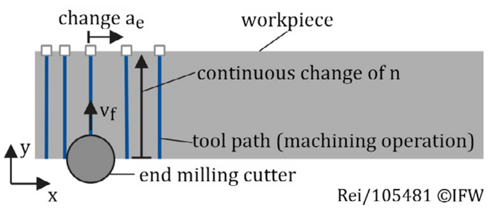

Fig. 3. Parameter adaption while the test process.

realized by synchronous actions and querying new a_e after each tool path. To avoid overfitting in the LSLD, a simultaneous optimization of all parameters is reasonable. Otherwise, by continuously adjusting v_c during machining, more process information can be obtained. Two adaption strategies can thus be selected: a) only simultaneous change of a_e and n, b) continuous change of n and change of a_e at special adaption points.

5. Experimental investigations

5.1. Experimental setup

Experimental cutting tests were carried out on the machining centre HSC 30 in milling operations without coolant. To show the general validity of the approach different materials (aluminum Al7075 and steel AISI 1045) were investigated. All experiments were carried out with initial flank wear of < 15 μm. After optimization, no significant change in flank wear could be observed. The experiments have been conducted with a solid carbide endmill with four teeth fixed in a shrinking chuck. The depth of cut was kept constant at 3 mm. The process parameters are listed in Table 1. As initial values for the cutting speed, the process parameters recommended by the tool manufacturer were used. Experimental SLD were created to evaluate the optimization. For this purpose, ramps with increasing width of cut and different cutting speeds (step size 200 1/min) were machined.

Table 1
Experimental design

		Al7075	AISI 1045
end milling cutter	tool diameter D	10 mm	8 mm
	tool length L	72 mm	63 mm
	helix angle	45°	36°/38°
	coating	TiAIN	AlTiN
ref. values	f_z	0.04 mm	0.045 mm
	v_c	300 m/min	180 m/min
start values	$a_{e,start}$	0.1 mm	0.1 mm
optimization parameters	constraint v_c	+/- 20%	+/- 20%
	Δa_e	0.5 mm	0.5 mm

5.2. Influence of the optimization parameters

The kernel parameter ϵ and the regularization parameter λ of the LSLD are crucial for the optimization result. Large values for λ or small values for ϵ lead to a fast learning progress, but also to a smoothing of the learned stability limit so that small peaks may not be detected. In experimental investigations, $\epsilon = 1$ and $\lambda = 4$ have been proven to be suitable. The weights λ_{target}, λ_{SLD} and λ_{v_c} in f_{target} are further relevant. Too small values of λ_{target} lead to a prematurely stagnation. Too large values of λ_{target} in relation to λ_{SLD} can lead to increased chatter. In experiments with previously recorded SLD, λ_{target}=10 and λ_{SLD}= 20 have proven to be suitable. If the optimization stagnates and the dynamic behavior does not change significantly, there is no motivation for the algorithm to further adapt the parameters. To enforce further learning, λ_{target} or the kernel parameter ϵ can be increased.

5.3. Results

The optimization results of both strategies are shown in Fig. 4 and Fig. 5. For all processes, the productivity has been increased in less than 20 milling paths by more than 450%. For Al7075, both strategies increased Q_w from 458 mm^3/min to about 50,000 mm^3/min. For AISI 1045, an increase in Q_w from 387 mm^3/min to 21,070 mm^3/min with strategy a) and to 22,860 mm^3/min with strategy b) has been achieved. Chatter occurred two times with strategy a) and four times with strategy b). Due to the continuous adaption of v_c with strategy b), the time periods with chatter are smaller than with strategy a). As soon as chatter is detected, v_c is adapted to stable parameter ranges. Only if no stable point is found, long chatter phases occur until a_e can be

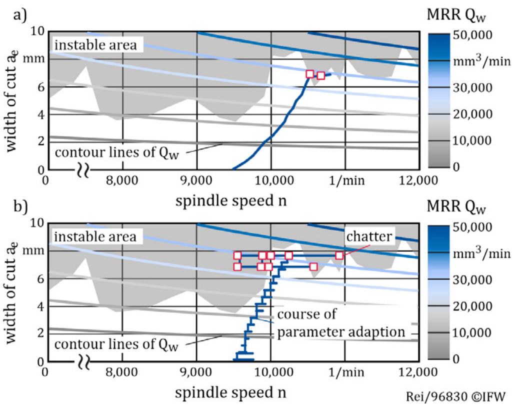

Fig. 4. Results of the two optimization strategies for Al7075: a) simultaneous change of a_e and n, b) continuous change of.

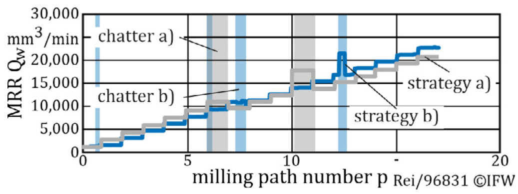

Fig. 5. Results of the two optimization strategies for AISI 1045: a) simultaneous change of a_e and n, b) continuous change of.

reduced again. With strategy a), less data is available for training the LSLD. Investigations with $a_{e,start}$ = 1 mm showed significant chatter at the end of the optimization with strategy a) because cutting speeds were selected in parameter ranges that have not yet been tested. To compare the determined operation points of both strategies, measurements of the surface roughness were performed. For Al7075 a surface roughness of R_z = 7.9 μm for strategy a) and R_z = 7.6 μm for strategy b) results. This high roughness values can be traced back to the increased adhesion at maximum Q_W due to absence of coolant. The surface with visible chatter marks has a surface roughness of R_z = 14.4 μm.

6. Conclusion and Outlook

This paper provides a new method for the autonomous online optimization of milling parameters with regard to process stability and productivity. It was shown, that the machine tool can autonomously increase productivity by an online adaption of n and a_e based on a self-learning algorithm. By applying the method to two different materials, the general applicability was demonstrated. In contrast to analytical methods, the approach does not require any prior knowledge or complex physical models. Therefore, time and cost-consuming determination of the dynamic behavior is unnecessary and errors due to model uncertainties are avoided. During the experiments for this paper, no further adjustment of the optimization parameters was necessary for the different workpiece materials. This shows the generality of the chosen hyperparameters. Continuous learning allows the consideration of time-varying changes.

However, due to the stochastic nature of the optimization, convergence to the global optimum is not assured. In contrast to analytically calculated SLD, LSLD requires data from stable and unstable processes for learning. Thus, the stability limit may be exceeded during optimization. Even if the process can be stabilized rapidly by an adaption of v_c, the surface quality may already be affected. For practical application, the optimization should be performed on non-critical parts of the workpiece. A stock allowance can be defined for which the optimization is disabled. If knowledge about the dynamic behavior is already available an augmented strategy could be used in the future, where the LSLD are pre-trained by analytically determined SLD. Thus, the training of the LSLD can be accelerated and model uncertainties and time-varying changes can be considered. For a higher performance a_p can also be added as additional parameter in the LSLD and thus be considered in the optimization. Although online path adaption via adaption points in the NC code works for many standard 3-axis milling operations, further research is required for more complex geometries. One limitation is that unrecognized process influences, such as chatter due to thin-walled workpieces, can lead to inaccurate training data. To avoid contour violations, additional limitations for a_p and a_e must be defined. For practical application further technical limitations, such as a maximum torque, must also be considered.

Technical limitations for predictable values can be added equivalent to Eq. 3, while additional physical or statistical models are needed for unknown values.

Further research addresses the consideration of additional technical limitations and their impact on the optimization. Furthermore, the choice of optimal optimization parameters depending on the particular practical requirements and of prior knowledge will be investigated.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors thank the German Research Foundation (DFG) for its financial and organizational support of the project DE 447/156-1.

References

[1] Munoa J, Beudaert X, Dombovari Z, Altintas Y, Budak E, Brecher C, Stepan G (2016) Chatter suppression techniques in metal cutting. *Annals of the CIRP* 65(2):785–808.
[2] Altintas Y, Weck M (2004) Chatter Stability of Metal Cutting and Grinding. *Annals of the CIRP* 53(2):619–642.
[3] Honeycutt A, Schmitz TL (2018) Milling Bifurcations: A Review of Literature and Experiment. *Journal of Manufacturing Science and Engineering* 140(12):120801.
[4] Brecher C, Chavan P, Epple A (2018) Efficient determination of stability lobe diagrams by in-process varying of spindle speed and cutting depth. *Advances in Manufacturing* 6(1):272–279.
[5] Budak E, Tekeli A (2005) Maximizing Chatter Free Material Removal Rate in Milling through Optimal Selection of Axial and Radial Depth of Cut Pairs. *Annals of the CIRP* 54(1):353–356.
[6] Löser M, Otto A, Ihlenfeldt S, Radons G (2018) Chatter prediction for uncertain parameters. *Advances in Manufacturing* 6(2):319–333.
[7] Liu Y, Wang Z, Liu K, Zhang Y (2017) Chatter stability prediction in milling using time-varying uncertainties. *International Journal of Advances Manufacturing Technology* 89:2627–2636.
[8] Möhring HC, Wiederkehr P, Erkorkmaz K, Kakinuma Y (2020) Self-optimizing machining systems. *Annals of the CIRP* 69(2):740–763.
[9] Monostori L, Kádár B, Bauernhansl T, Kondoh S, Kumara S, Reinhart G, Sauer O, Schuh G, Sihn W, Ueda K (2016) Cyber-physical systems in manufacturing. *Annals of the CIRP* 65(2):621–641.
[10] van Dijk NJM, Doppenberg EJJ, Faassen RPH, van de Wouw N, Oosterling JAJ, Nijmeijer H (2010) Automatic In-Process Chatter Avoidance in the High-Speed Milling Process. *Journal of Dynamic Systems Measurement and Control* 132(3):031006.
[11] Delio T, Smith S, Tlusty J (1992) Use of Audio Signals for Chatter Detection and Control. *ASME Journal of Engineering for Industry* 114(2):146–157.
[12] Smith S, Tlusty J (1992) Stabilizing Chatter by Automatic Spindle Speed Regulation. *Annals of the CIRP* 41(1):405–409.
[13] Abele E, Fiedler U (2004) Creating Stability Lobe Diagrams during Milling. *Annals of the CIRP* 53(1):309–312.
[14] Friedrich J, Torzewski J, Verl A (2018) Online Learnings of Stability Lobe Diagrams in Milling. *Procedia CIRP* 67:278–283.
[15] Denkena B, Bergmann B, Reimer S (2020) Analysis of different machine learning algorithms to learn stability lobe diagrams. *Procedia CIRP* 88:282–287.
[16] Cherukuri H, Perez-Bernabeu E, Selles MA, Schmitz TL (2019) Machining Chatter Prediction Using a Data Learning Model. *Journal of Manufacturing and Materials Processing* 3(2):45.
[17] Forrester AIJ, Keane AJ (2009) Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences* 45(1-3):50–79.
[18] Denkena B, Boujnah H (2018) Feeling machines for online detection and compensation of tool deflection in milling. *Annals of the CIRP* 67(1):423–426.
[19] Luenberg DG (2016) *Penalty and barrier methods.* Linear and Nonlinear Programming, Springer, 397–428.
[20] Dittrich MA, Uhlich F, Denkena B (2018) Self-Optimizing Tool Path Generation for 5-Axis Machining Processes. *Journal of Manufacturing Science and Technology* 24:49–54.
[21] Schmitz TL, Medicus K, Dutterer B (2002) Exploring one-per-revolution audio signal variance as a chatter indicator. *Machining Science and technology* 6(2):215–233.