



TECHNISCHE  
UNIVERSITÄT  
WIEN



DIPLOMARBEIT

---

# Machine Learning: Identifikation von Prozessinstabilitäten beim Fräsen

---

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

**Diplom-Ingenieurs**

unter der Leitung von

**Univ.-Prof. Dipl.-Ing. Dr.techn. Friedrich BLEICHER**

(Institut für Fertigungstechnik und Photonische Technologien)

und unter der Betreuung durch

**Dipl.-Ing. Paul SCHÖRGHOFER**

**Dipl.-Ing. Dr.techn. Norbert LEDER**

**Dipl.-Ing. Christoph RAMSAUER**

eingereicht an der Technischen Universität Wien

**Fakultät für Maschinenwesen und Betriebswissenschaften**

von

**Ralf OSWALD**



Wien, im Oktober 2019

---

Ralf Oswald, BSc



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Eidesstattliche Erklärung

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

DIPLOMARBEIT

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiters eidesstattlich, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe.

Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Kurzfassung

Diese Arbeit baut auf der Entwicklung des Sensorischen Werkzeughalters auf. Dieser Werkzeughalter ist unter anderem mit einem Beschleunigungssensor ausgestattet und wurde von der Firma myTool IT und dem Institut für Fertigungstechnik und Photonische Technologien entwickelt. Nun ist es entscheidend, wie mit den gemessenen Beschleunigungsdaten verfahren wird, um Aussagen über den Fräsprozess zu treffen und diesen zu optimieren. Das Hauptaugenmerk dieser Arbeit liegt dabei auf dem Unterscheiden von stabilen und instabilen Prozessen mithilfe von selbst erstellten Stabilitätsindikatoren. Dafür wird zuerst die Theorie der Prozessstabilität beim Fräsen mit Hilfe der Stabilitätskarten erläutert, um tiefere Einblicke in die Thematik und ein besseres Verständnis für die Stabilität des Fräsprozesses zu erlangen. In weiterer Folge werden Fräsversuche durchgeführt. Die Versuche werden mit den zuvor erstellten Stabilitätskarten verglichen und in Einklang gebracht. Bei diesen Versuchen wird auch der Luftschall gemessen, welcher es ermöglicht, die Akustik des Prozesses zu späteren Zeitpunkten noch einmal wahrzunehmen. Neben den Schallmessungen wird außerdem mit Bildaufnahmen die Qualität der Oberfläche dokumentiert. Die Akustik und Oberflächen der Fräsversuche werden analysiert und dienen als Hilfe zur Klassifikation der Stabilität. Die einzelnen Stabilitätsklassen sind für die Trainingsdaten der Machine Learning Algorithmen entscheidend. Die Beschleunigungsdaten dienen zum Erstellen von Stabilitätsindikatoren, welche sowohl aus dem Zeitbereich, als auch dem Frequenzbereich gewonnen werden. Durch die Bewertung der ermittelten Indikatoren nach unterschiedlichsten Kriterien, werden nur die signifikantesten unter ihnen für das Machine Learning verwendet. Folgend wird ein neuronales Netz trainiert. Die damit geschaffene künstliche Intelligenz ist in der Lage, anhand des Sensorsignales zu bewerten, ob der aktuell vorliegende Prozess stabil oder instabil ist. Durch einen zuvor durchgeführten Impulshammerversuch ist eine Abschätzung dahingehend möglich, wie weit die Drehzahl, bei Vorliegen eines instabilen Prozesses, erhöht beziehungsweise vermindert werden muss, damit sich der Prozess stabilisiert.

Diese Arbeit dient als Grundstein für weitere Überlegungen zur Prozessoptimierung mit Hilfe des Sensorischen Werkzeughalters. Dies könnte unter anderem die Implementierung eines Systems sein, welches selbst anhand von Beschleunigungsdaten eines Fräsprozesses eigenständig lernt und damit die manuelle Feature Extraction und Feature Selection obsolet wird. Der Sensorische Werkzeughalter könnte hier als Subkomponente in einem kognitiven System gesehen werden, welches aus einer Vielzahl an Sensoren besteht.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Abstract

This work is based on the development of the Sensory Tool Holder, which was developed by the company myTool IT cooperating with the Institute of Production Engineering and Photonic Technologies. This diploma thesis should show how the measured data has to be treated to predict the stability of the milling process with the help of machine learning algorithms.

The main task is to demonstrate the classification of milling processes with self-made stability indicators. The first part, is a comprehensive research about the theory of stability lobe diagrams to get a much deeper insight and better understanding of milling instabilities. The next step is to carry out tests, compare them to the stability lobe diagram and adapt the diagram. The machine learning algorithms require classified training and testing data. Sound measurements and pictures of the workpiece surface are recorded, to support the classification of the data in the right way. From the acceleration signal of the Sensory Tool Holder, stability indicators are created, which have their origin of the time domain and the frequency domain. After the creation of indicators, they get rated and the most significant get picked. With the right indicators and an adequate amount of training data, the neural network can be trained. Based on the Sensory Tool Holder signal, this artificial intelligence is able to classify the process into the classes stable or unstable. How much the spindle speed should be increased or decreased to stabilize the process will be estimated by a tap test of the machine structure. This work is an important foundation for the optimization of the milling process with the Sensory Tool Holder. Subsequently, the machine learning algorithm can be improved in order to skip the manual feature extraction and feature selection to create a self-learning algorithm. The Sensory Tool Holder function as a subsystem of a cognitive system, consisting of multiple sensors, for example temperature or force sensors, to monitor and control the process in the best way.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

## *Vorwort und Dank*

Als Erstes will ich hier die Möglichkeit nutzen, um meinen Freunden und meiner Familie, insbesondere meinen Eltern, für die Unterstützung während des gesamten Studiums zu danken. Ebenso will ich meiner Freundin Lisa danken, welche mir immer eine große Stütze war und mich meine gesamte Studienzeit begleitet hat.

Weiterer Dank gebührt meinen (ehemaligen) Arbeitskollegen Dr. Benjamin Mörzinger und DI Thomas Weiler, welche es mir ermöglicht haben, am IFT zu arbeiten und meine ersten wissenschaftlichen Erfahrungen zu sammeln. Ebenso sind mir diese Kollegen des Öfteren mit gutem Rat beiseite gestanden und haben zum Gehalt dieser Arbeit indirekt beigetragen.

Direkte Unterstützung erfuhr ich dankenderweise von meinen Betreuern Dr. Norbert Leder, DI Paul Schörghofer, und DI Christoph Ramsauer. Sowohl im Bereich des Maschinenbaus, als auch im Bereich der Signalauswertung und Signalverarbeitung konnte ich immer auf deren Unterstützung bauen. Auch das Eingehen auf meine Vorschläge und den Entfaltungsspielraum, welchen ich bekam, empfand ich als große Wertschätzung.

Zu guter Letzt will ich mich beim gesamten Personal des Instituts für Fertigungstechnik und Photonische Technologien, insbesondere bei meinem Hauptbetreuer Prof. Friedrich Bleicher, dafür bedanken, dass mir das Schreiben der Arbeit in dieser Form ermöglicht wurde.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Inhaltsverzeichnis

<b>Eidesstattliche Erklärung</b>	<b>i</b>
<b>Kurzfassung</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Vorwort und Dank</b>	<b>vii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Forschungsfragen und Ziel . . . . .	2
1.2 Motivation für diese Arbeit . . . . .	3
1.3 Arbeits- und Herangehensweise . . . . .	3
1.4 System Sensorischer Werkzeughalter . . . . .	3
1.4.1 Voruntersuchung - Drehzahlabhängigkeit des Signals . . . . .	4
1.5 Prozessinstabilitäten beim Fräsen . . . . .	7
1.6 Potenzialanalyse . . . . .	7
<b>2 Stabilitätskarten</b>	<b>11</b>
2.1 Allgemein . . . . .	11
2.2 Stabilitätsalgorithmus von Altintas und Budak . . . . .	13
2.2.1 Erstellen von Stabilitätskarten für zwei Freiheitsgrade . . . . .	13
2.2.2 Stabilitätsalgorithmus für einen Freiheitsgrad . . . . .	19
2.2.3 Superposition verschiedener Eigenformen . . . . .	20
2.3 Verifikation des Modells . . . . .	21
2.3.1 Vergleich der Ergebnisse des abgeleiteten Modells mit jenen aus einer Parameterstudie . . . . .	21
2.3.2 Vergleich mit Software CutPro . . . . .	23
2.4 Schlussfolgerungen aus dem Stabilitätsalgorithmus . . . . .	24
2.4.1 Drehzahlbereiche bei maximaler Grenzschnitttiefe . . . . .	24
2.4.2 Drehzahlbereiche bei minimaler Grenzschnitttiefe . . . . .	25
2.4.3 Weitere Überlegungen . . . . .	26
2.5 Erkenntnisse aus den Stabilitätskarten für die Regelung . . . . .	27
<b>3 Versuche</b>	<b>29</b>
3.1 Versuchsaufbau und Vorversuche . . . . .	29
3.1.1 Aufbau und verwendete Maschinen . . . . .	29
3.1.2 Bestimmung der Eigenfrequenzen mittels Impulshammerversuch	31

3.1.3	Abschätzung von Eigenfrequenz und Gesamtsteifigkeit des Systems . . . . .	31
3.1.4	Erstellung einer Versuchs-Stabilitätskarte . . . . .	33
3.2	Versuchsdurchführung . . . . .	33
3.3	Ergebnisse und Auswertung . . . . .	34
3.3.1	Klassifikation von stabilen und instabilen Prozessen . . . . .	37
3.3.2	Gegenüberstellung von Signalen im Zeitbereich . . . . .	38
3.3.3	Gegenüberstellung von Signalen im Frequenzbereich . . . . .	42
<b>4</b>	<b>Erstellung und Auswahl von Stabilitätsindikatoren</b>	<b>45</b>
4.1	Feature Extraction im Zeitbereich . . . . .	45
4.1.1	Zentrale Tendenz: Arithmetisches Mittel, Median, Quantilen . . .	46
4.1.2	Streuungsmaße: Varianz, Interquartilsabstand . . . . .	46
4.1.3	Gestaltmaße: Schiefe, Wölbung . . . . .	47
4.1.4	Störmaße: Entropie . . . . .	47
4.2	Feature Extraction im Frequenzbereich . . . . .	47
4.3	Feature Selection . . . . .	51
<b>5</b>	<b>Machine Learning</b>	<b>55</b>
5.1	Einführung zu künstlicher Intelligenz . . . . .	55
5.1.1	Machine Learning . . . . .	56
5.1.2	Supervised Machine Learning . . . . .	57
5.1.3	Deep Learning . . . . .	58
5.2	Anwendung von ML: Herzton-Klassifizierung . . . . .	59
5.3	Erstellung des neuronalen Netzes . . . . .	60
5.3.1	Training und Validierung . . . . .	61
5.4	Analyse der falsch klassifizierten Messreihen . . . . .	63
5.4.1	Versuch 1040 (false-positive / Fehler 1.Art) . . . . .	64
5.4.2	Versuch 1170 (false-positive / Fehler 1.Art) . . . . .	65
5.4.3	Versuch 2060 (true-negative / Fehler 2.Art) . . . . .	65
5.5	Verwendung des Modells . . . . .	66
<b>6</b>	<b>Conclusio und Ausblick</b>	<b>67</b>
6.1	Beantwortete Forschungsfragen . . . . .	67
6.1.1	Gefundene Stabilitätsindikatoren . . . . .	68
6.1.2	Implementierter Algorithmus . . . . .	68
6.2	Ausblick . . . . .	68
6.2.1	Automatisiertes Machine Learning . . . . .	69
6.2.2	Kognitive Systeme . . . . .	69
	<b>Literatur</b>	<b>71</b>
	<b>Abbildungsverzeichnis</b>	<b>73</b>

<b>Tabellenverzeichnis</b>	<b>76</b>
<b>Abkürzungsverzeichnis und Physikalische Konstanten</b>	<b>79</b>
<b>A Python Code</b>	<b>83</b>
A.1 Berechnung und Plotten der Stabilitätskarte . . . . .	83
A.2 Brechnung und Plotten des Übertragungsfrequenzgangs . . . . .	89
A.3 Selbsterstellte Funktionen . . . . .	91
A.4 Datenauswertung: Zeitreihenplots und Spektrumplots . . . . .	97
A.5 Feature Extraction . . . . .	100
A.6 Feature Selection . . . . .	102
A.7 Erstellen des Neuronalen Netzwerks . . . . .	105
A.8 Nutzen des Neuronalen Netzwerks . . . . .	107
<b>B Versuchsprotokoll</b>	<b>109</b>
<b>C Fräserdatenblatt</b>	<b>115</b>



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

## Kapitel 1

# Einleitung

Um die Qualität des NC-gestützten Fräsprozesses zu gewährleisten, müssen Prozessinstabilitäten vermieden beziehungsweise klein gehalten werden. Es gibt mehrere Möglichkeiten diese Instabilitäten zu identifizieren. Es werden dafür Schnittkraft, Leistung, Schalldruck, Vibrationen am Spindelgehäuse oder Vibrationen an der Einspannung des Werkstücks, etc. gemessen und ausgewertet [Kuljanic, Totis und Sortino, 2009]. Das Problem bei diesen Methoden ist, dass Messstelle und Wirkstelle zueinander variabel oder schwach gekoppelt [Bleicher, 2019a] sind und ein Zusammenhang zwischen Messwert und Stabilität schwerer herzustellen ist. Aus diesem Grund wurde in Zusammenarbeit von myTool IT GmbH und dem Institut für Fertigungstechnik und Photonische Technologien der Technischen Universität Wien ein sensorunterstützter Werkzeughalter entwickelt, welcher Beschleunigungen im Halter misst. Dieser Sensorische Werkzeughalter ist Teil eines geschlossenen Regelkreises, der auf Prozessinstabilitäten rückschließt und aktiv in den Fertigungsprozess eingreift, um die Bearbeitungsparameter Vorschub und Drehzahl zu adaptieren. Aufgrund dieser Entwicklung ist es jetzt möglich die In-Prozess-Regelung bei der Fräsbearbeitung sinnvoll umzusetzen. Ziel ist es, für dieses bestehende Setup eine Erkennung von Instabilitäten zu implementieren und eine geeignete Regelungsstrategie zu finden. Deshalb sollen Indikatoren ermittelt werden, welche das Stabilitätsverhalten des Prozesses charakterisieren. Unter Berücksichtigung anderer Effekte, wie zum Beispiel übermäßiger Verschleiß oder Überschreiten der Leistungsgrenze der Maschine, sollen in weiterer Folge die optimalen Fertigungsparameter an die NC-Steuerung der Fräsmaschine übergeben werden, um somit den Prozess zu optimieren. Das Paradebeispiel für Instabilitäten im Fräsprozess ist das regenerative Rattern. Entsprechend der Literatur bezüglich Stabilitätskarten ist es möglich, die auftretenden Schwingungen durch das Erhöhen oder Vermindern von Vorschubgeschwindigkeit und Drehzahl abklingen zu lassen und in einem stabilen Bereich zu arbeiten. Dieser Parameter-Bereiche lässt sich nutzen, wenn Schnittdaten unter Kenntnis der Stabilitätskarten so gewählt werden, um das Zeitspanvolumen zu optimieren [Quintana und Ciurana, 2011]. Ohne Kenntnis über die Lage der stabilen Bereiche ist der Arbeitspunkt unter Umständen weit unter der Grenzschnitttiefe, was bedeutet, dass das Rattern zur Gänze vermieden wird, jedoch das Zeitspanvolumen nicht zwingenderweise optimal ist und vorhandenes Potential ungenutzt bleibt. Ebenso könnte der Arbeitspunkt oberhalb der Grenzschnitttiefe sein

und es treten Rattereffekte auf. Die Grenzschnitttiefe ist als Grenze zwischen stabilen und instabilen Bereich in der Stabilitätskarte definiert (siehe Abbildung 1.1 - blau). Ziel dieser Arbeit ist es, einen Algorithmus zu formulieren, der es ermöglicht, auf Basis von Messungen mit dem Sensorischen Werkzeughalter knapp unter der Grenzschnitttiefe im stabilen Bereich zu arbeiten. Durch den geschlossenen Regelkreis soll es im besten Fall zu einer Steigerung der Produktqualität, zur Erhöhung der Standzeit des Fräswerkzeuges und zu einer Minimierung der Bearbeitungszeit führen.

## 1.1 Forschungsfragen und Ziel

- Ist es möglich mit Hilfe der Beschleunigungsdaten des Sensorischen Werkzeughalters die Stabilität des Fräsprozesses zu charakterisieren, um so eine analytisch berechnete Stabilitätskarte zu verifizieren? Wie genau stimmen die ausgewerteten Daten mit den theoretischen Funktionen überein?
- Kann anhand eines mechanischen Modells des Fertigungsprozesses, in Verbindung mit einer Vibrationsmessung durch den Sensorischen Werkzeughalter, das auftretende Rattern prognostiziert und wie in Abbildung 1.1 mit Senkung der Drehzahl und der Vorschubgeschwindigkeit der Prozess stabilisiert werden?
- Kann diese Stabilisierung des Prozesses auch bei einer Steigerung der Drehzahl und Vorschubgeschwindigkeit erreicht werden, um in weiterer Folge die Bearbeitungszeit unter stabilen Schnittverhältnissen zu minimieren?

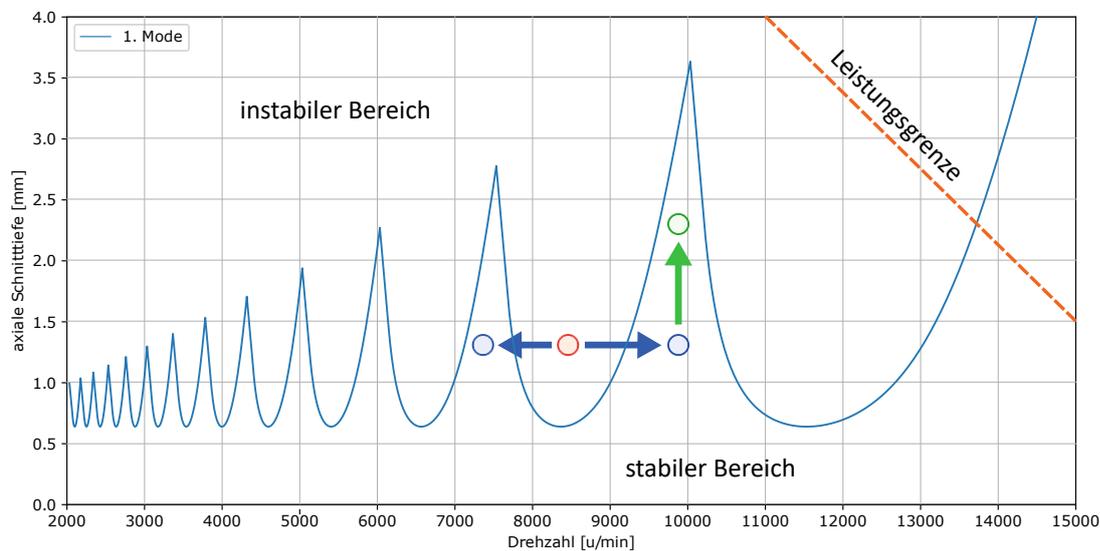


ABBILDUNG 1.1: Optimale Prozessregelung

- Ist es möglich anhand von Qualitätsindikatoren einen Rückschluss zu ziehen, in welchem Maß die Bearbeitungsparameter Vorschub und Drehzahl verschoben werden sollten, um höheres Zeitspanvolumen bei gleichzeitig stabilen Schnittverhältnissen zu realisieren?

## 1.2 Motivation für diese Arbeit

Das System des Sensorischen Werkzeughalters war zu Beginn dieser Arbeit schon sehr ausgereift. Die einachsige Beschleunigungsmessung, die Datenübertragung und der Eingriff in die NC-Steuerung zum Steuern von Drehzahl und Vorschub waren bereits implementiert. Es gab bis zu diesem Zeitpunkt wenig ausschlaggebende Kriterien zur Charakterisierung der Stabilität des Fräsprozesses und somit auch keine adäquate Form der Regelung. Eine 3-achsige Beschleunigungsmessung wurde im Laufe der Arbeit vom Team der myTool IT ebenfalls implementiert. Der entwickelte Algorithmus zur Klassifikation der Stabilität soll Machine Learning basierend sein.

## 1.3 Arbeits- und Herangehensweise

Um Wissen dieser Arbeit bestmöglich zu teilen, wird versucht möglichst viel mit Open-Source Software zu arbeiten. Weiters findet sich der Quellcode der Skripte im Anhang. Bei den verwendeten Programmen handelt es sich unter anderem um:



- Spyder → IDE zum Programmieren mit Python
- JabRef → Literaturverwaltungsprogramm
- Notepad++ → Texteditor mit Syntax-Highlighting
- Overleaf → Online Plattform zum Schreiben in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
- GIMP/InkScape → Grafikprogramme

## 1.4 System Sensorischer Werkzeughalter

Der schon zuvor angesprochene Sensorische Werkzeughalter ist nur ein einziges Element aus einem gesamten Echtzeit-Prozessregelsystem. Das Gesamtsystem besteht aus denen in Abbildung 1.2 gezeigten Komponenten.

- Signal Processing Unit (SPU)  
Die Signal Processing Unit ist für die Signalverarbeitung und die in-Prozess-Regelung zuständig. Es werden die hier in dieser Arbeit entwickelten Stabilitätsindikatoren berechnet und über die Prozessstabilität entschieden, um so Drehzahl und Vorschub an die gegebene Situation anzupassen.



ABBILDUNG 1.2: Übersicht des ICOTRONIC Systems [My Tool IT GmbH, 2019]

- **Stationary Transceiver Unit (STU)**  
Das vom Sensorischen Werkzeughalter via drahtloser Datenübertragung gesendete Signal wird von der STU empfangen und per Kabel an die SPU geleitet. Das ist notwendig, da die in sich gekapselte Fräsmaschine ähnlich wie ein faradayscher Käfig wirkt und das Bluetooth-Signal nahezu komplett abschottet. Die STU befindet sich demnach notwendigerweise im Bearbeitungsraum der Maschine.
- **Sensory Tool Holder (STH)**  
Der Werkzeughalter selbst ist ein genormter Hohlschaftkegel-Werkzeughalter, im Fall dieser Arbeit ein HSK-63-L100, der aus drei Hauptkomponenten besteht. Die erste Komponente ist der Beschleunigungssensor, der die Schwingungen während des Fräsprozesses misst. Zu Beginn der Arbeit handelte es sich um einen 1-achsigen Sensor, der radial zur Drehachse verbaut war. Im späteren Verlauf der Arbeit wurden auch Versuche mit dem weiterentwickelten Halter mit 3-achsiger Sensorik durchgeführt. Die zweite Komponente ist das Funkmodul mit dem die gemessenen Beschleunigungen an die STU via Bluetooth (BLE) gesendet wird. Als dritte und letzte Hauptkomponente ist die Akku-Einheit zu nennen, welche die beiden zuvor genannten Komponenten mit Energie versorgt. Diese Akku-Einheit kann über die Charging Cradle geladen werden.
- **Charging Cradle (CC)**  
Zum Aufladen des Werkzeughalters ist eine Ladestation vorhanden, welche für das Laden von unterschiedlichen Werkzeughaltern (z.B.: HSK-63, HSK-100; etc.) einfach adaptiert werden kann.

#### 1.4.1 Voruntersuchung - Drehzahlabhängigkeit des Signals

Der Beschleunigungssensor im Sensorischen Werkzeughalter kann aufgrund von Fertigungstoleranzen und dem Aufbau des Sensors selbst, nur mit begrenzter Genauigkeit fluchtend mit der Drehachse verbaut werden. Aus diesem Grund ist das Sensorsignal in den radialen Richtungen abhängig von der drehzahlabhängigen Zentripetalbeschleunigung. Durch eine Schräglage des Sensors könnte auch die z-Achse, welche mit der Drehachse fluchten sollte, eine Drehzahlabhängigkeit aufweisen. Dieser Effekt

war in diesem Fall nicht messbar (siehe Abbildung 1.3 beziehungsweise 1.4) und ist deshalb vernachlässigbar klein.

Um auftretendes Messrauschen zu verhindern wird auf die Signale von Abbildung 1.3 ein Tiefpass Filter mit Butterworth Charakteristik angewandt. Dieser dämpft Frequenzanteile die höher als eine eingestellte Grenzfrequenz sind und lässt die darunterliegenden Frequenzen nahezu unberührt. Dieser Filter wird mit der Python Funktion `signal.butter()` definiert und mit `signal.filtfilt()` angewandt. Verwendung findet der Butterworth-Filter nur bei diesem gestuften Hochdrehen der Hauptspindel, da bei der Beschleunigungsmessung beim Fräsen das Signal nicht verfälscht werden soll.

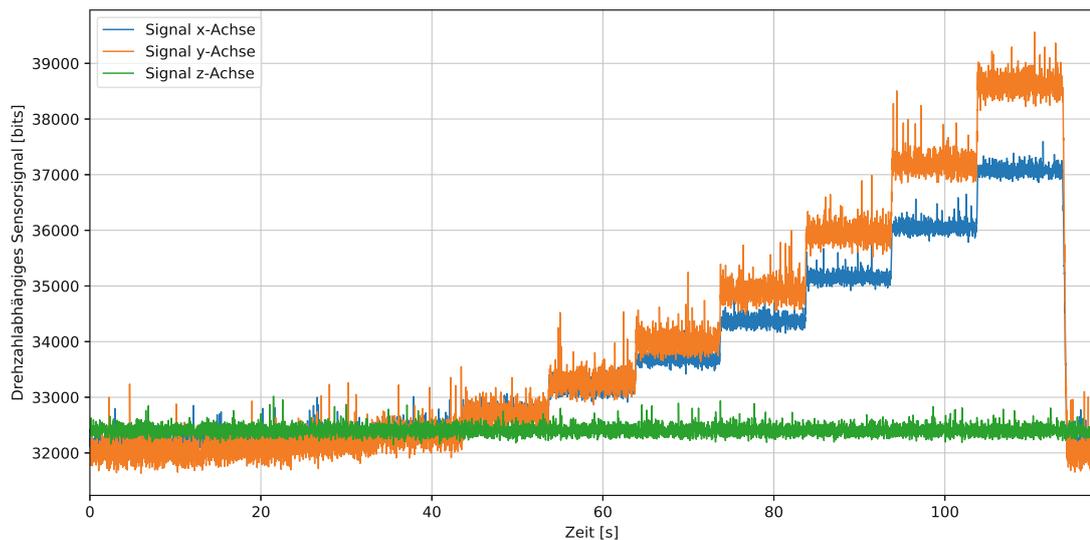


ABBILDUNG 1.3: ungefiltertes Ausgangssignal des Sensorischen Werkzeughalters für verschiedene Drehzahlen der Spindel

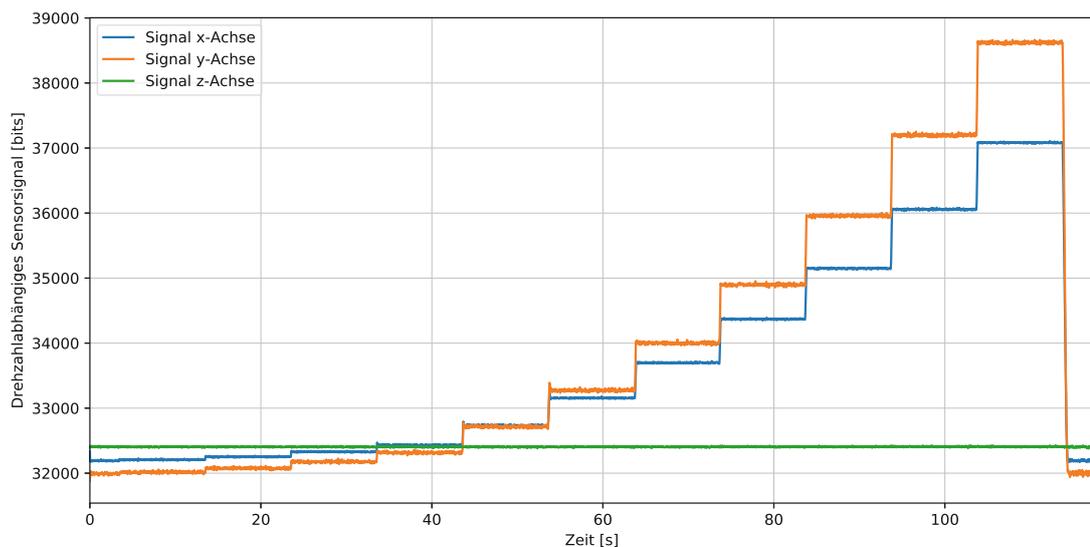


ABBILDUNG 1.4: Tiefpass-gesfiltertes Signal des Sensorischen Werkzeughalters für verschiedene Drehzahlen der Spindel

Als Ausgleichsfunktion wird eine quadratische Funktion der Form

$f(\omega) = r^2 \cdot \omega + c$  angenommen, da der Effekt nur von der Zentripetalbeschleunigung abhängig ist. Mit dieser Ausgleichsfunktion soll die Drehzahlabhängigkeit im Signal eliminiert werden. In Abbildung 1.5 ist die Ausgleichsfunktion, für die in Abbildung 1.4 gezeigte y-Achse, dargestellt. Zum Bestimmen der Funktion werden die Punkte bei 2000 u/min und 6000 u/min herangezogen. Es ist zu erkennen, dass alle weiteren Punkte von der Funktion mit akzeptabler Genauigkeit geschnitten werden. Auch beim Stillstand der Achse ist der gerundete Funktionswert 32000 Bit, was die Richtigkeit der Analyse bestätigt. Der maximale relative Fehler von den gemessenen Punkten und der Ausgleichsfunktion tritt bei 10000 u/min auf und hat einen Betrag von 0.23%. Die ausgewerteten Parameter der Gleichung der Zentripetalbeschleunigung (Gleichung 1.1) sind einerseits die Konstante  $c = 31992.2164$  Bit und andererseits die Konstante  $r = 0.29043$  Bit/s<sup>2</sup>.

Die hier beschriebenen Werte gelten nur für einen spezifischen Sensorischen Werkzeughalter (Modell STH-216), da die Einbauabweichungen von Halter zu Halter variieren.

$$\text{Beschleunigungssignal}_{x/y}(\omega) = r_{x/y} \cdot \omega^2 + c_{x/y} \quad (1.1)$$

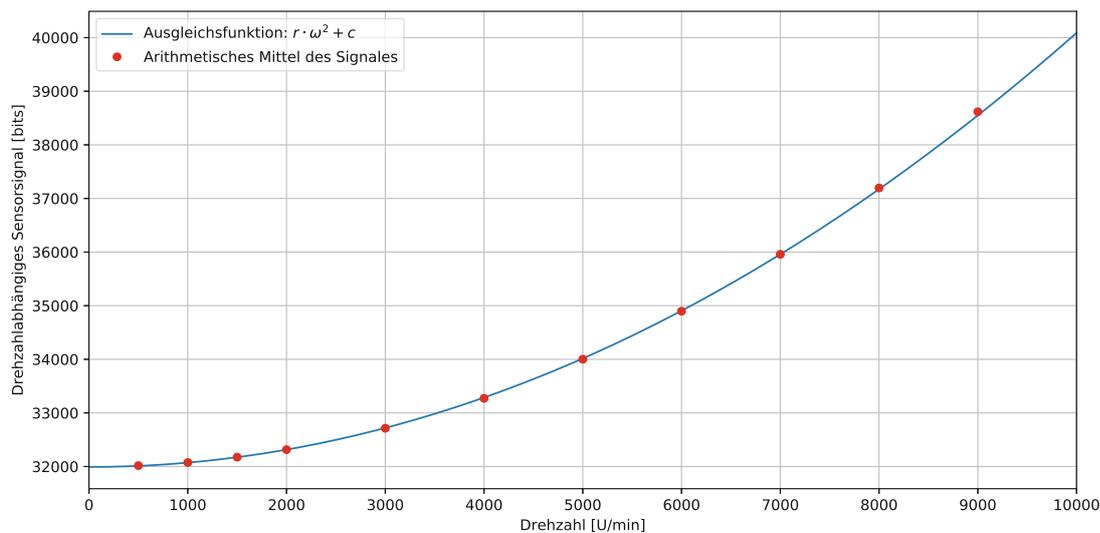


ABBILDUNG 1.5: Ausgleichsfunktion des zur Beschleunigung proportionalen Sensorsignales des Sensorischen Werkzeughalters in y-Richtung (radial)

In dem hier gezeigten Vorversuch dieser Arbeit wird überprüft, ob der Sensor auch bei höheren Drehzahlen das zu erwartende Ergebnis liefert. Im Fertigungsprozess ist dies kaum möglich, jedoch ist die Zentripetalbeschleunigung ein offensichtlich auftretender Effekt, welcher gut als Kontrolle dient. Ein weiterer Schluss der aus diesem Versuch gezogen wird, ist jener, dass ein Übersteuern des Sensors unterhalb einer Drehzahl von 10000 u/min ausgeschlossen werden kann.

## 1.5 Prozessinstabilitäten beim Fräsen

Die Resultate von Prozessinstabilitäten beim Fräsen sind unter anderem geringere Standzeiten der Werkzeuge, höhere Belastung der Maschinenführungen, Lärmbelastung für Maschinenbediener und Schäden an den Werkstückoberflächen. Diese Instabilitäten müssen nicht nur durch den Regenerativeffekt beziehungsweise durch regeneratives Rattern entstehen, sondern können auch durch erhöhten Schneidenverschleiß oder Schneidenbruch zustande kommen. Das Erkennen eines Schneidenbruchs mit Hilfe des Sensorischen Werkzeughalters wird in [Ramsauer, 2019] behandelt.

Das Grundproblem des Ratterns wird in [Klocke und König, 2008] auf den Punkt gebracht. *„Der Fräsprozess ist nur bei bestimmten Kombinationen für Schnittgeschwindigkeit und Zustellung stabil durchführbar, außerhalb dieser Bereiche treten Ratterschwingungen auf. Diese charakteristischen Stabilitätsbereiche müssen bekannt sein, sonst kann der Prozess nicht an der Leistungsgrenze geführt werden.“*

Eine historische Zusammenfassung zum Thema Rattern von 1966 bis ins Jahr 2009 gibt es in [Quintana und Ciurana, 2011]. Darin zeichnet sich ab, dass die Professoren Altintas und Budak, welche auch in dieser Arbeit noch öfters zitiert werden, einen enormen Einfluss auf dieses Forschungsgebiet hatten und noch immer haben. Es wird ebenso angemerkt, dass es passive und aktive Methoden gibt um Ratterschwingungen zu vermeiden. So kann teilweise passiv durch verschiedene Fräserformen, welche eine nicht symmetrische Teilung der Zähne aufweisen, dieser Art von Instabilität Abhilfe geschaffen werden. Eine andere Möglichkeit ist aktiv einzugreifen und durch das ständige variieren der Drehzahl das Auftreten von Rattern zu vermeiden. Das in dieser Arbeit angestrebte Ziel Stabilitätsindikatoren zu finden, welche anhand eines Beschleunigungssignals ausgewertet werden um den Prozess stabil zu halten, fällt auch unter den aktiven Bereich.

## 1.6 Potenzialanalyse

Ziel der Potenzialanalyse ist es aufzuzeigen, warum die Betrachtung der Stabilität und den daraus folgenden aktuellen Entwicklungen im Maschinenbau zunehmend an Bedeutung gewinnen, auch unter praktischen Gesichtspunkten, obwohl die Theorien dazu schon seit den 60er Jahren existieren [Tobias, 1965b und Tobias, 1965a]. Dazu werden zwei Maschinen ähnlicher Größe verglichen, welche vom Baujahr zirka 30 Jahre auseinander liegen. Bei der älteren Maschine handelt es sich um eine Thyssen Hüller Hille nb-h 90 mit einer maximalen Spindeldrehzahl von 6000 u/min. Beim neueren Fräszentrum handelt es sich um eine EMCO UMILL 750 mit einer maximalen Spindeldrehzahl von 15000 u/min.

Laut Stabilitätskarte [Tobias, 1965b, Gradišek u. a., 2005, Brecher und Weck, 2006, Quintana und Ciurana, 2011] sind bei höheren Drehzahlen tendenziell größere Schnitttiefen realisierbar und die stabilen Bereiche erstrecken sich auf einen weit größeren



(A) Hüller Hille nb-h 90 (in den 80er Jahren weit verbreitet)



(B) EMCO UMILL 750 [EMCO GmbH, 2019]

ABBILDUNG 1.6: Vergleich zweier Fräszentren ähnlicher Größe

Drehzahlbereich. In Abbildung 1.7 wird eine Stabilitätskarte für gleiche modale Parameter gezeigt. Der gelb markierte Bereich ist der Drehzahlbereich der Hüller Hille Fräsmaschine. Der darüber hinausgehende rote Bereich beschreibt den zusätzliche Drehzahlbereich der von der EMCO Fräsmaschine genutzt werden kann. Da bei diesem Beispiel nur der erste Eigenmode berücksichtigt wird und auch die modalen Parameter nur grob abgeschätzt werden können, kann nicht davon ausgegangen werden, dass es in der Praxis möglich ist, mit den hier angenommenen Parametern und zum Beispiel einer axialen Schnitttiefe von 3 mm und einer Spindeldrehzahl von 10000 u/min stabil zu fräsen. Bei einer theoretischen Schnitttiefe von 1 mm, bleibt der Fräsmaschine mit langsam drehender Spindel lediglich ein Bereich von maximal 350 u/min zum stabilen Bearbeiten. Die schneller drehende Maschine kann hingegen in einem Drehzahlbereich von 1680 u/min arbeiten. Dies hat den Vorteil, dass eine mögliche Regelung, deren Ausgangsbasis geschätzte Eigenfrequenzen sind, mehr Spielraum im stabilen Bereich hat.

In [Brecher und Weck, 2017] wurde eine Stabilitätskarte experimentell bestimmt. Diese ist in Abbildung 1.8 wiedergegeben und zeigt qualitativ gute Übereinstimmung mit dem hier präsentierten, vereinfachten Modell. Deutlich zum Vorschein kommt, dass es im ungünstigsten Fall eine axiale Schnitttiefe von 3 mm und im günstigsten Fall eine axiale Schnitttiefe von 15 mm möglich ist. Das heißt, dass die Schnitttiefe in diesem Extremfall, um Faktor 5 gesteigert werden kann. Es wird von einem Extremfall gesprochen, weil Highspeed-Cutting nicht standardmäßig Verwendung findet und hier auch plakativ das schlechteste mit dem besten Ergebnis verglichen wird.

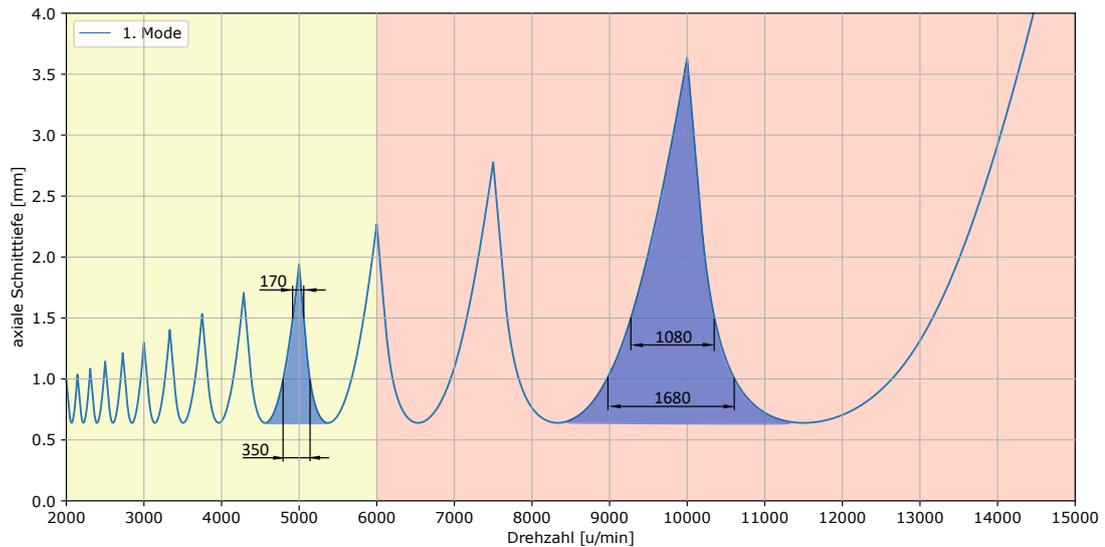


ABBILDUNG 1.7: Aus Erfahrungswerten für modale Parameter berechnete Stabilitätskarte: gelb  $\hat{=}$  Hüller Hille / gelb + rot  $\hat{=}$  EMCO

Werkzeug-Durchmesser	32mm
Schneidenzahl	2
Werkstück-Material	Aluminium 7050
Spindeldrehzahl	variiert von 18000-30000 u/min
Zahnvorschub	0.18mm
Axiale Schnitttiefe	variiert von 3-15mm

TABELLE 1.1: Versuchsparameter aus [Brecher und Weck, 2017]

Ein weiteres deutliches Beispiel kann aus diesem Versuch abgeleitet werden. Bei einer axialen Schnitttiefe von 10mm und einer Drehzahl von 20.000 u/min kann durch Erhöhung der Drehzahl auf 29.500 u/min der Prozess immer noch stabil gehalten werden. Die Vorschubgeschwindigkeit würde dabei um 47,5% erhöht werden, was zu einer Verkürzung der Bearbeitungszeit führt. Voraussetzung hierfür wäre jedoch ein rasches Beschleunigen. Während des Durchfahrens des instabilen Bereichs kommt es unweigerlich zu einem Anklingen von Ratterschwingungen. Ist die Verweildauer kurz genug, damit die aufschwingende Ratter-Amplitude keine problematischen Werte erreicht, kann dieser Umstand toleriert werden.

Ebenso klar erkennbar ist der Umstand, dass ein mäßig instabiler Prozess, also ein Punkt oberhalb der Kurve, durch Verringerung, aber auch durch Erhöhung der Drehzahl stabilisiert werden kann. So sollte das Ziel einer Regelung sein, den Prozess durch Drehzahlerhöhung zu stabilisieren, um somit nicht nur die Oberfläche des Werkstücks und die Standzeit des Werkzeuges zu verbessern, sondern auch die Bearbeitungsdauer zu reduzieren. Es steht außer Frage, dass eine Erhöhung von Drehzahl und Vorschubgeschwindigkeit zu höheren Schnittkräften führt, weshalb sich die Standzeit des Werkzeuges auch verschlechtern kann.

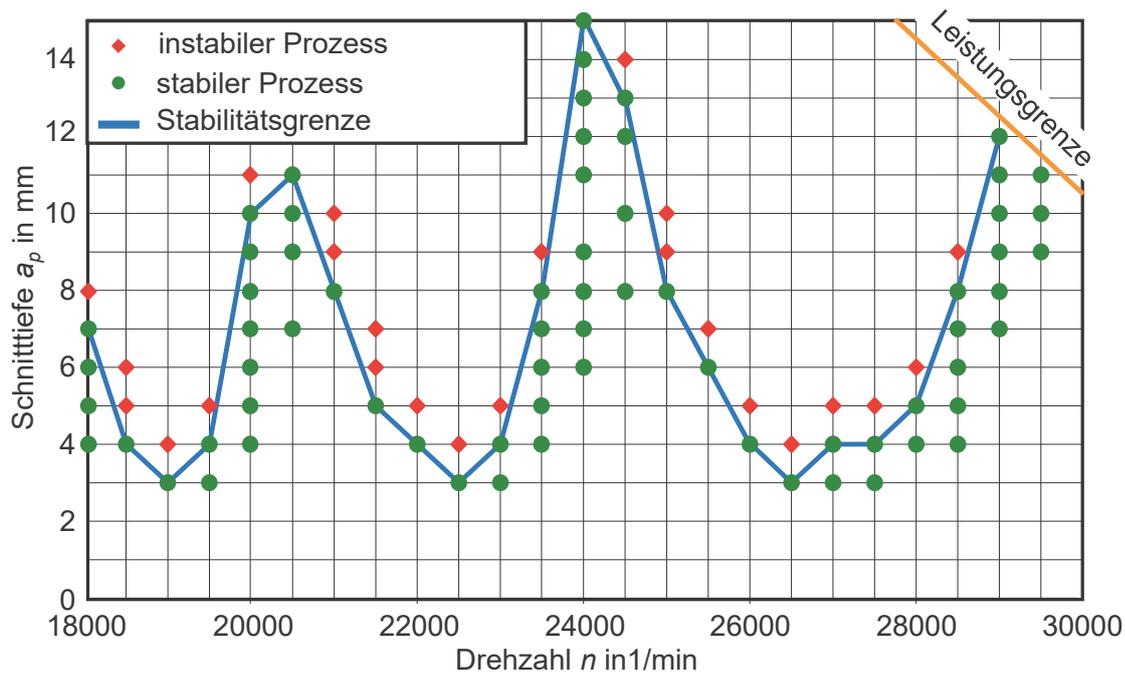


ABBILDUNG 1.8: Experimentelle Stabilitätskarte [Brecher und Weck, 2017]

In Kapitel 3 wird durch eigene Versuche gezeigt, dass selbst bei konservativ angenommenen Schnittparametern und geringen Drehzahlen die stabile Schnitttiefe zirka um den Faktor 2 variieren kann.

## Kapitel 2

# Stabilitätskarten

### 2.1 Allgemein

Stabilitätskarten sind spezielle Diagramme, welche es ermöglichen, Fräsprozesse anhand von axialer Schnitttiefe und Drehzahl in stabile und instabile Prozesse zu kategorisieren. Alle anderen Parameter wie zum Beispiel Zahnvorschub, radiale Schnittbreite oder modale Parameter wie Steifigkeit, Dämpfung oder schwingende Masse der Maschinenstruktur werden als konstant angenommen. Auch zeitliches Einschwingverhalten wird nicht berücksichtigt. Dieser Umstand verleiht den Stabilitätskarten einen sehr theoretischen Charakter. In der folgenden Abbildung (2.1) ist so eine Stabilitätskarte abgebildet.

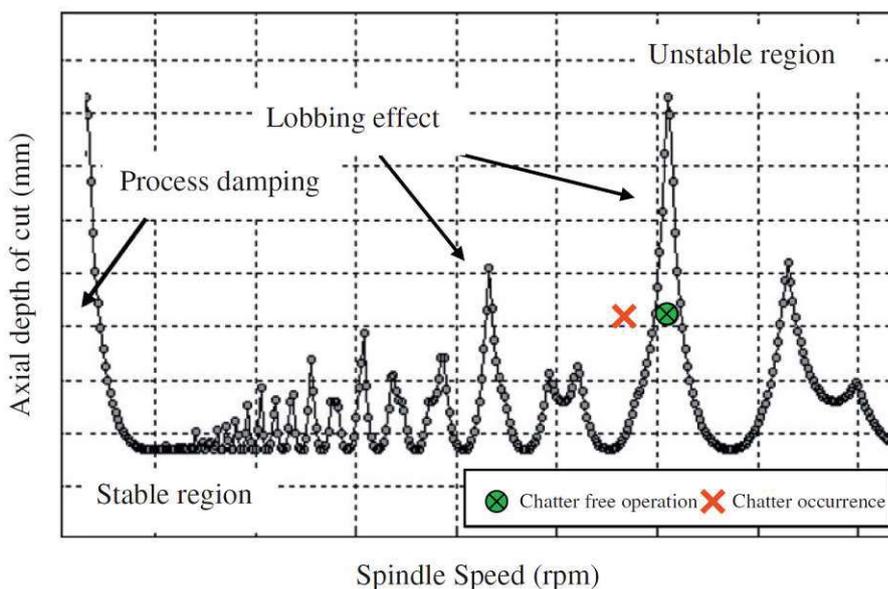


ABBILDUNG 2.1: Allgemeine Stabilitätskarte [Quintana und Ciurana, 2011]

Die Kurve in der Stabilitätskarte unterteilt mögliche Betriebspunkte in einen stabilen und einen instabilen Bereich. Der Bereich oberhalb der Kurve gilt als instabil. Wird eine Schnitttiefen-Drehzahl-Kombination dort gewählt, beginnt das System sich aufgrund der Anregung der Maschinenstruktur mit bei Eigenfrequenz, aufzuschwingen.

Die Folgen werden von Baur wie folgt beschrieben: "Das Auftreten von Ratterschwingungen ist äußerst problematisch, da diese aufgrund der großen Schwingungsamplituden eine mangelhafte Oberflächenqualität und Maßhaltigkeit, einen starken Verschleiß an Werkzeug und Maschine sowie eine hohe Lärmbelastung hervorrufen. Zudem kann die installierte Hauptspindelleistung der Werkzeugmaschine nicht voll ausgenutzt werden, wodurch die Produktivität sinkt." [Baur, 2013]

Die angesprochene mangelhafte Oberflächenqualität äußert sich durch Rattermarken. Exemplarisch werden diese Marken in Kapitel 3 in Abbildung 3.7b gezeigt, welche durch eigens durchgeführte Versuche erzeugt wurden. Unterhalb der Kurve ist der stabile Bereich, wo dieser Effekt nicht auftritt und die Oberfläche keine unerwarteten Qualitätseinbußen erleidet. In der angeführten Stabilitätskarte ist auch ein anderer Effekt zu erkennen. Es gibt nämlich Bereiche, welche besonders hohe, stabile Schnitttiefen aufweisen. Dieser Effekt - in [Quintana und Ciurana, 2011] als *Lobbing Effect* bezeichnet - soll bei einer Regelung so gut wie möglich ausgenutzt werden. Das würde bedeuten, dass es beim Auftreten von Ratterschwingungen möglich ist, auch durch Erhöhung der Drehzahl und Beibehaltung der anderen Parametern, was einer Steigerung der Produktivität gleichzusetzen ist, den Prozess zu stabilisieren.

Der zuvor genannte theoretische Charakter der Stabilitätskarten wird am folgenden Beispiel sichtbar. Wird die Spindel in z-Richtung, aufgrund axialer Zustellung, weiter ausgefahren, so verringert sich unter anderem die Steifigkeit und die frei schwingende Masse wird erhöht. Für unterschiedliche modale Parameter muss es daher unterschiedliche Stabilitätskarten geben. Aufgrund der geringen Relevanz dieser Einflüsse werden solche Effekte vernachlässigt. Ebenso kann auch nur eine endliche Anzahl an Eigenmoden berücksichtigt werden, deren Abschätzung mit hohem Aufwand verbunden ist, beziehungsweise die Genauigkeit der Abschätzungen mehr oder weniger ungenau sein können. Aus diesem Grund werden in Kapitel 3, in dem die Versuche dargestellt werden, auch die modalen Parameter erst abgeschätzt und in weiterer Folge so nachjustiert, damit sie mit den Ergebnissen optimal zusammenpassen.

Um die weiteren Kapitel besser zu verstehen, wird angemerkt, dass die häufig im mitteleuropäischen Raum verwendete Kienzle Gleichung (nach Otto Kienzle) nicht für die Berechnung von Stabilitätskarten herangezogen wird. International findet die Nomenklatur von Yusuf Altintas nahezu ausschließlichen Gebrauch. Es ist naheliegend, dass die von ihm entwickelten Algorithmen auf den von ihm entwickelten Schnittkraftmodellen aufbauen. In [Yangui u. a., 2010] wird ein Überblick über gängige Schnittkraftmodelle gegeben.

Autor	Modell
Kienzle O. (1950)	$F_t = a_p K_{c1.1} h^{1-m_c}$
Altintas Y. (1998)	$F_t = a_p K_{tc} h + a_p K_{te} \quad F_r = a_p K_{rc} h + a_p K_{re}$
Insperger T. (2000)	$F_t = a_p K_t h^{x_f} \quad F_r = a_p K_r h^{x_f}$

TABELLE 2.1: Modellierungsarten der Schnittkraft [Yangui u. a., 2010]

## 2.2 Stabilitätsalgorithmus von Altintas und Budak

Das folgende Berechnungsschema von Stabilitätskarten wurde in einigen Publikationen, u.a. von Altintas und Budak, behandelt und schlussendlich wurde dieses Wissen in [Budak und Altintas, 1998a] und [Budak und Altintas, 1998b] konzentriert und aufbereitet. Die Stabilität des Fertigungsprozesses ist ein nicht triviales Thema, weswegen dieses seit vielen Jahren immer wieder neu aufgearbeitet wird. Hierzu weiterführende Literatur ist unter anderem [Altintas, Engin und Budak, 1999], [Altintas und Weck, 2004], [Lacerda und Lima, 2004], [Tekeli und Budak, 2004], [Insperger, Lehotzky und Stepan, 2015] und [Li, Wang und Shi, 2017].

### 2.2.1 Erstellen von Stabilitätskarten für zwei Freiheitsgrade

Ausgangsbasis des gesamten Algorithmus zur Berechnung von Stabilitätskarten ist das in Gleichung 2.2 dargestellte Eigenwertproblem. Hergeleitet wird dieses Eigenwertproblem in [Altintas und Budak, 1995] und stammt von der Grundgleichung 2.1. Die Lösung dieser Gleichung beinhaltet nicht mehr die dynamische Auslenkung zufolge der periodischen Zerspanungskräfte (Fremderregung), sondern trifft Aussagen über die Stabilität für parametererregte Schwingungen (Selbsterregung).

$$\underline{x} = \underline{G} \cdot \underline{F} = \Lambda \cdot \underline{\alpha} \cdot \underline{G} \cdot \underline{x} \quad (2.1)$$

Altintas begründet, dass das Ergebnis des Eigenwertproblems der Produktmatrix von  $\underline{G}$  und  $\underline{\alpha}$  Auskunft über die Stabilität des Fräsprozesses gibt, wobei  $\underline{G}$  die Übertragungsfunktion darstellt und die Matrix  $\underline{\alpha}$  als deren Gewichtung gesehen werden kann. Das heißt, dass der Eigenwert  $\Lambda$  Informationen über die axiale Grenzschnitttiefe beinhalten muss. Würde das Eigenwertproblem nur für Übertragungsfunktionen  $\underline{G}$  gelöst werden, ohne den gerichteten dynamischen Fräskoeffizienten zu berücksichtigen, wäre der Eigenwert gleich der Eigenfrequenz. In weiterer Folge werden die Anteile der Nebendiagonale von  $\underline{G}$ ,  $G_{xy}$  und  $G_{yx}$  (cross-transfer functions), vernachlässigt. Dies führt zu einer unkomplizierteren Berechnung und auch einer schnelleren und einfacheren Abschätzung der Parameter.

$$\det \left( [I] + \Lambda \begin{bmatrix} \alpha_{xx} & \alpha_{xy} \\ \alpha_{yx} & \alpha_{yy} \end{bmatrix} \begin{bmatrix} G_{xx}(j\omega_c) & G_{xy}(j\omega_c) \\ G_{yx}(j\omega_c) & G_{yy}(j\omega_c) \end{bmatrix} \right) = 0 \quad (2.2)$$

$$\det \left( [I] + \Lambda \begin{bmatrix} \alpha_{xx} & \alpha_{xy} \\ \alpha_{yx} & \alpha_{yy} \end{bmatrix} \begin{bmatrix} G_x(j\omega_c) & 0 \\ 0 & G_y(j\omega_c) \end{bmatrix} \right) = 0 \quad (2.3)$$

Um das Eigenwertproblem zu lösen, müssen zuerst die gerichteten dynamischen Fräskoeffizienten nach Gleichung 2.7 berechnet werden. In der Literatur wird beim Stirn-Umfangsfräsen zwischen Gleichlauf- und Gegenlaufräsen unterschieden, was jedoch bei der Verwendung des richtigen Formalismus nicht notwendig ist. Für

Gleichlaufräsen (downmilling) gelten die Gleichungen 2.4 und für Gegenlaufräsen (upmilling) die Gleichungen 2.5.

$$\begin{aligned}\phi &= \phi_{start} \\ \phi_{start} &= \arccos(1 - 2a_e/D)\end{aligned}\quad (2.4)$$

$$\begin{aligned}\phi_{exit} &= \pi \\ \phi &= \phi_{exit} \\ \phi_{start} &= 0 \\ \phi_{exit} &= \arccos(1 - 2a_e/D)\end{aligned}\quad (2.5)$$

Da Gleichung 2.6 für den Eingriffswinkel  $\phi$  immer gültig ist, unabhängig davon, ob Gleichlauf- oder Gegenlaufräsen praktiziert wird, findet diese auch im selbst erstellen Berechnungsskript (siehe Anhang A.1) Verwendung.

$$\phi = \arccos\left(1 - \frac{2a_e}{D}\right)\quad (2.6)$$

Die letzte Unbekannte in den Gleichungen 2.7 ist der spezifische radiale Schnittkraftkoeffizient  $K_r = K_{tc}/K_{rc}$ . Dieser Koeffizient ist einheitenlos und darf nicht mit dem radialen Schnittkraftkoeffizienten  $K_{rc}$  vertauscht werden. In [Rubeo und Schmitz, 2016] wird gezeigt, dass die Schnittkraftkoeffizienten vom zu bearbeitenden Werkstoff, vom Zahnvorschub und von der Drehzahl abhängig sind. Diese werden jedoch als konstant angenommen. Wie in Tabelle 2.1 gezeigt, wird die Modellierung dieser Schnittkraftkoeffizienten Altintas zugesprochen. Mit dem berechneten Eingriffswinkel  $\phi$  und dem spezifischen radialen Schnittkraftkoeffizient  $K_r$  können nun die gerichteten dynamischen Fräskoeffizienten berechnet werden. Für den Fall des Einfreiheitsgradschwingers reicht die Berechnung von  $\alpha_{xx}$  beziehungsweise  $\alpha_{yy}$ .

$$\begin{aligned}\alpha_{xx} &= \frac{1}{2} [\cos 2\phi - 2K_r\phi + K_r \sin 2\phi] \\ \alpha_{xy} &= \frac{1}{2} [-\sin 2\phi - 2\phi + K_r \cos 2\phi] \\ \alpha_{yx} &= \frac{1}{2} [-\sin 2\phi + 2\phi + K_r \cos 2\phi] \\ \alpha_{yy} &= \frac{1}{2} [-\cos 2\phi - 2K_r\phi - K_r \sin 2\phi]\end{aligned}\quad (2.7)$$

Als nächstes wird die Übertragungsfunktion  $G$  bestimmt. Dafür wird das Differentialgleichungssystem für einen Einmassenschwinger in einem krafterregten System (Gleichung 2.8 beziehungsweise 2.9) aufgestellt. Hier können zwei gleichwertige Funktionen in zwei orthogonal zueinander stehenden Richtungen ausgewertet werden. Allgemein können die Differentialgleichungen mit Gleichung 2.10 angeschrieben

werden. Diese werden umgeformt und LaPlace-transformiert um den Nachgiebigkeitsfrequenzgang beziehungsweise auch Übertragungsfunktion laut Gleichung 2.11 zu erhalten. Durch das Einsetzen der komplexwertigen Ratterfrequenz  $j\omega_c$  erhält man Gleichung 2.12.

$$\underline{\underline{M}} \cdot \ddot{x} + \underline{\underline{C}} \cdot \dot{x} + \underline{\underline{K}} \cdot x = \underline{F}(t) \quad (2.8)$$

$$\begin{bmatrix} m_m & 0 \\ 0 & m_m \end{bmatrix} \cdot \ddot{x} + \begin{bmatrix} c_m & 0 \\ 0 & c_m \end{bmatrix} \cdot \dot{x} + \begin{bmatrix} k_m & 0 \\ 0 & k_m \end{bmatrix} \cdot x = \underline{F}(t) \quad (2.9)$$

$$m_m \cdot \ddot{x} + c_m \cdot \dot{x} + k_m \cdot x = F_x(t) \quad (2.10)$$

$$m_m \cdot \ddot{y} + c_m \cdot \dot{y} + k_m \cdot y = F_y(t)$$

$$G(s) = \frac{x(s)}{F_x(s)} = \frac{y(s)}{F_y(s)} = \frac{1}{k_m + c_m \cdot s + m_m \cdot s^2} \quad (2.11)$$

$$G(j\omega_c) = \frac{1}{k_m + c_m \cdot j\omega_c - m_m \cdot \omega_c^2} \quad (2.12)$$

Werden die Dämpfung und modale Masse durch den Dämpfungsparameter (auch Lehr'sches Dämpfungsmaß genannt) und durch die Eigenfrequenz laut Gleichungen 2.13 ersetzt, kann man die Übertragungsfunktion auch wie in Gleichung 2.14 dargestellt, ausdrücken. Das hat den Vorteil, dass manche Zusammenhänge leichter ersichtlich sind wie zum Beispiel, dass der Realteil der Übertragungsfunktion gegen Null geht, sobald die Ratterfrequenz  $\omega_c$  gleich der Eigenfrequenz  $\omega_n$  ist. Gleichung 2.14 ist in dieser Form auch in [Budak, 1994] zu finden.

$$\omega_n = \sqrt{\frac{k_m}{m_m}} \quad (2.13)$$

$$\xi_m = \frac{c}{2\sqrt{m_m k_m}}$$

$$G(j\omega_c) = \frac{\omega_n^2 / k_m}{\omega_n^2 - \omega_c^2 + 2j\xi_m \omega_n \omega_c} \quad (2.14)$$

Die Übertragungsfunktion aus Gleichung 2.14 besteht im Allgemeinen aus einem Real- und Imaginärteil. Diese beiden Teile werden in Abbildung 2.2 dargestellt. Diese Grafik wurde für eine Eigenfrequenz von  $f_n = 1000\text{Hz}$  beziehungsweise  $\omega_n = 1000 \cdot 2\pi$  rad geplottet. Im Kapitel 3 werden mittels Impulshammer experimentell ermittelte Übertragungsfunktionen gezeigt, welche ähnlich zu den hier analytisch berechneten Übertragungsfunktionen sind.

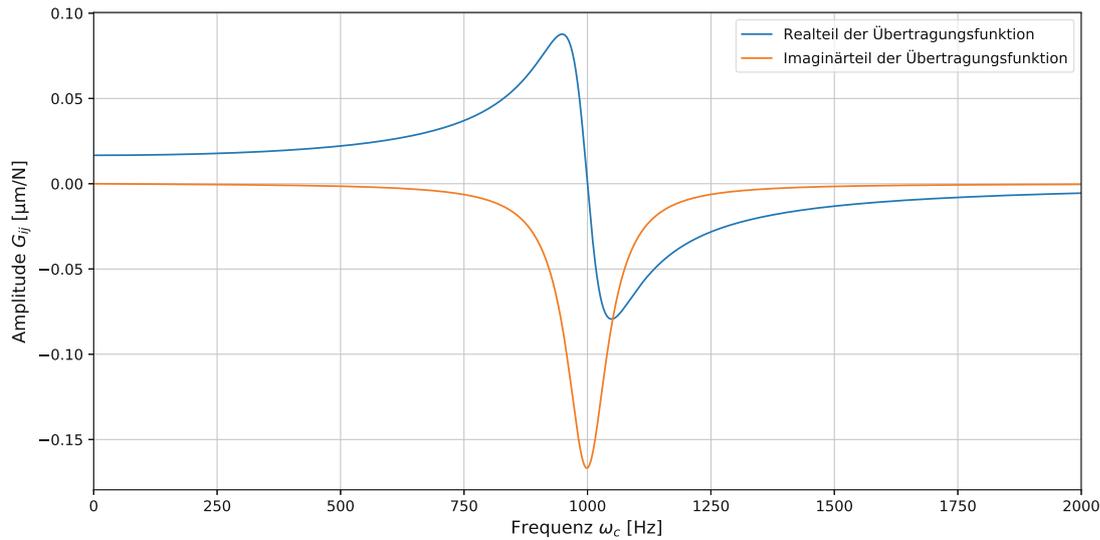


ABBILDUNG 2.2: Übertragungsfunktion für eine Eigenfrequenz von 1000Hz

Für den Fall des in Abbildung 2.3 dargestellten Systems, werden die richtungsgekoppelten Übertragungsfunktionen von Werkstück (Index  $w \rightarrow workpiece$ ) und Werkzeug (Index  $c \rightarrow cutter$ ) in und normal zur Fräsrichtung durch Addition auf zwei Übertragungsfunktionen reduziert (Gleichung 2.15).

$$\begin{aligned} G_x &= G_{xc} + G_{xw} \\ G_y &= G_{yc} + G_{yw} \end{aligned} \quad (2.15)$$

Mit dem Aufstellen der Übertragungsfunktion  $G$  und der gerichteten dynamische Fräskraftkoeffizienten wurden alle notwendigen Parameter berechnet, um das in Gleichung 2.2 beschriebene Eigenwertproblem zu lösen. Bei einem System mit zwei Freiheitsgraden führt das Eigenwertproblem zu der in Gleichung 2.17 beschriebenen quadratischen Gleichung mit den Parametern  $a_0$  und  $a_1$ , welche in den Gleichungen 2.16 beschrieben werden. Für die quadratische Gleichung gibt es zwei Lösungen von denen in diesem Fall immer lediglich die Lösung mit negativen Vorzeichen vor der Wurzel relevant sind. Keine der beiden Lösungen ist offensichtlich trivial, da beide Lösungen positive und teils realistische Werte für die axiale Grenzschnitttiefe liefern.

$$\begin{aligned} a_0 &= G_x(j\omega_c)G_y(j\omega_c) (\alpha_{xx}\alpha_{yy} - \alpha_{xy}\alpha_{yx}) \\ a_1 &= \alpha_{xx}G_x(j\omega_c) + \alpha_{yy}G_y(j\omega_c) \end{aligned} \quad (2.16)$$

$$\Lambda = -\frac{1}{2a_0} \left( a_1 \pm \sqrt{a_1^2 - 4a_0} \right) \quad (2.17)$$

Um die Gleichung für die Grenzschnitttiefe zu erhalten, müssen Zähnezahl, tangentialer Schnittkraftkoeffizient und das Verhältnis von Realteil und Imaginärteil der

Lösung des Eigenwertproblems kombiniert werden, um daraus Gleichung 2.19 zu gewinnen. Zuvor wird noch das Verhältnis  $\kappa$ , aus Realteil zu Imaginärteil der Lösung des Eigenwertproblems, definiert (Gleichung 2.18). Die Abhängigkeit des Schnittkraftkoeffizienten widerspiegelt die Materialabhängigkeit der axialen Grenzschnitttiefe.

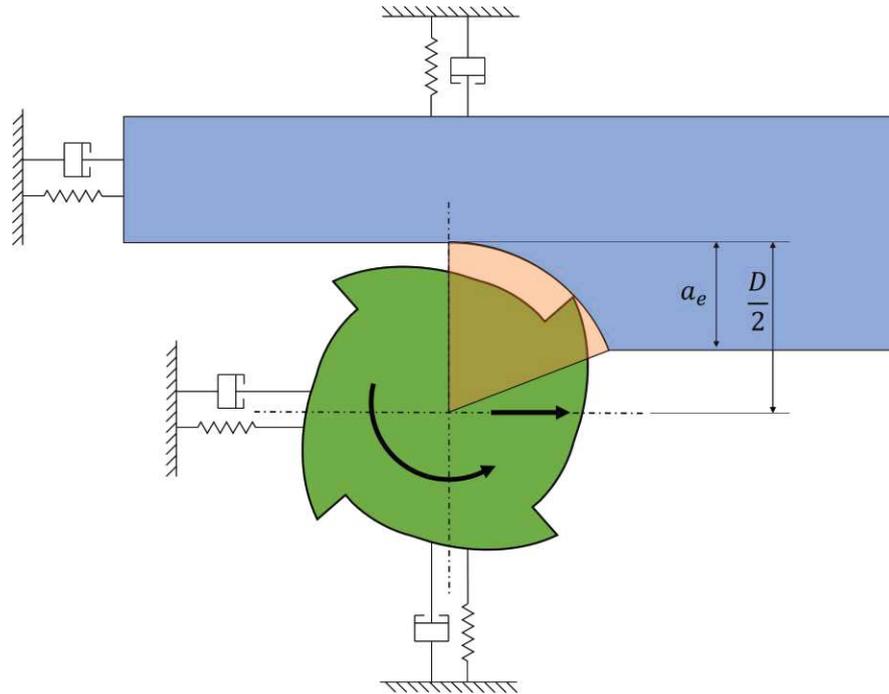


ABBILDUNG 2.3: System mit zwei Freiheitsgraden

$$\kappa = \frac{\operatorname{Re}(\Lambda)}{\operatorname{Im}(\Lambda)} \quad (2.18)$$

$$a_{lim} = -\frac{2\pi \operatorname{Im}(\Lambda)}{NK_{tc}} \left( \kappa + \frac{1}{\kappa} \right) \quad (2.19)$$

Als nächstes wird für die berechnete Grenzschnitttiefe  $a_{lim}$  die zugehörige Drehzahl berechnet. Dafür muss zuerst die Phasenverschiebung der Spannungsänderung  $\epsilon$  bestimmt werden. Für zwei Freiheitsgrade wird für die Berechnung von  $\epsilon$  die Gleichung 2.20 herangezogen.

$$\epsilon = \pi - 2 \tan^{-1} (1/\kappa) \quad (2.20)$$

In [Quintana und Ciurana, 2011] wird die Phasenverschiebung der Spannungsänderung exemplarisch anhand dreier Beispielen gezeigt. Es ist ersichtlich, dass nur bei der Phasenverschiebung von null oder einer geraden ganzzahligen Vielfachen von  $\pi$  ein konstant dicker Span entsteht. Bei einer ungeraden ganzzahligen Vielfachen von  $\pi$  ist die Spannungsänderung maximal und somit auch die geringsten axialen Schnitttiefen möglich. Es ist anzumerken, dass die maximale Spandicke deutlich größer als theoretische Spandicke sein kann. Im Abschnitt 2.4.3 wird noch näher auf Sonderfälle von  $\epsilon$  eingegangen.

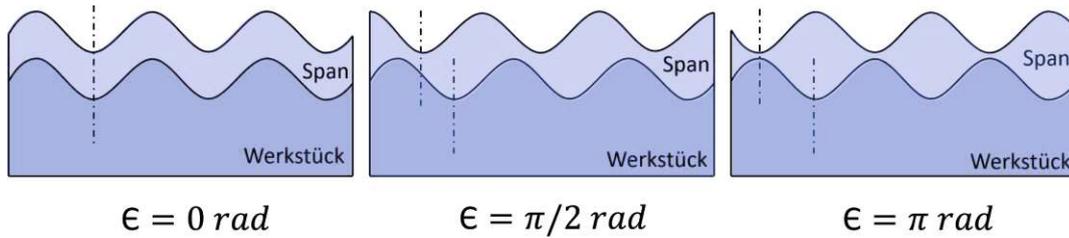


ABBILDUNG 2.4: Phasenverschiebung der Spanndickenänderung [Quintana und Ciurana, 2011]

Die Phasenverschiebung der Spanndickenänderung  $\epsilon$  findet auch in [Altintas und Weck, 2004] und [Baur, 2013] Erwähnung. In der dort enthaltenen Abbildung (2.5) wird gezeigt, wie der Faktor  $k$  zu interpretieren ist. Dieser Faktor beschreibt die Ordnung des Rattersackes beziehungsweise die Anzahl von Wellen zwischen zwei Zahnengriffen. Hier ist auch deutlich zu sehen, dass es ab der Drehzahl, welche größer als die maximale Drehzahl des Rattersackes erster Ordnung ist, einen Bereich der absoluten Stabilität für diesen speziellen Eigenmode gibt.

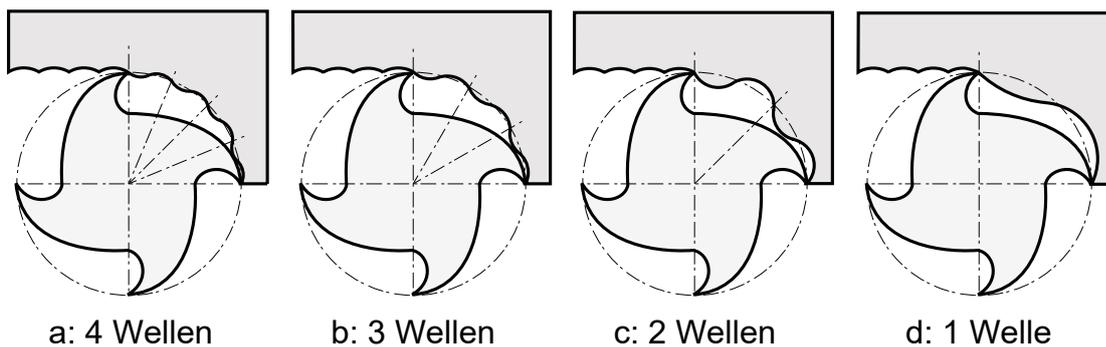
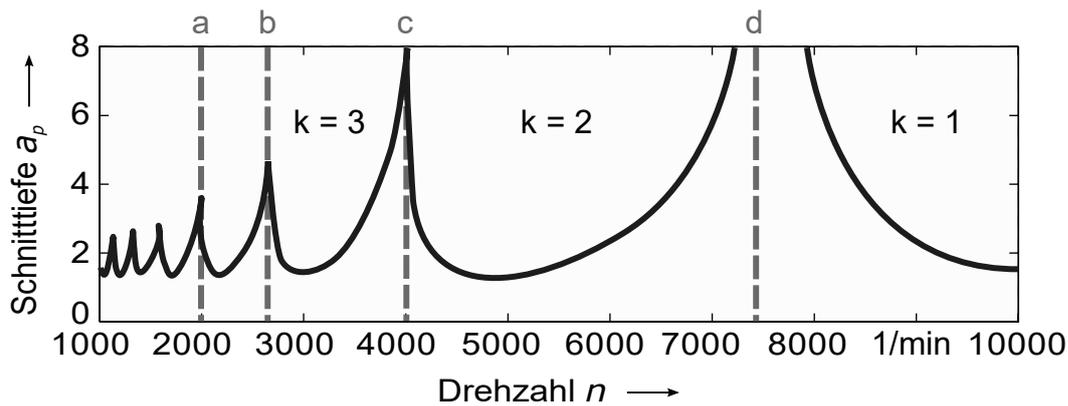


ABBILDUNG 2.5: Beschreibung des Faktors  $k$  [Baur, 2013]

Im Anschluss kann über die Ratterfrequenz, die schon zuvor bei der Übertragungsfunktion angenommen werden musste, und der Phasenverschiebung der Spanndickenänderung die Periodendauer des Zahnengriffes laut Gleichung 2.21 berechnet werden. Aus dem Kehrwert der Periodendauer lässt sich die Zahnengriffsfrequenz, entweder in Hertz oder Radiant pro Sekunden, bilden. (Gleichungen 2.22). Mit Hilfe

von Zähnezahl und Periode des Zahneingriffes lässt sich mit Gleichung 2.23 die Drehzahl in Umdrehungen pro Minute berechnen.

$$T_{ZEF} = \frac{\epsilon + 2k\pi}{\omega_c} \quad (2.21)$$

$$f_{ZEF} = \frac{1}{T_{ZEF}} \quad (2.22)$$

$$\omega_{ZEF} = \frac{2\pi}{T_{ZEF}}$$

$$n = \frac{60}{NT_{ZEF}} \quad (2.23)$$

Das Ergebnis dieses Algorithmus ist aus Gleichung 2.19 die Grenzschnitttiefe und aus Gleichung 2.23 die dazugehörige Drehzahl, welche abhängig von der Ratterfrequenz  $\omega_c$  sind. Beide werden in die Stabilitätskarte eingezeichnet. Durch Variation der Ratterfrequenz und einem konstanten  $k$  aus Gleichung 2.21 wird ein gesamter Rattersack berechnet. Jeder weitere Rattersack entsteht durch die Erhöhung von  $k$ . Der Faktor  $k$  kann dabei jede natürlich Zahl annehmen.

Bis zu diesem Punkt wird die Berechnung eines wahllosen Punktes in der Stabilitätskarte erläutert. Wie es zur Berechnung von einem gesamten Rattersack, zur Berechnung einer Stabilitätskurve für einen bestimmten Eigenmode oder zur Berechnung einer gesamten Stabilitätskarte mit endlich vielen Eigenmoden kommt, wird in Abschnitt 2.2.3 näher erläutert.

### 2.2.2 Stabilitätsalgorithmus für einen Freiheitsgrad

Für einen kürzeren Rechenaufwand und einen geringeren Aufwand der Parameterschätzung kann die vereinfachte Annahme getroffen werden, dass für das Ratterverhalten, Schwingungen normal zur Bearbeitungsrichtung die größte Relevanz haben. Trotz der Sperrung eines Freiheitsgrades kann dieser Algorithmus ebenfalls ausgeführt werden. Dafür reicht es aus, die Übertragungsfunktion der nicht zu berücksichtigenden Richtung Null zu setzen und das oben genannte Schema zu verwenden. Um weitere Rechenzeit zu sparen gibt es für den vereinfachten Fall ein modifiziertes Schema, welches in diesem Absatz beschrieben wird.

Für das System mit einem Freiheitsgrad vereinfacht sich das Eigenwertproblem zu Gleichung 2.24 und mit der Kombination der Zähnezahl und des tangentialen Schnittkraftkoeffizienten, folgt die Gleichung für die Grenzschnitttiefe  $a_{lim}$  (Gleichung 2.25).

$$\Lambda = -\frac{1}{\alpha_{yy}G_y(j\omega_c)} \quad (2.24)$$

$$a_{lim} = \frac{2\pi}{N\alpha_{yy}K_{tc}Re[G_y(j\omega_c)]} \quad (2.25)$$

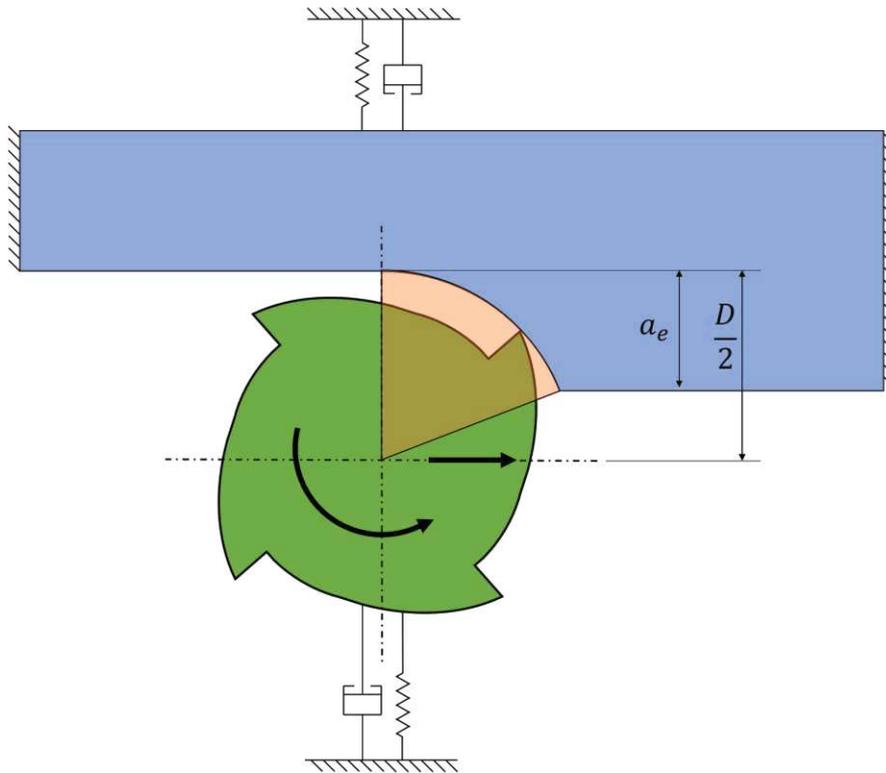


ABBILDUNG 2.6: System mit einem Freiheitsgrad

Die Phasenverschiebung der Spannungsdickenänderung wird laut Gleichung 2.26 berechnet. Im Gegensatz zu Abschnitt 2.2.1 wird hier eine Hilfsgröße  $d_y$  eingeführt.  $d_y$  ist das Verhältnis von Ratterfrequenz zu Eigenfrequenz und wird nur für diesen Fall verwendet.

$$d_y = \omega_c / \omega_{ny} \quad (2.26)$$

$$\epsilon = 2\pi - 2 \tan^{-1} \frac{d_y^2 - 1}{2\zeta_y d_y}$$

Nach der Berechnung von  $\epsilon$  kann wie beim System mit zwei Freiheitsgraden mit der Berechnung der Periodendauer der Zahneingriffsfrequenz und der Drehzahl (Gleichungen 2.21 bis 2.23) fortgefahren werden.

### 2.2.3 Superposition verschiedener Eigenformen

Bisher wurde immer nur der erste Eigenmode berücksichtigt. Um nun die Stabilitätskarte für mehrere Eigenmoden zu zeichnen, müssen mit dem gleichen Schema mehrere Stabilitätskarten berechnet und superponiert werden [(Brecher und Weck, 2017), (Li, Wang und Shi, 2017)].

Die Berechnung der Stabilitätskarten unter der Berücksichtigung endlich vieler Eigenmoden, welche aus drei grundlegenden Schleifen besteht, soll in Abbildung 2.7 verdeutlicht werden. Die innerste Schleife berechnet die axiale Schnitttiefe und Drehzahl mit einem konstanten Setting an modalen Parametern und  $k = const.$ , jedoch

für unterschiedlichste Ratterfrequenzen. Dadurch entsteht ein Rattersack einer spezifischen Eigenform. Eine Schleife darüber wird stetig der Parameter  $k \in \mathbb{N}^+$ , aus Gleichung 2.21 sukzessive ganzzahlig erhöht. Hier kann zirka nach 12 bis 15 Wiederholungen abgebrochen werden, da diese Anzahl an Rattersäcken für einen Eigenmode mehr als ausreichend ist. Mit der am äußersten liegende Schleife werden die unterschiedlichen Eigenmoden berechnet. Das heißt, hier werden die modalen Parameter, wie zum Beispiel die Steifigkeit, Dämpfung, Dämpfungsgrad und schwingende Masse oder Eigenfrequenz variiert. In dieser Arbeit werden maximal drei Eigenmoden berücksichtigt, da das Ergebnis durch die vielen notwendigen Abschätzungen für jeden weiteren Mode noch unaussagekräftiger wird.

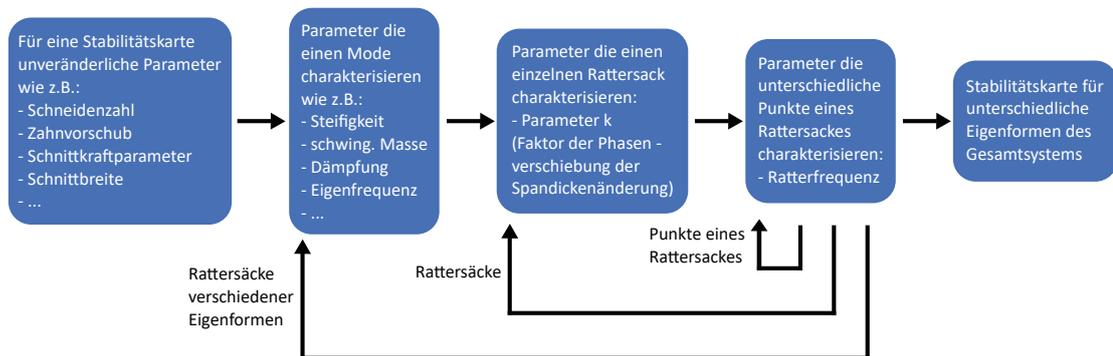


ABBILDUNG 2.7: Drei Schleifen des Berechnungsalgorithmus

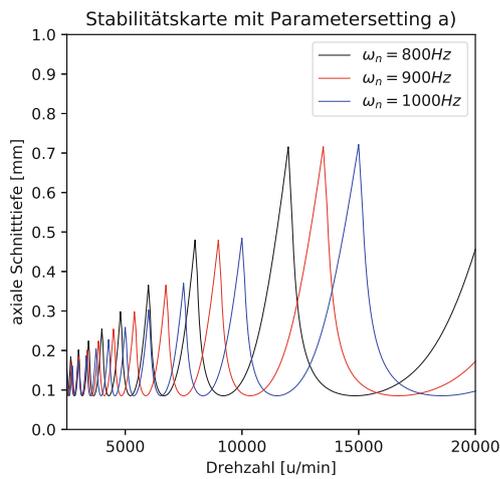
## 2.3 Verifikation des Modells

Um zu überprüfen, dass die Theorien richtig interpretiert wurden und der in Kapitel 2.2 beschriebene Algorithmus richtig in Python umgesetzt wurde, sollen vorhandene Stabilitätskarten nachgebildet werden. Dafür wird zum einen auf fach einschlägige Publikationen und zum anderen auf Simulationssoftware zurückgegriffen. Erst danach werden Stabilitätskarten durch experimentelle Versuche verifiziert.

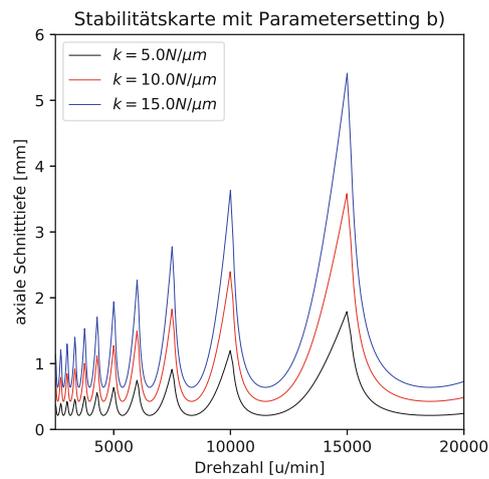
Der Vorteil an der Programmierung des Algorithmus in Python ist, dass es praktisch keine Schnittstellenprobleme mit potentiellen Regelungseinheiten geben kann, da es sich um eine etablierte, von vielen Systemen unterstützte und frei zugängliche Programmiersprache handelt.

### 2.3.1 Vergleich der Ergebnisse des abgeleiteten Modells mit jenen aus einer Parameterstudie

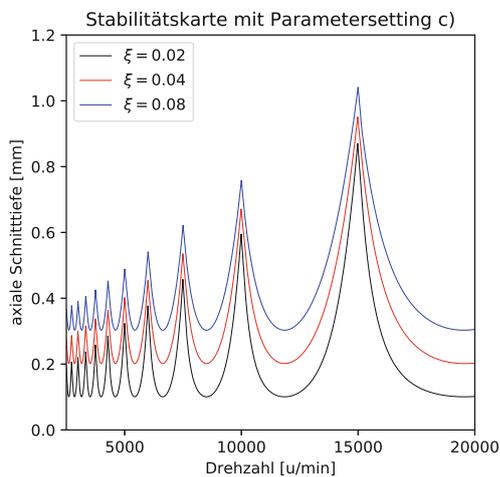
In diesem Abschnitt wird das Python Skript mit den in [Li, Wang und Shi, 2017] angeführten Parametern ausgeführt und die Ergebnisse, laut Abbildungen 2.8a bis 2.8d geplottet. Es entstehen die gleichen Stabilitätskarten wie in der erwähnten Literatur. Aus diesem Grund kann davon ausgegangen werden, dass der Algorithmus richtig verstanden und umgesetzt wurde.



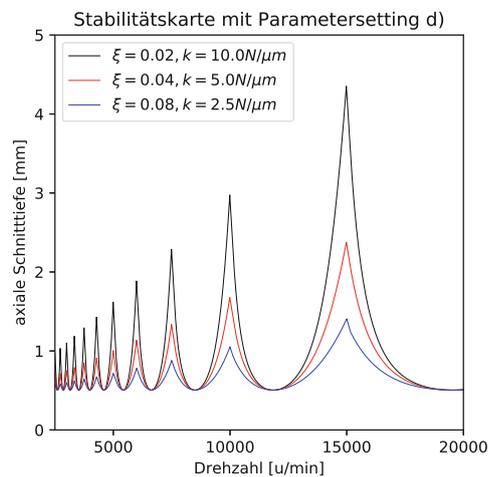
(A) Variation der Eigenfrequenz



(B) Variation der Steifigkeit



(C) Variation der Dämpfung



(D) Variation der dynamischen Steifigkeit

ABBILDUNG 2.8: Selbst entwickelte Stabilitätsdiagramme mit Parametern nach [Li, Wang und Shi, 2017]

Die Parameterstudie von [Li, Wang und Shi, 2017] zeigt, wie sich die Stabilität im Fräsprozess ändert, wenn sich die modalen Parameter ändern. Die wichtigste Erkenntnis hier ist, dass die stabilsten und instabilsten Drehzahlbereiche rein von der Eigenfrequenz abhängig sind, sprich die horizontale Lage der Rattersäcke ist alleine durch die Eigenfrequenz definiert. Das ist auch in Gleichung 2.28 erkennbar, da die Drehzahlen der maximalen Grenzschnitttiefen direkt proportional zur Eigenfrequenz sind. Dieser Effekt wird beim Aufstellen der Regelungsalgorithmen von großer Bedeutung sein, da die alleinige Information der Eigenfrequenz(en) reicht, um zu wissen, wo die stabilsten Drehzahlbereiche liegen.

Die vertikale Lage der Rattersäcke ist von den modalen Parametern und der radialen Schnittbreite abhängig. Die Grenzschnitttiefe ist jedoch stark davon abhängig, um welchen Parameter es sich handelt. Wird zum Beispiel die Steifigkeit (Abbildung 2.8b)

erhöht, so erhöht sich die Grenzschnitttiefe homogener, als bei der Erhöhung der Dämpfung (Abbildung 2.8c). Die Dämpfung wirkt sich auf die minimalen Grenzschnitttiefen deutlich stärker aus, als auf die maximalen Grenzschnitttiefen. Bei der Änderung der radialen Schnittbreite, sieht es wiederum ähnlich aus wie bei der Änderung der Steifigkeit.

Eine weiterführende Erklärung der Begriffe *minimale Grenzschnitttiefe* und *maximale Grenzschnitttiefe* ist in Abschnitt 2.4 zu finden.

Für die Variation des radialen Eintauchwinkels kann auch das Schnittbreiten-Durchmesser-Verhältnis  $a_e/D$ , laut Gleichung 2.27 gewählt werden. Dieser Formalismus ist etwas anschaulicher, da bei zwingenderweise konstantem Durchmesser des Werkzeuges nur die radiale Schnittbreite variiert wird.

$$\frac{a_e}{D} = \frac{1 - \cos(\phi_{exit})}{2} \quad (2.27)$$

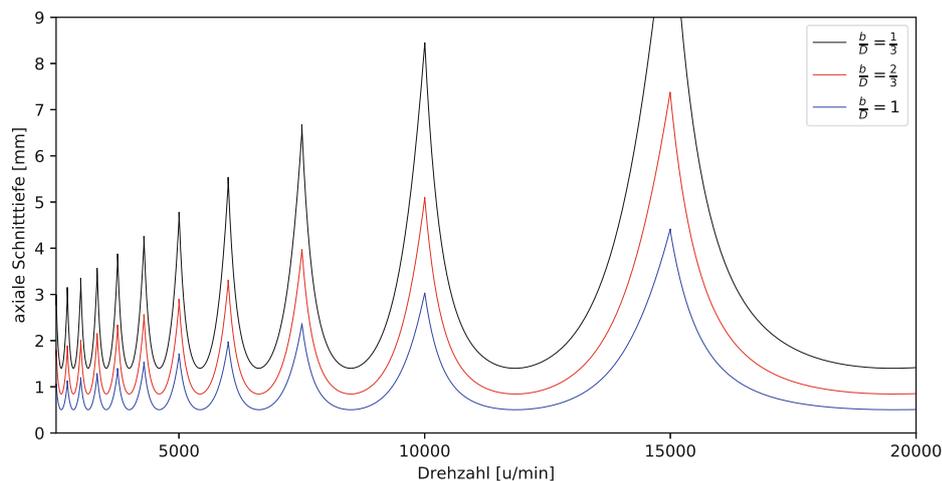


ABBILDUNG 2.9: Variation der radialen Schnittbreite ähnlich wie in [Li, Wang und Shi, 2017]

### 2.3.2 Vergleich mit Software CutPro

CutPro ist eine von dem kanadischen Unternehmen *Manufacturing Automation Laboratories* (<https://www.malinc.com/about-us/companyoverview/>) entwickelte Software, welche es unter anderem ermöglicht Stabilitätskarten zu berechnen. Die in der Software implementierte Erstellung von Stability-Lobe Diagrammen baut im einfachsten Falle des Einmassenschwingers auf dem zuvor beschriebenen Algorithmus auf. Der einzig wesentliche Unterschied ist, dass in dieser Software eine große Datenbank an Materialkennwerten und Erfahrungswerten vorliegt, was auch dazu führt, dass es leichte Unterschiede in der vertikalen Verschiebung der Rattersäcke gibt. Dieser Unterschied ist in Abbildung 2.10 erkennbar. Er ist dadurch entstanden, dass in CutPro Material, Zahnvorschub, Schneidenwerkstoff, etc. extra angegeben werden muss und

in dem hier implementierten Modell direkt der tangentialen und radiale Schnittkraftkoeffizient, welche von diesen Parametern abhängig sind, abgeschätzt werden. Dieser Vergleich ist eine weitere Bestätigung für die Richtigkeit des in Python umgesetzten Algorithmus.

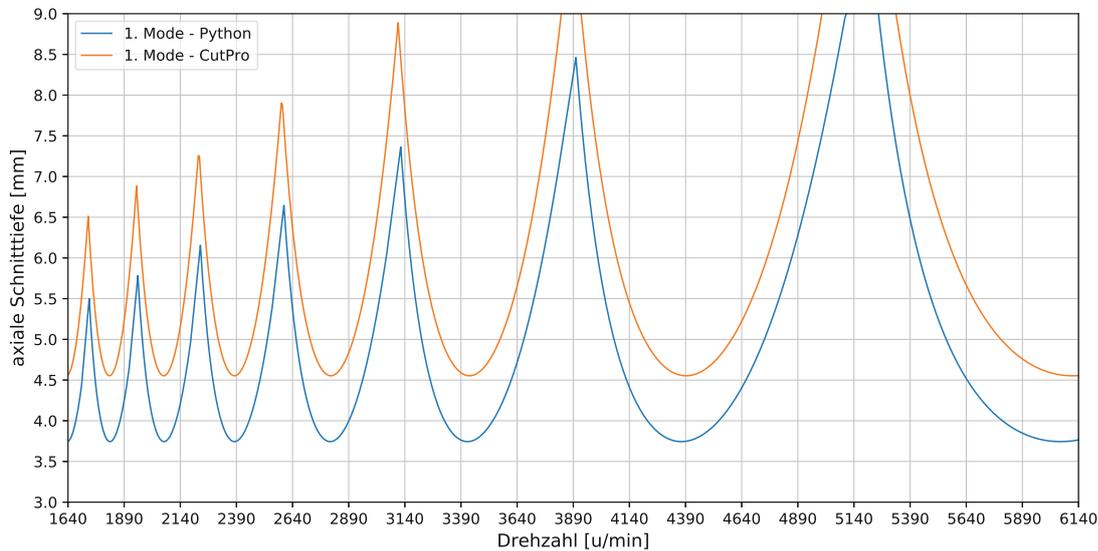


ABBILDUNG 2.10: Vergleich mit dem Ergebnis von CutPro ©

## 2.4 Schlussfolgerungen aus dem Stabilitätsalgorithmus

Es gibt in diesem System drei relevante Frequenzen: die Eigenfrequenz, die Zahn-eingriffsfrequenz bzw. die Umdrehungsfrequenz und die Ratterfrequenz. Es wird in diesem Abschnitt erklärt, wie die am wenigsten stabilen Drehzahlen, der minimalen Grenzschnitttiefen und die stabilsten Drehzahlen, die der maximalen Grenzschnitttiefe berechnet werden. In Abbildung 2.11 sind diese Grenzschnitttiefen exemplarisch dargestellt. Hier ist zu sehen, dass die minimalen Grenzschnitttiefen (grün) sich zwar bei unterschiedlichen Drehzahlen befinden, der Wert der minimalen Grenzschnitttiefe jedoch eine Konstante ist. Die maximalen Grenzschnitttiefen hingegen, steigen von Rattersack zu Rattersack jedoch linear an. Weiters gibt es Überlegungen darüber, was geschieht, wenn gewisse Parameter spezielle Werte annehmen beziehungsweise diese singular, etc. werden.

### 2.4.1 Drehzahlbereiche bei maximaler Grenzschnitttiefe

Die maximale Grenzschnitttiefe ist im Gegensatz zur minimalen Grenzschnitttiefe von  $k$ , aus Gleichung 2.21, abhängig. Sie liegt genau auf dem Schnittpunkt vom  $[n]$ -ten Rattersack auf den  $[n+1]$ -ten Rattersack. Diese stabilen Bereiche werden dann erreicht, wenn die Zahn-eingriffsfrequenz gleich der Eigenfrequenz des Systems ist. Diese Aussage erscheint auf den ersten Blick entgegen der Intuition. Deswegen sei an dieser

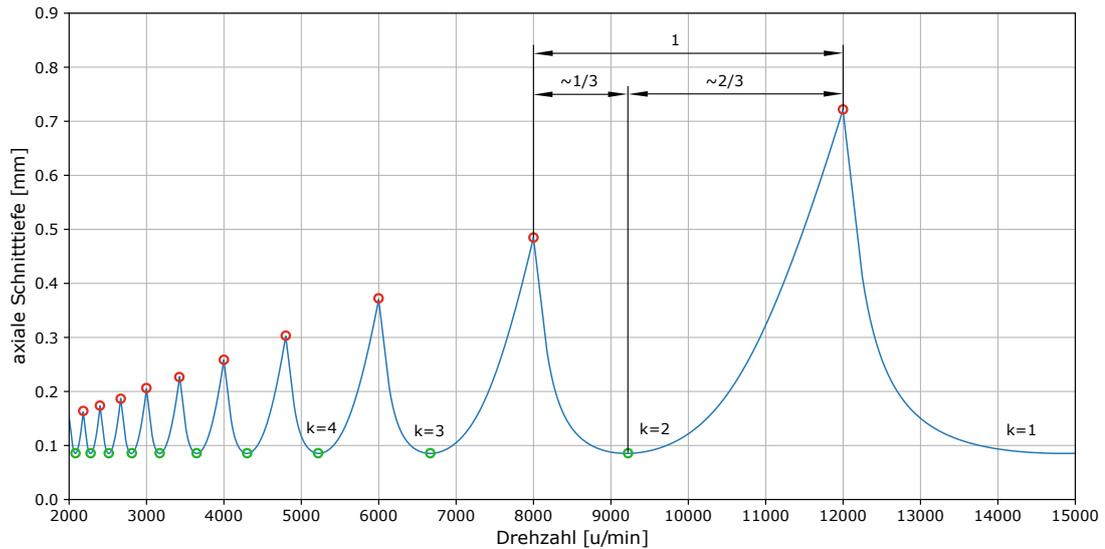


ABBILDUNG 2.11: Stabilitätskarte mit min./max. Grenzschnittiefen

Stelle nochmals erwähnt, dass diese Folgerungen für den Schwingungsfall des Ratters und die herleitbaren Aussagen bezüglich Ratter-Stabilität gelten. Auslenkungen durch die deutlich geringere periodische Fremderregung werden hier vernachlässigt. In Gleichung 2.28 wird gezeigt, dass die Drehzahlen der maximal erreichbaren Schnitttiefen, unter der Voraussetzung der Berücksichtigung eines Eigenmodes, nur von der Eigenfrequenz und der Zähnezahl abhängig sind. Dieser Effekt wird in Kapitel 2.3.1 noch einmal graphisch verdeutlicht. Der Betrag der maximalen Grenzschnitttiefe kann mit dem oben genannten Schema natürlich berechnet werden, ist jedoch für die Regelung eher uninteressant, da die axiale Schnitttiefe meist ein nicht regelbarer Parameter ist. Jedoch sind die jeweiligen Drehzahlen bei den maximalen Grenzschnitttiefen von immenser Bedeutung, da die Drehzahl regelbar ist. Würde in Zukunft ein instabiler Prozess auftreten, so könnte auf diese relativ stabilen Drehzahlen geregelt werden. Gleichung 2.28 ist eine Berechnungsformel für diese stabilsten Drehzahlen.

$$\begin{aligned}\omega_{ZEF} &= \omega_n \\ n &= \frac{60\omega_n}{kN}\end{aligned}\quad (2.28)$$

#### 2.4.2 Drehzahlbereiche bei minimaler Grenzschnitttiefe

Die minimale Grenzschnitttiefe wird erreicht, wenn die Ratterfrequenz gleich der Eigenfrequenz des Systems ist. Das hat zur Folge, dass der Realteil der Übertragungsfunktion laut Gleichung 2.14 gegen Null geht. Der Wert der minimalen Grenzschnitttiefe ist von der Variablen  $k$  und somit von jedem einzelnen Rattersack unabhängig. Jeder Rattersack hat laut Theorie das gleiche globale und lokale Minimum. In Gleichungssatz 2.29 wird gezeigt, wie die minimale Grenzschnitttiefe direkt berechnet werden kann. Die Drehzahl muss wieder nach Gleichungen 2.20 bis 2.23

berechnet werden.

$$\begin{aligned}
 \omega_c &= \omega_n \\
 \operatorname{Re}(G) &= 0 \\
 \operatorname{Im}(G) &= -\frac{1}{2k_m\zeta_m} \\
 a_0 &= -\frac{1}{4k_m^2\zeta_m^2} (\alpha_{xx}\alpha_{yy} - \alpha_{xy}\alpha_{xx}) = -\frac{1}{4k^2\zeta^2} \alpha_0 \\
 a_1 &= -\frac{1}{2k_m\zeta_m} (\alpha_{xx} + \alpha_{yy}) = -\frac{1}{2k\zeta} \alpha_1 \\
 \operatorname{Re}(\Lambda) &= -k\zeta \frac{1}{\alpha_0} \sqrt{-\alpha_1^2 + 4\alpha_0} \\
 \operatorname{Im}(\Lambda) &= -k_m\zeta_m \frac{1}{\alpha_0} \alpha_1 \\
 \kappa &= \frac{\sqrt{-\alpha_1^2 + 4\alpha_0}}{\alpha_1} \\
 \kappa &= f(\phi, K_r) \\
 a_{lim} &= \frac{2\pi\alpha_1}{NK_{tc}\alpha_0} k_m\zeta_m \left( \kappa + \frac{1}{\kappa} \right)
 \end{aligned} \tag{2.29}$$

Für eine grobe und einfache Abschätzung der am wenigsten stabilen Drehzahlen, kann angenommen werden, dass diese sich zirka 1/3 vom linken Maximum eines Rattersackes entfernt befinden. (siehe Abbildung 2.11)

$$\begin{aligned}
 n &= \frac{1}{3} \left[ \frac{60\omega_n}{kN} - \frac{60\omega_n}{(k+1)N} \right] + \frac{60\omega_n}{(k+1)N} \\
 n &= \frac{1}{3} \frac{60\omega_n}{kN} + \frac{2}{3} \frac{60\omega_n}{(k+1)N} \\
 n &= \frac{60\omega_n(3k+1)}{3kN(k+1)}
 \end{aligned} \tag{2.30}$$

### 2.4.3 Weitere Überlegungen

- $\omega_c = \omega_{ZEF}$  beziehungsweise  $\epsilon = 0$  deg

Für die minimalen und maximalen Grenzschnittiefen wurden zuvor jeweils zwei der drei unterschiedlich auftretenden Frequenzen gleichgesetzt. Es stellt sich die Frage, was passiert, wenn die Zahneingriffsfrequenz gleich der Ratterfrequenz ist. Dieser Fall wurde hier bis jetzt noch nicht erläutert. Wenn gilt,  $\omega_{ZEF} = \omega_c$ , dann gilt für Gleichung 2.22, dass der Kehrwert der Periodendauer der Zahneingriffsfrequenz gleich der Ratterfrequenz ist. Dies führt dazu, dass der Parameter  $\epsilon$  in Gleichung 2.20, gegen  $-2k\pi$  geht. Sprich die Phasenverschiebung der Spannungsdickenänderung ist gleich Null. Das bedeutet wiederum, dass

laut Gleichung 2.20, der Parameter  $\kappa$  gegen Null geht und somit der Realteil des Eigenwertes ( $Re(\Lambda)$ ) ebenfalls gegen Null geht. Das hätte zur Folge, dass in Gleichung 2.19 die Grenzschnittiefe  $a_{lim}$  singular wird. Dieser Fall kann dementsprechend nicht eintreten.

Der hier beschriebene Formalismus wird in den Gleichungen 2.31 beschrieben.

$$\begin{aligned}\frac{2\pi}{T_{ZEF}} &= \omega_c \\ \omega_c T_{ZEF} &= \epsilon + 2k\pi \\ \epsilon &= -2k\pi = 0 \text{ deg} \\ \kappa &= 0 \\ Re(\Lambda) &= 0 \\ a_{lim} &\rightarrow \infty\end{aligned}\tag{2.31}$$

- $\epsilon = 180 \text{ deg}$  beziehungsweise  $Im(\Lambda) = 0$   
In Gleichung 2.20 ist erkennbar, dass  $Im(\Lambda)$  nur dann Null wird, wenn die Phasenverschiebung der Spannungsdickenänderung gleich  $180 \text{ deg}$  ist. Das bedeutet, dass für  $Re(\Lambda) > 0$  die Grenzschnittiefe reelle Werte größer Null annimmt.

## 2.5 Erkenntnisse aus den Stabilitätskarten für die Regelung

Die erste Erkenntnis ist das Vorhandensein stabiler Bereiche bei unterschiedlichen axialen Schnitttiefen und Drehzahlen. Auch der Umstand, dass ein instabiler Prozess nicht nur durch Reduktion der Drehzahl stabilisiert werden kann, sondern auch durch die Erhöhung der Spindeldrehzahl, was einer Produktivitätssteigerung entsprechen würde. Voraussetzung für diese Aussage ist ein konstanter Zahnvorschub. Weiters ist das Wissen von der Existenz der stabilsten Drehzahlen und deren alleiniger Abhängigkeit von den Eigenfrequenzen der Eigenformen von immenser Bedeutung. Da es bei dem vorliegenden ICOTronic System nur möglich ist die Drehzahl und den Vorschub in der in-Prozess-Regelung zu beeinflussen, sind gerade die tendenziellen stabilen und instabilen Drehzahlen wichtig. Die Beeinflussung der axialen Schnitttiefe ist im Prozess nicht möglich, da hier ein Eingriff in den G-Code nötig wäre und dies nur schwer umsetzbar wäre, da die Werkstückgeometrie hierfür verändert werden würde. Lediglich beim Schruppen könnte die Bearbeitungszeit verkürzt werden, indem die Anzahl der Schruppyklen durch eine höhere axiale Schnitttiefe, verringert wird.

In weiterer Folge werden mit dem hier beschriebenen Algorithmus Stabilitätskarten für Fräsversuche erstellt, die unter anderem dazu dienen die Klassifikation von stabilen und instabilen Schnitten zu erleichtern.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

## Kapitel 3

# Versuche

Die Versuchsdaten sind grundlegend notwendig, um zu überprüfen wie gut die analytisch berechneten Stabilitätskarten mit realen Prozessen übereinstimmen. Weiters dienen die gemessenen Beschleunigungswerte als Grundlage zur Erstellung von Stabilitätsindikatoren, welche aus dem Zeit- und Frequenzbereich abgeleitet werden. Diese Indikatoren beziehungsweise Features dienen als Eingangsgrößen des Machine Learning Algorithmus und in weiterer Folge zum Umsetzen von Regelungsstrategien. Dafür werden laut Versuchsplan im Anhang B und laut Tabelle 3.1 Bahnen in einen Aluminiumblock gefräst. Die Parameter axiale Schnitttiefe und Drehzahl werden bewusst so variiert, dass stabile, aber auch instabile Prozesse gemessen werden können.

### 3.1 Versuchsaufbau und Vorversuche

Der grundsätzliche Versuchsablauf sieht vor, dass zuerst ein Impulshammerversuch durchgeführt wird und anhand dieses die Parameter zur Erstellung der Stabilitätskarte abgeschätzt werden. Danach werden laut Versuchsplan Bahnen mittels Stirn-Umfangsfräsen in den Aluminiumblock gefräst. Diese Daten werden ausgewertet, gespeichert und klassifiziert.

#### 3.1.1 Aufbau und verwendete Maschinen

Die Versuche wurden mit den in Tabelle 3.1 angeführten Parametern durchgeführt. Das Datenblatt des Werkzeug-Grundkörpers und die dazu passenden Wendeschneidplatten sind in Anhang C zu finden.

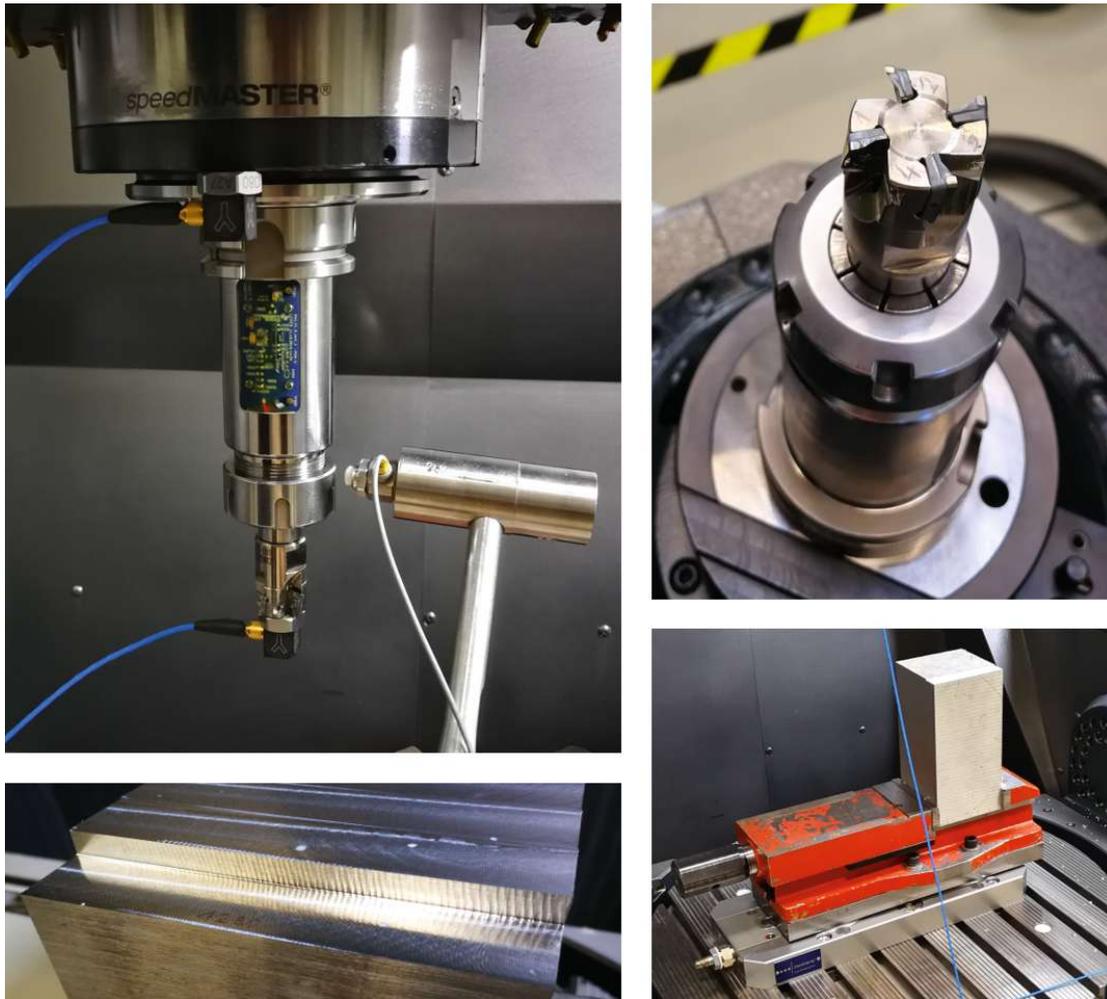


ABBILDUNG 3.1: Überblick des Versuchsaufbaus

Fräsmaschine	DMU 75 monoBLOCK
Werkzeug-Grundkörper	WEX 2025M12Z4
Werkzeug-Wendeschnidplatten	AXET 123508 PEFR-S H1:WED
Werkzeug-Durchmesser	25mm
Schneidenzahl	4
Werkstück-Material	EN AW-5083 (AlMg4,5Mn)
Werkstück-Querschnitt	100x100
Kühlschmierstoff	40l/min mit 3,5bar
Spindeldrehzahl	variiert von 3890-9140 u/min
Zahnvorschub	0.1mm
Axiale Schnitttiefe	variiert von 1-7mm
Radiale Schnittbreite	20mm

TABELLE 3.1: Versuchsparameter

Beschleunigungsmessung	Sensorischer Werkzeughalter HSK63
Stationäre Beschleunigungsmessung	Triaxial ICP® Accelerometer 356A16
Krafterregung	Impulshammer mit Kraftmessdose

TABELLE 3.2: Verwendete Sensorik

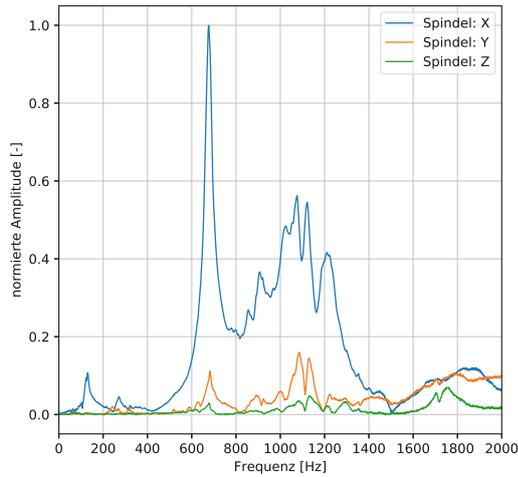
### 3.1.2 Bestimmung der Eigenfrequenzen mittels Impulshammerversuch

Für die Beurteilung der Stabilität beziehungsweise für die Erstellung von Stabilitätskarten spielen die Eigenfrequenzen der Eigenmoden eine wichtige Rolle. Hierfür soll das Übertragungsverhalten des Systems mittels Impulshammerversuch bestimmt werden. Der Impulshammer wird in [Baur, 2013] wie folgt beschrieben: *„Der Impulshammer ist ein hammerförmiger Erreger mit auswechselbaren Schlageinsätzen und einer integrierten Kraftmesszelle, welche den zeitlichen Verlauf der Kraft beim Aufschlag misst.“* Der Ablauf des Versuches sieht so aus, dass zuerst mit dem Hammer auf die Werkzeugaufnahme geschlagen wird und die Stoßanregung über den im Hammer verbauten Sensor gemessen wird. Zeitgleich wird mit stationärer Beschleunigungsmessung die Schwingungsantwort des Systems gemessen. Aus diesen beiden Signalen wird die Übertragungsfunktion gebildet. Für die in diesem Versuch verwendete Konfiguration in Abbildung 3.2a und 3.2c für die x-Richtung und Abbildung 3.2b und 3.2d für die y-Richtung.

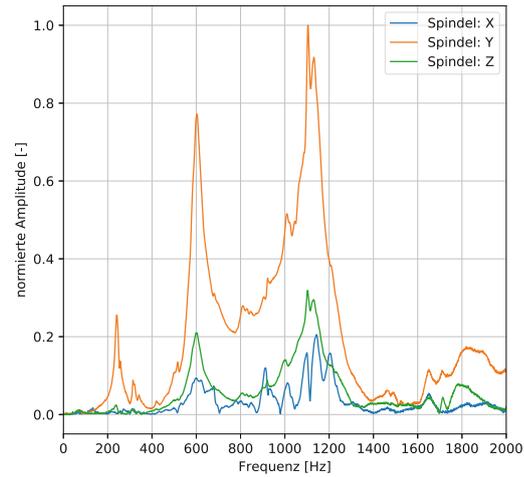
### 3.1.3 Abschätzung von Eigenfrequenz und Gesamtsteifigkeit des Systems

Für lange auskragende Werkzeuge kann die Steifigkeit über die Durchbiegung eines modellhaft angenommenen Kragträgers, mit einer Einzellast am Ende, bestimmt werden. Dafür wird die Formel für die Durchbiegung auf Gleichung 3.2 umgeformt. Die Flächenträgheitsmomente, welche zur Bestimmung der Einzelsteifigkeiten von Fräser und Werkzeughalter benötigt werden, werden laut Gleichung 3.1 berechnet. Für den Kerndurchmesser des Ersatzstabes muss ein kleinerer Durchmesser als der der Außendurchmesser der Werkzeugschneiden angenommen werden. Laut Gleichung 3.3 wird der Kehrwert der Gesamtsteifigkeiten gebildet, indem die Kehrwerte der einzelnen Steifigkeiten im System aufsummiert werden [Bleicher, 2019b]. In diesem Fall werden Steifigkeiten von Fräser und Werkzeughalter berücksichtigt. Dies ist zielführend, da die Steifigkeit eines langen Werkzeuges in Verbindung mit einem längeren Werkzeughalter deutlich geringer ist als die Steifigkeiten anderer Komponenten im System und somit diese ausschlaggebend ist. Um die Steifigkeiten der restlichen Komponenten nicht zu vernachlässigen, können diese mit einem Faktor (z.B.: 0,9) bei der Gesamtsteifigkeit mitberücksichtigt werden. Dieser Faktor muss kleiner als 1 sein, da die Gesamtsteifigkeit vom schwächsten Element nach oben limitiert wird. Daher kann durch die Berücksichtigung von anderen Steifigkeiten im System die Gesamtsteifigkeit nur kleiner werden.

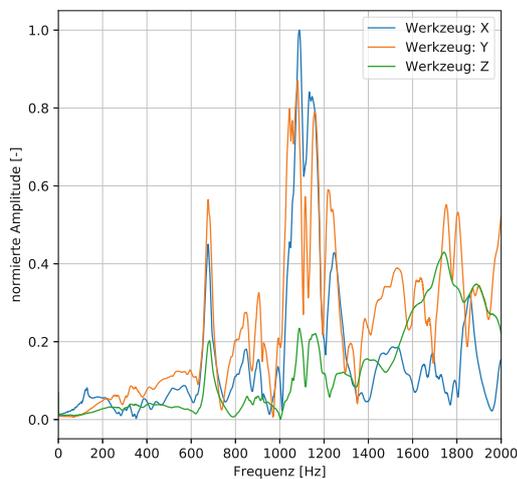
$$I = \frac{d^4 \pi}{64} \quad (3.1)$$



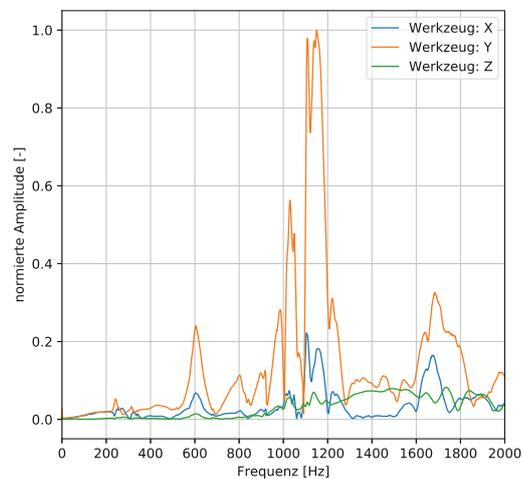
(A) Anregung der Spindel in x-Richtung



(B) Anregung der Spindel in y-Richtung



(C) Anregung des Werkzeuges in x-Richtung



(D) Anregung des Werkzeuges in y-Richtung

ABBILDUNG 3.2: Ermittelte Übertragungsfunktionen des Impulshammerversuches

$$k_m = \frac{F}{w} = \frac{3EI}{l^3} \quad (3.2)$$

$$\frac{1}{k_{m_{ges}}} = \sum_{i=0}^N \frac{1}{k_{m_i}} \quad (3.3)$$

Die Abschätzung der ersten Eigenfrequenz über die schwingende Masse und Gesamtsteifigkeit ist zu ungenau und wird deshalb nur erwähnt, aber nicht zur weiteren Beurteilung herangezogen. Die Eigenfrequenz lässt sich dann nach Gleichung 3.4 berechnen. Um das Ergebnis in der gängigeren Einheit Hertz zu erhalten muss laut Gleichung 3.5 das vorherige Ergebnis durch  $2\pi$  dividiert werden.

$$\omega_n = \sqrt{\frac{k_m}{m_m}} \quad (3.4)$$

$$f_n = \frac{\omega_n}{2\pi} \quad (3.5)$$

Allgemein ist zu sagen, dass SI-einheitenkonform gerechnet wird und Frequenzen daher immer in rad/s einzusetzen sind. Einheiten wie Hz oder u/min dienen lediglich zu besserer Veranschaulichung.

### 3.1.4 Erstellung einer Versuchs-Stabilitätskarte

Für Versuchsreihe I (siehe Tabelle 3.4) wird eine Stabilitätskarte geplottet, welche die ersten drei möglichen Eigenmoden beinhaltet. Anzumerken ist jedoch, dass es sich nur um Eigenmoden des Werkzeuges handelt, da mit dem Impulshammer nur die werkzeugseitigen Eigenfrequenzen untersucht werden. Aus den durch den Impulshammer ermittelten Übertragungsfunktionen werden die Eigenfrequenzen ausgewertet und in weiterer Folge so angepasst, dass Stabilitätskarte mit den Ergebnissen aus der Versuchsreihe I bestmöglich übereinstimmt (siehe Abbildung 3.6). Die dabei verwendeten Eigenfrequenzen sind aus Tabelle 3.3 zu entnehmen. Da der Aluminiumblock von Schnitt zu Schnitt kleiner wird, ändern sich teilweise auch die modalen Parameter, welche jedoch vereinfachend als konstant angenommen werden. Aus diesem Grund wird für Versuchsreihe II auch auf die Stabilitätskarte verzichtet, da sich hier die Rattersäcke mit den Messungen nur schwer in Einklang bringen ließen.

	x-Richtung	y-Richtung
Eigenfrequenz-Mode 1	1020 Hz	1050 Hz
Eigenfrequenz-Mode 2	680 Hz	600 Hz
Eigenfrequenz-Mode 3	860 Hz	860 Hz

TABELLE 3.3: Verwendete Eigenfrequenzen

## 3.2 Versuchsdurchführung

Bei der Versuchsdurchführung ist darauf zu achten, dass alle Informationen so detailliert wie möglich aufgenommen werden. Das hierfür verwendete Versuchsprotokoll ist in Anhang B zu finden. Um eine objektive Bewertung nach den Versuchen zu garantieren, werden die Versuche nicht nur mit dem Sensorischen Werkzeughalter, sondern auch mit einem Luftschallmikrofon aufgezeichnet. Da das Luftschallmikrofon (Abbildung 3.4) nicht spritzwassergeschützt ist, wird dieses durch eine Isolierung gegen den Einfluss von der Kühlschmierstoff-Emulsion geschützt. Das Mikrofon wird in diesem Zustand getestet, mit dem Resultat, dass dies keinen beziehungsweise nur einen verschwindend geringen Einfluss auf die Schallmessung hat. Ebenso werden nach jedem Schnitt Fotos von den entstehenden Oberflächen angefertigt. Die Oberfläche dient als Indikator zur Klassifikation der Stabilität, da hier Rattermarken deutlich ersichtlich sind. Bei den fehlenden Nummern zwischen der Versuchsreihe I und Versuchsreihe II

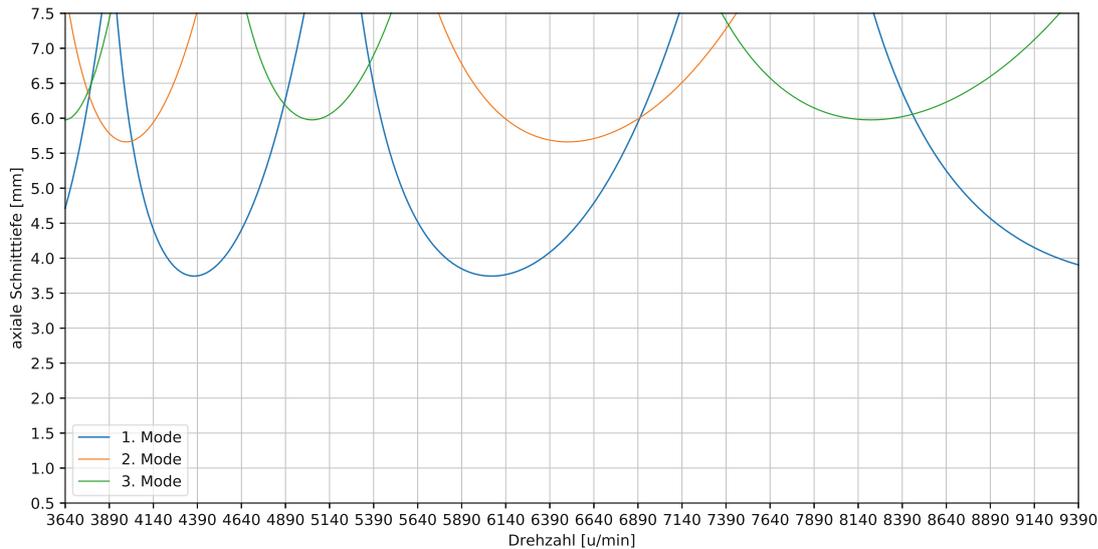


ABBILDUNG 3.3: Berechnete und an die Versuchsreihe I angepasste Stabilitätskarte

handelt es sich um nicht aussagekräftige Versuche die mit einer zu geringen Abtastrate (870 Hz) aufgenommen wurden.

Versuchsreihe	Versuchsnummern	gemessene Achsen	Abtastrate $f_s$ [Hz]
Versuchsreihe I	0700-1310	x-Achse	4070
Versuchsreihe II	1840-2160	x-, y-, z-Achse	3160

TABELLE 3.4: Durchgeführte Versuche

Die in Abbildung 3.5 dargestellten Diagramme dienen zur besseren Übersicht über die Versuchsreihen. Zu den einzelnen Punkten werden die Testnummern hinzugefügt, welche mit denen im Versuchsprotokoll in Anhang B ident sind.

### 3.3 Ergebnisse und Auswertung

In den folgenden Abschnitten sollen einzelne Versuche gegenübergestellt und verglichen werden. Explizit werden in Tabelle 3.5 Schnitte mit geringer Drehzahlabweichung, aber unterschiedlicher Stabilität miteinander verglichen. Die konstanten Parameter dieser Reihen können der Tabelle 3.1 entnommen werden. Das Außergewöhnliche an diesen Reihen ist, dass sich die Oberflächen und die ausgewerteten Signale stark voneinander unterscheiden, obwohl der einzige Unterschied eine Drehzahldifferenz von 250 u/min ist. In diesen Versuch ist der scharfe Übergang vom stabilen in den instabilen Bereich und umgekehrt, der auch als solcher in Stabilitätskarten durch eine Linie dargestellt wird, erkennbar. Wird also nahe der Eigenfrequenz angeregt, kann durch eine kleine Änderung im System, wie zum Beispiel das Erhöhen/Vermindern von Drehzahl und Vorschub, der Prozess instabil werden. Tabelle 3.6 beinhaltet zwei Schnitte, welche zeigen, dass für einen stabilen Prozess nicht zwangsläufig eine höhere

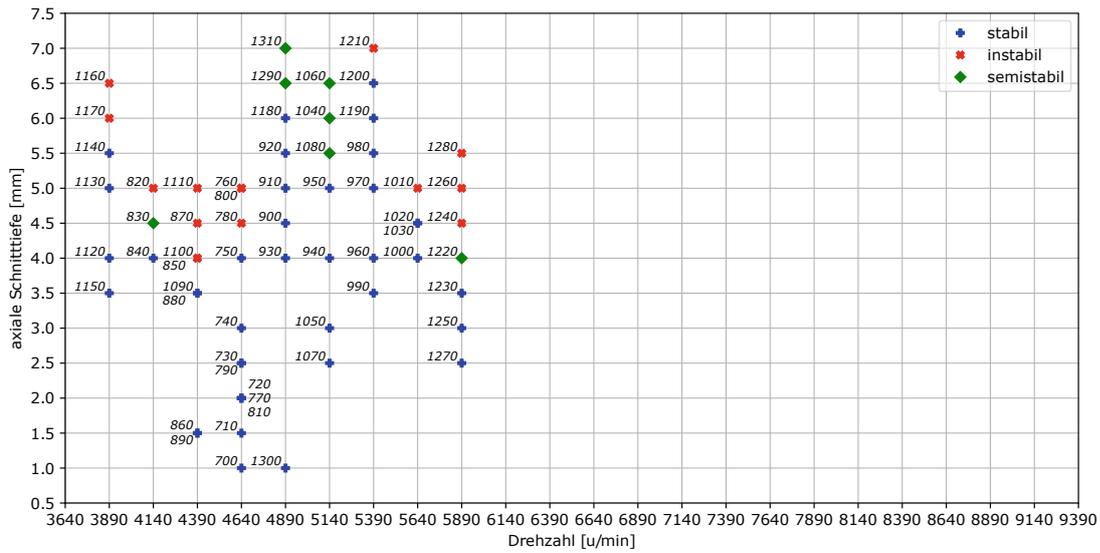


ABBILDUNG 3.4: Verwendetes Messmikrofon

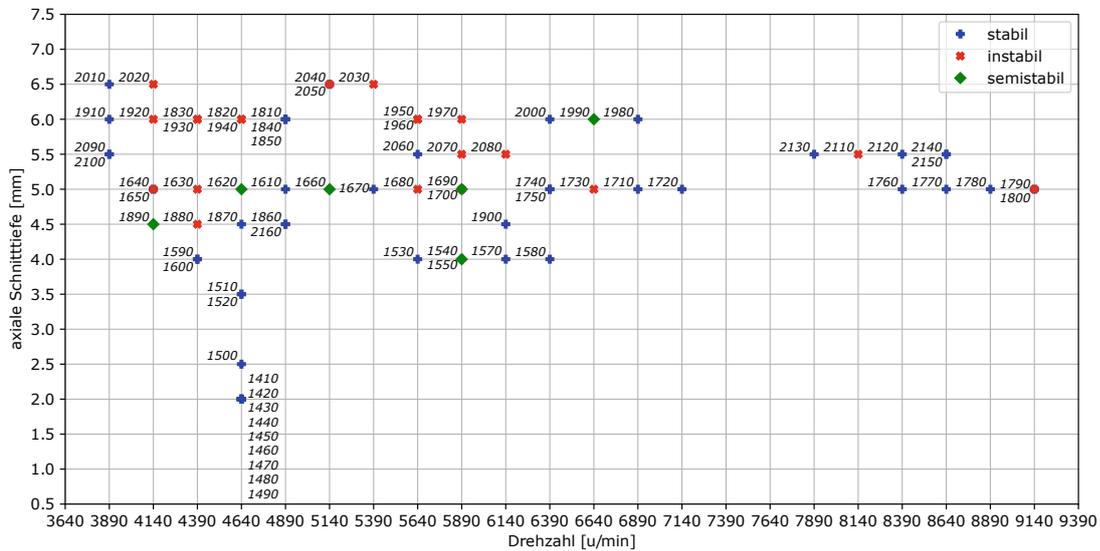
Bearbeitungszeit in Kauf zu nehmen ist. Der Versuch 1780 wird als stabiler Prozess bewertet, obwohl dieser nur die Hälfte der Bearbeitungszeit des instabilen Versuchs 1640 in Anspruch nimmt. Hier wird gezeigt, dass das Potenzial einer Beschleunigungsmessung mit kombinierter in-Prozess Regelung enorm ist. Der Nummerierung der Versuche wird keine spezielle Bedeutung zugesprochen.

Versuche	Drehzahl [u/min]	Axiale Schnitttiefe [mm]	Hauptzeit [s]	Klassifizierung
2010	3890	6.5	4.43	stabil
2020	4140	6.5	4.08	instabil
1740	6390	5.0	2.64	stabil
1730	6640	5.0	2.54	instabil
1710	6890	5.0	2.45	stabil
2130	7890	5.5	2.14	stabil
2110	8140	5.5	2.07	instabil
2120	8390	5.5	2.01	stabil

TABELLE 3.5: Vergleich des Übergangs zwischen stabilen und instabilen Systemverhaltens aus Versuchsreihe II



(A) Versuchssreihe I



(B) Versuchssreihe II

ABBILDUNG 3.5: Darstellung der Versuchsreihen

Versuche	Drehzahl [u/min]	Axiale Schnitttiefe [mm]	Hauptzeit [s]	Klassifizierung
1640	4140	5.0	4.08	instabil
1780	8890	5.0	1.90	stabil

TABELLE 3.6: Größtmögliche Reduktion der Hauptzeit bei gleichzeitiger Stabilisierung des Prozesses in Versuchsreihe I

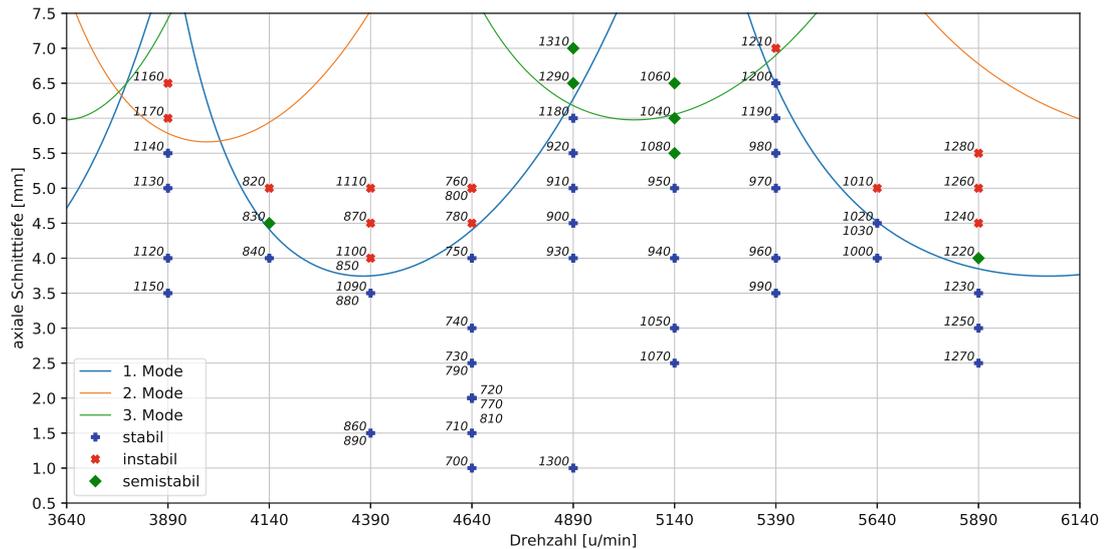


ABBILDUNG 3.6: Versuchsreihe I mit Stabilitätskarte

### 3.3.1 Klassifikation von stabilen und instabilen Prozessen

Um festzustellen, ob es sich um einen stabilen oder instabilen Schnitt handelt, werden mehrere Kriterien herangezogen. Als grobe Orientierung im Vorhinein wird die erstellte Stabilitätskarte zu Hilfe genommen. Ein weiteres Kriterium ist auch das rein subjektiv wahrgenommene Geräusch beim Zerspanen, welches beim Rattern als ein deutliches „Kreischen“ mit hörbar hoher Frequenz wahrgenommen werden kann. Zum anderen werden die Oberfläche kontrolliert und auf markante Rattermarken untersucht. Ein sehr gutes Beispiel hierfür sind die oben angesprochenen Versuche. In Abbildung 3.7 sind die Oberflächen zu den in Tabelle 3.5 angeführten Versuchen dargestellt. So ist bei den stabilen Schnitten eine deutlich glattere Oberfläche, frei von jeglichen Störungen oder Rauheitsextrema, erkennbar. Die instabilen Schnitte hingegen weisen deutlich Strukturen von Rattermarken auf. Die Oberfläche ist daher das Hauptunterscheidungsmerkmal zwischen stabilen und instabilen Schnitten.

Bei der Messung werden Schnitte die nicht eindeutig zuordenbar sind, als semistabil bezeichnet. Bei diesen Schnitten weist die Oberfläche zwar Rattermarken auf, diese sind meist jedoch weniger stark ausgeprägt. Auch das akustisch hörbare Signal ist bei diesen Versuchen schwächer. Aus diesem Grund werden die semistabilen Versuche später genau analysiert und abhängig vom Sensorsignal als stabil oder instabil klassifiziert. Da das spätere Machine Learning Modell nur zwischen stabil und instabil unterscheiden soll, ist das ein entscheidender Schritt. Die binäre Klassifikation macht deshalb Sinn, da es für die Regelung auch nur die zwei Zustände, Eingriff oder kein Eingriff, geben kann.



ABBILDUNG 3.7: Vergleich der Oberfläche mehrerer Schnitte

### 3.3.2 Gegenüberstellung von Signalen im Zeitbereich

Die Signale im Zeitbereich werden mit dem Sensorischen Werkzeughalter aufgezeichnet. Der Halter misst die Beschleunigungen in einer beziehungsweise drei Achsen. Im Sensorsignal der z-Achse, wie in den Abbildungen 3.8c und 3.9c dargestellt, ist deutlich erkennbar, dass Beschleunigungen in dieser Richtung für das Stabilitätsverhalten beim Stirn- und Umfangsfräsen im Allgemeinen keine Aussagekraft haben. Beim Fräsen in z-Richtung oder auch beim Bohren macht die 3-achsige Beschleunigungsmessung beziehungsweise die Beschleunigungsmessung in der Drehachse durchaus Sinn. Die Signalintensität bei der hier vorliegenden Anwendung ist jedoch so schwach ausgeprägt, dass der Eingriffszeitpunkt aus dem Signal in z-Richtung gar nicht erkennbar ist. Das Messrauschen dominiert hier in dieser Richtung das zu messende Signal.

Die beiden Achsen in radialer Richtung zeigen ähnliche Ergebnisse, jedoch ist die x-Achse aufgrund der Eigenschaften des Sensors etwas sensitiver. Allgemein startet die Aufzeichnung bei stehender Spindel, weshalb der Anfang des Signales reines Leersignal mit Messrauschen ist. Danach gibt es einen Anstieg, der aufgrund der außermittigen Lage des Sensors von der Drehachse durch Zentrifugalbeschleunigung

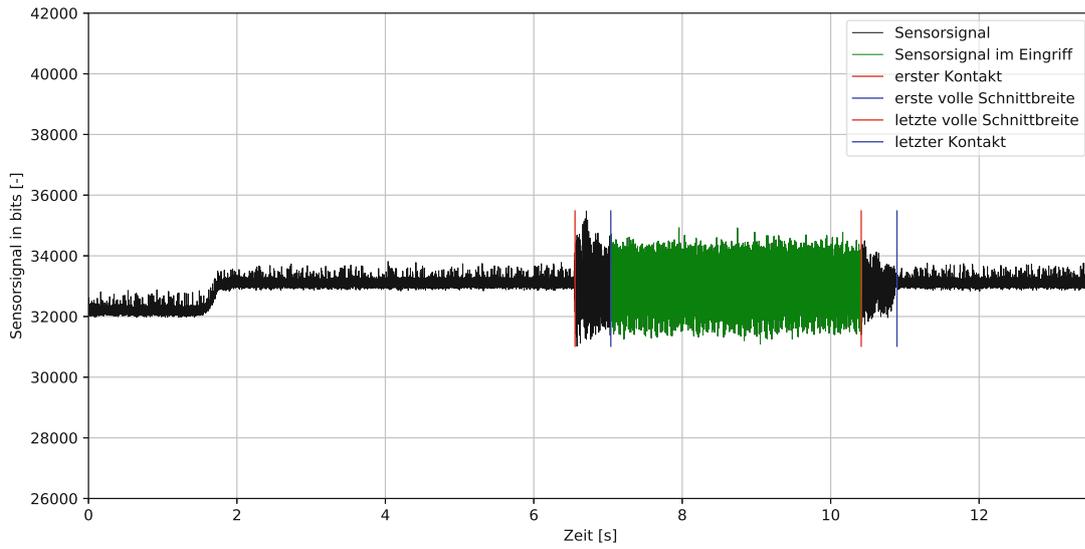
zustande kommt. Nach einer gewissen Zeit trifft eine Schneide des Werkzeuges das erste Mal auf das Werkstück. Dieser Zeitpunkt wird in Abbildung 3.8 und Abbildung 3.9 mit der ersten roten Linie dargestellt und mit Hilfe einer selbst-programmierten Funktion *timevalues()* (siehe Anhang A.3) von Python selbst ausgewertet. Diese automatische Eingriffserkennung vergleicht jeden Messwert mit dem darauffolgenden Messwert. Der erste Eingriff des Fräasers findet statt, wenn es einen Anstieg des Signales von  $\geq 2.5\%$  gibt. Dieser doch recht einfache Algorithmus spricht mit seinem Schwellwert nicht beim Hochdrehen der Spindel an, sodass wirklich nur der Werkzeugeingriff erkannt wird.

Die zeitlich danach folgenden Linien in rot und blau in den Abbildungen 3.8 und 3.9 werden unter Berücksichtigung der Werkstücklänge, Vorschubgeschwindigkeit und Abtastrate berechnet. Bei der ersten blauen Linie ist der Fräser das erste Mal mit der radialen Schnittbreite von 20mm im Eingriff und bei der zweiten roten Linie ist er das letzte Mal mit dieser Schnittbreite im Eingriff. Die zweite blaue Linie charakterisiert den Zeitpunkt, an dem der Fräser vollständig aus dem Werkstück ausgetreten ist - sprich, wenn der Schnitt komplett vollzogen ist. Die Gesamtdauer des Schnittes, sprich die Hauptzeit, kann mit Gleichung 3.6 berechnet werden. Der Parameter  $L = 100\text{mm}$  ist in diesem Fall die Werkstücklänge und da die radiale Schnittbreite größer als die Hälfte des Fräserdurchmessers ist beträgt die gesamte Eingriffslänge  $100\text{mm} + \frac{25\text{mm}}{2} = 112,5\text{mm}$ . Die Vorschubgeschwindigkeit  $v_f$  ist bei konstant gehaltenem Zahnvorschub linear von der Drehzahl abhängig, weshalb jeder Versuch mit anderer Drehzahl auch eine andere Hauptzeit aufweist. Diese Gleichung berechnet die Hauptzeit rein aus den Fertigungsparametern und stimmt genau mit den Messungen überein.

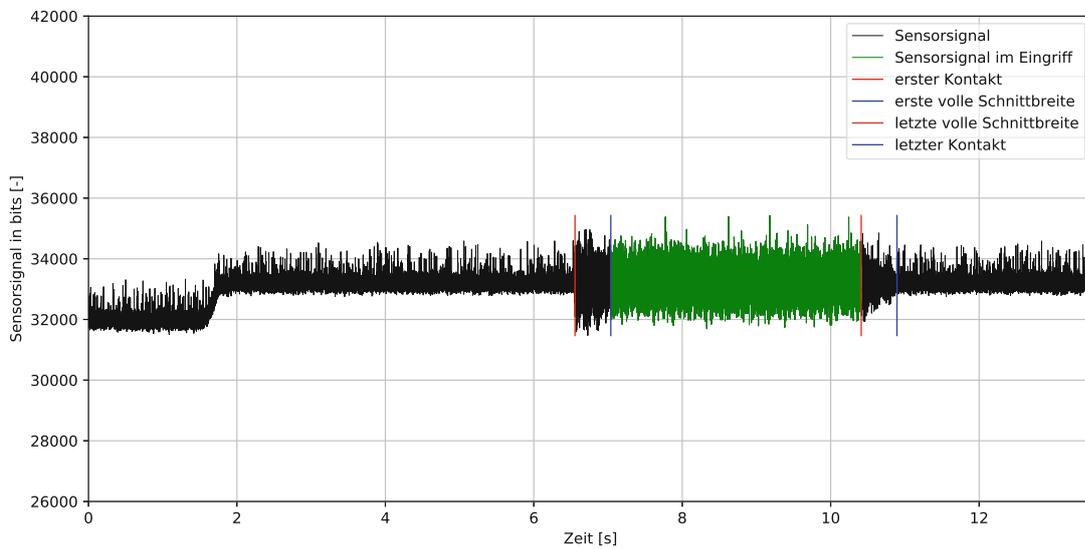
$$t_H[s] = \frac{(L + D/2)[\text{mm}] \cdot 60[\text{s}/\text{min}]}{v_f[\text{mm}/\text{min}]} \quad (3.6)$$

Für die spätere Betrachtung im Frequenzbereich wird nur der grüne Bereich zwischen erster blauen und zweiter roten Linie herangezogen, da in diesem Bereich von einem eingeschwungenen Systemzustand ausgegangen werden kann und so eine Verzerrung des Spektrums sicher vermieden wird.

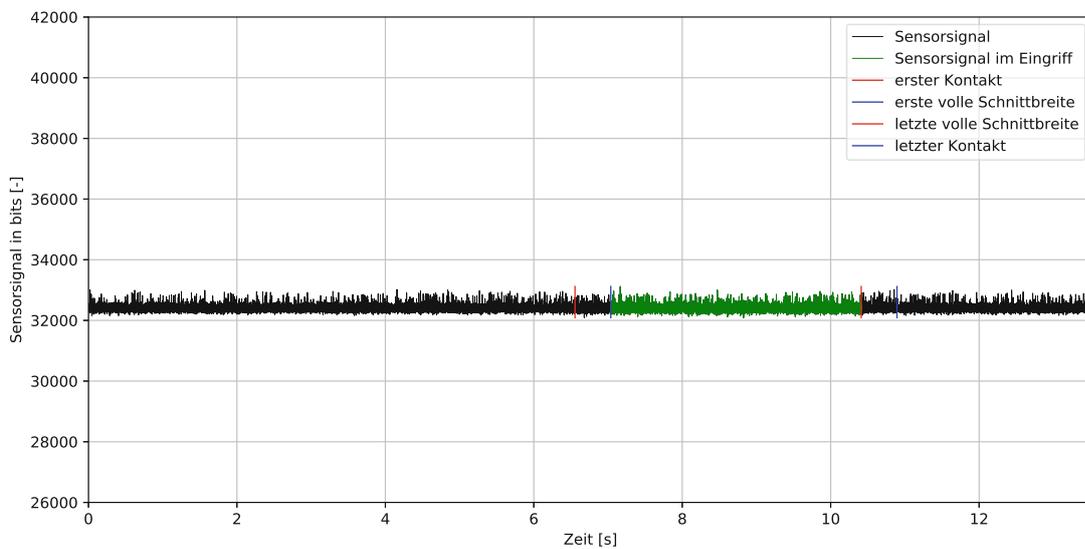
Im Zeitbereich gibt es gut erkennbare Unterschiede zwischen stabilen und instabilen Schnitten. So kommt es beim ersten Auftreffen der Schneiden auf das Werkstück immer zu erhöhten Beschleunigungswerten, egal ob stabiler oder instabiler Schnitt, jedoch klingen diese bei einem stabilen Schnitt wieder ab (siehe Abbildung 3.8). Beim instabilen hingegen, schwingen die Beschleunigungsamplituden auf und sind über den ganzen Schnitt hinweg deutlich höher (siehe Abbildung 3.9). Der Beginn der Eintrittsphase in das Werkstück und die gesamte Austrittsphase aus dem Werkstück sind bei stabilen und instabilen Schnitt sehr ähnlich.



(A) Signal des Versuchs 2010 (x-Richtung)

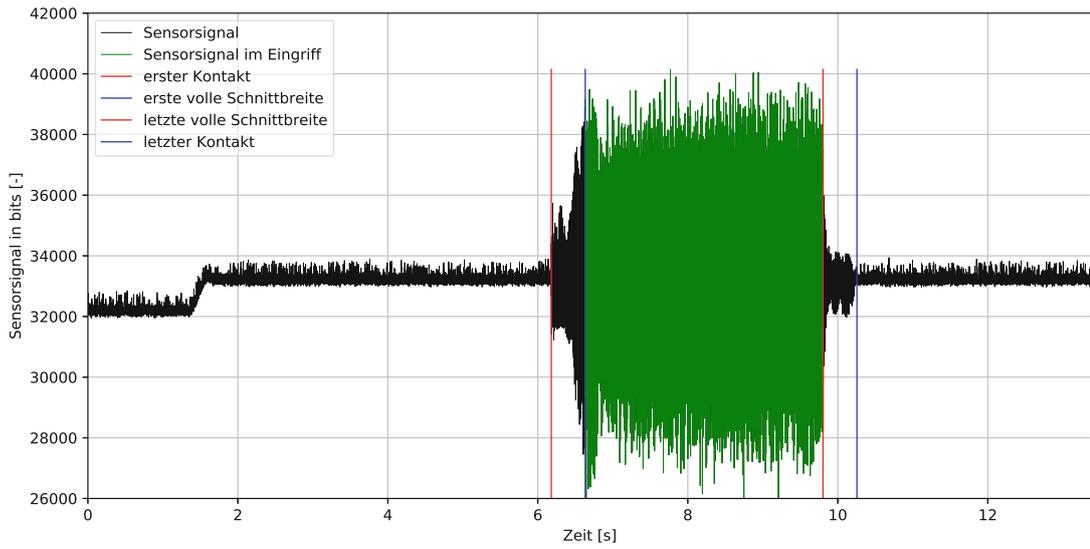


(B) Signal des Versuchs 2010 (y-Richtung)

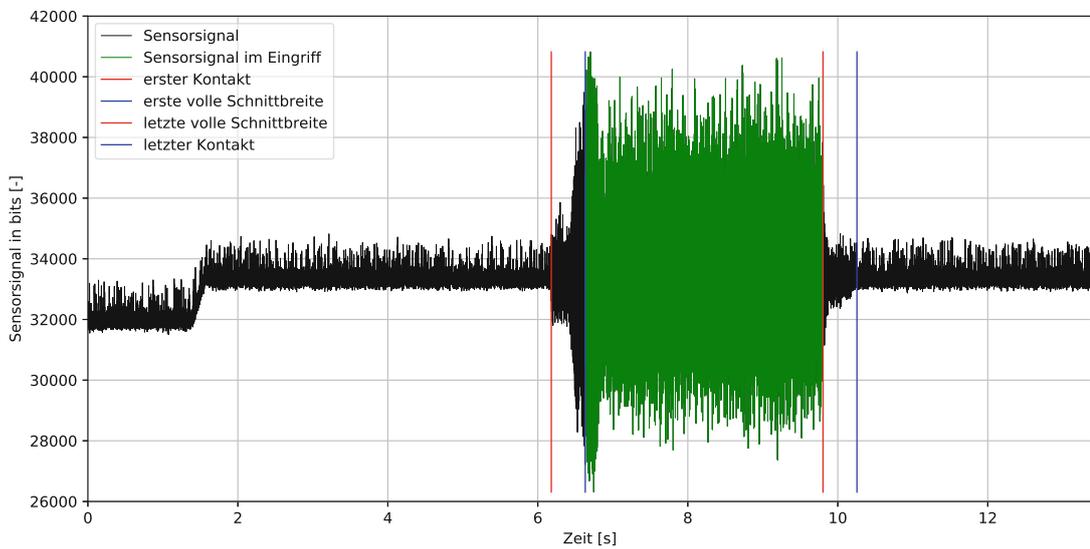


(C) Signal des Versuchs 2010 (z-Richtung)

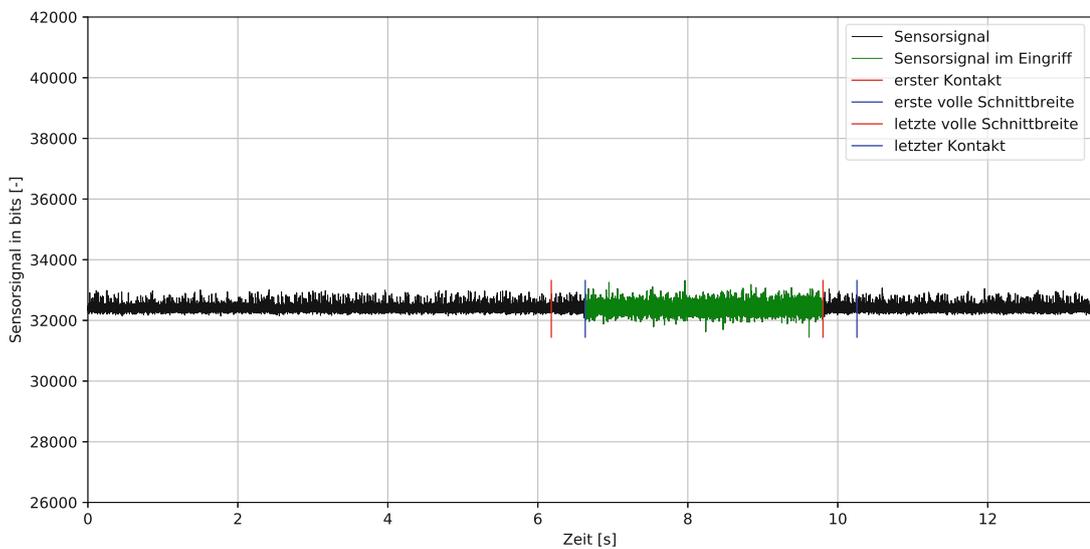
ABBILDUNG 3.8: Zeitreihendaten eines stabilen Prozesses



(A) Signal des Versuchs 2020 (x-Richtung)



(B) Signal des Versuchs 2020 (y-Richtung)



(C) Signal des Versuchs 2020 (z-Richtung)

ABBILDUNG 3.9: Zeitreihendaten eines instabilen Prozesses

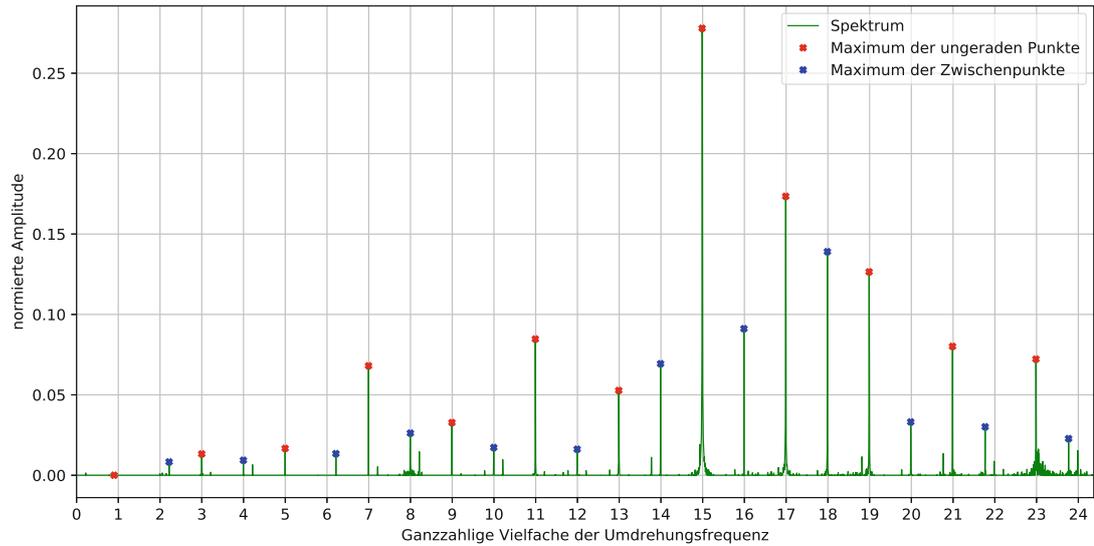
### 3.3.3 Gegenüberstellung von Signalen im Frequenzbereich

Mittels Division der Abszissenwerte der Frequenzspektren der Versuche 2010 und 2020 durch die Umdrehungsfrequenz, entstehen die in den Abbildungen 3.10 und 3.11 dargestellten Spektren. Grundsätzlich sollten laut Theorie bei einem Werkzeug mit  $N = 4$  Schneiden die Amplituden im Spektrum bei den  $i \cdot N \pm 1$  Vielfachen der Umdrehungsfrequenz liegen, sprich bei allen ungeraden Ordnungen der Umdrehungsfrequenz. Das ist dem Umstand geschuldet, dass es sich hier um keine stationäre Beschleunigungsmessung handelt, sondern um einen mitrotierenden Sensor. Dieser Modulationseffekt wird in [Ramsauer, 2019] beschrieben.

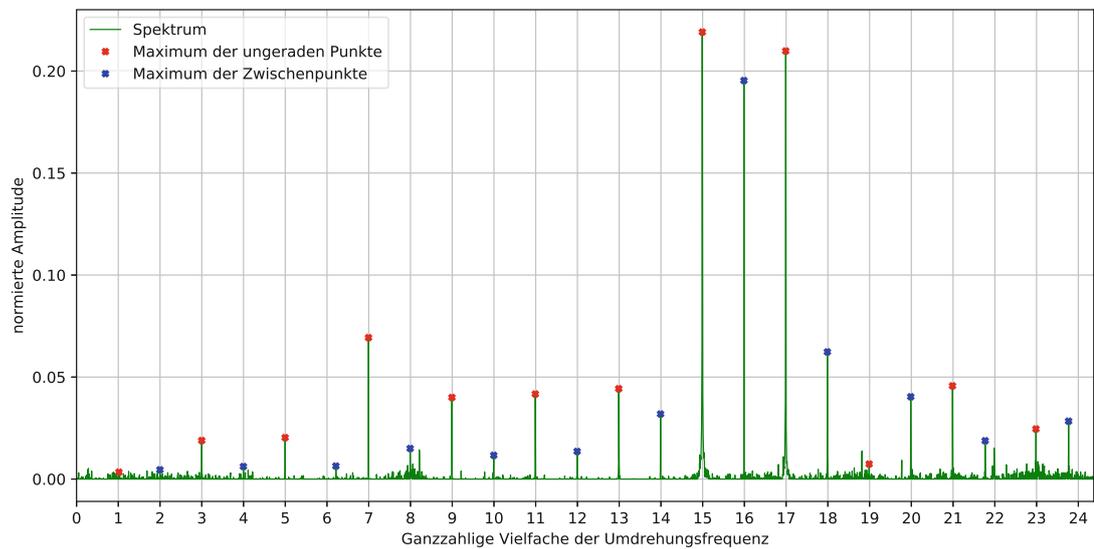
Die nicht vorhandene Aussagekraft der Messung in z-Richtung wird auch in den Spektren (Abbildungen 3.10c und 3.11c) sichtbar. Die Frequenzen treten wahllos mit relativ schwachen Amplituden auf. Dort ist kein signifikanter Unterschied zwischen stabilen und instabilen Schnitten erkennbar.

Bei einem stabilen Schnitt, wie es bei dem Versuch 2010 der Fall war, kann dieser Modulationseffekt in Abbildung 3.10 deutlich beobachtet werden. Es werden die Amplituden der ungeraden Vielfachen der Umdrehungsfrequenz in rot dargestellt. Die Spitzen die zwischen den ungeraden Vielfachen sind, sollten klein sein und zumindest so gut wie möglich auf geraden Vielfachen liegen. So liegen bei dem Versuch 2010 von den 12 höchsten Peaks ganze 11 auf den ungeraden Vielfachen (rote Punkte). Im instabilen Fall (Abbildung 3.11) ist das anders. Hier sind die roten Punkte schwach ausgeprägt und die blauen Punkte dominieren. Auch die blauen Punkte liegen deutlich abseits der geradzahligen Vielfachen der Umdrehungsfrequenz. Beim Versuch 2020 liegen die drei größten Peaks bei 13.8, 15.8 und 21.4.

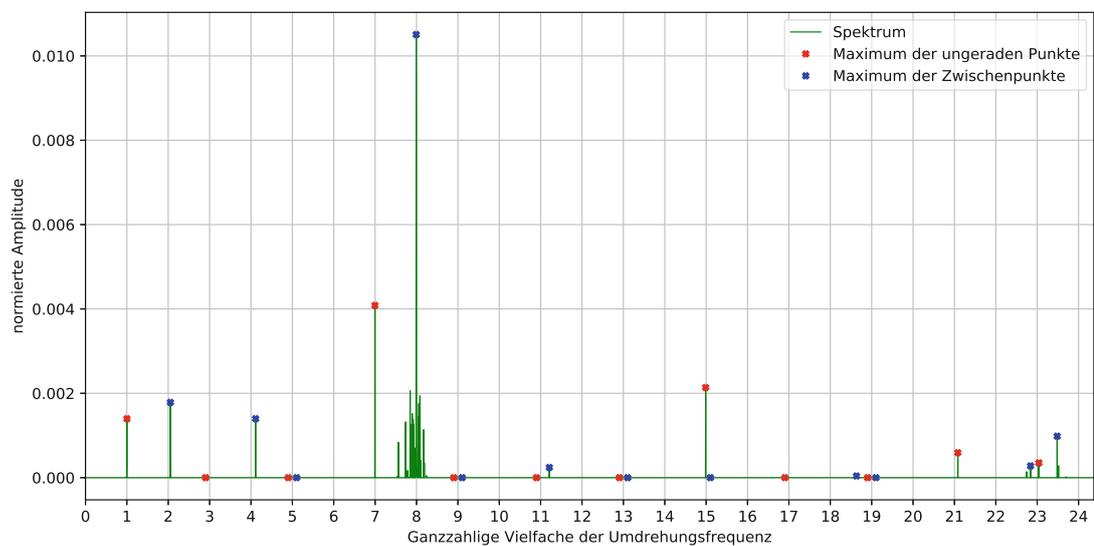
Allgemein gilt, dass bei instabilen Schnitten im modulierten Spektrum mit Ausschlägen zu rechnen ist, welche weder auf ungeraden noch auf geraden ganzzahligen Vielfachen der Umdrehungsfrequenz liegen, da Ratterschwingungen nahe der Eigenfrequenz der Maschine auftreten, die nicht notwendigerweise auf einer Vielfachen der Zahneingriffsfrequenz liegen muss, siehe Kapitel 2. Diese Überlegungen werden später Kapitel 4 in Features gebunden um mögliche Stabilitätsindikatoren zu finden. So werden zum Beispiel die i-größten roten Punkte mit den j-größten blauen Punkten verglichen oder der gesamte Signalgehalt mit allen roten Punkten, usw.



(A) Amplitudenspektrum des Versuchs 2010 (x-Richtung)

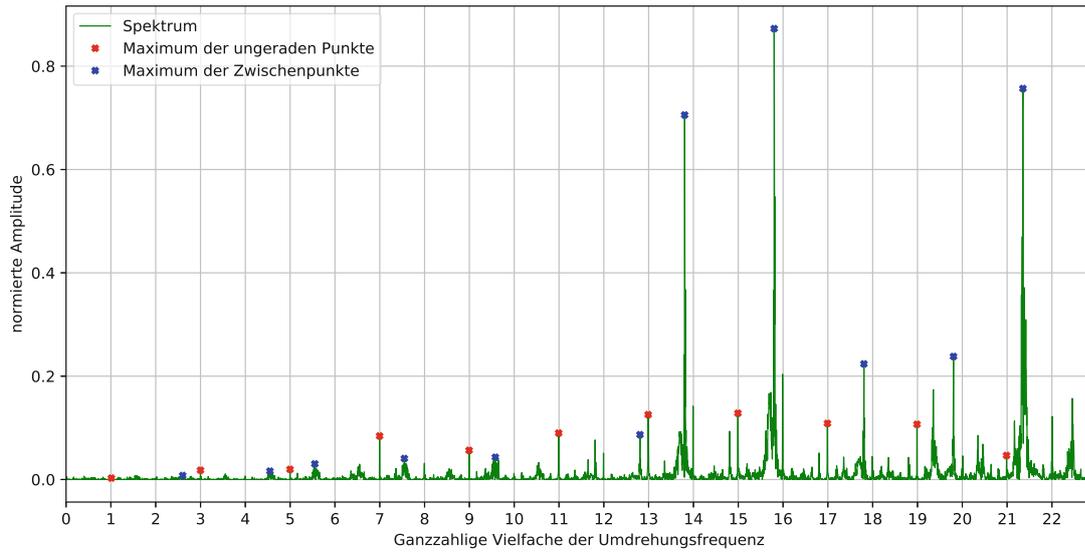


(B) Amplitudenspektrum des Versuchs 2010 (y-Richtung)

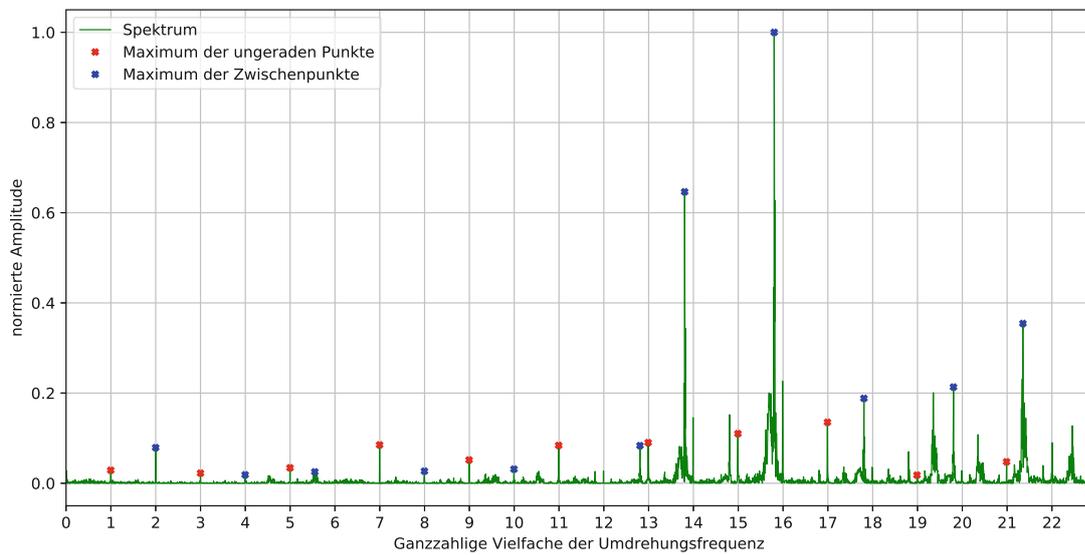


(C) Amplitudenspektrum des Versuchs 2010 (z-Richtung)

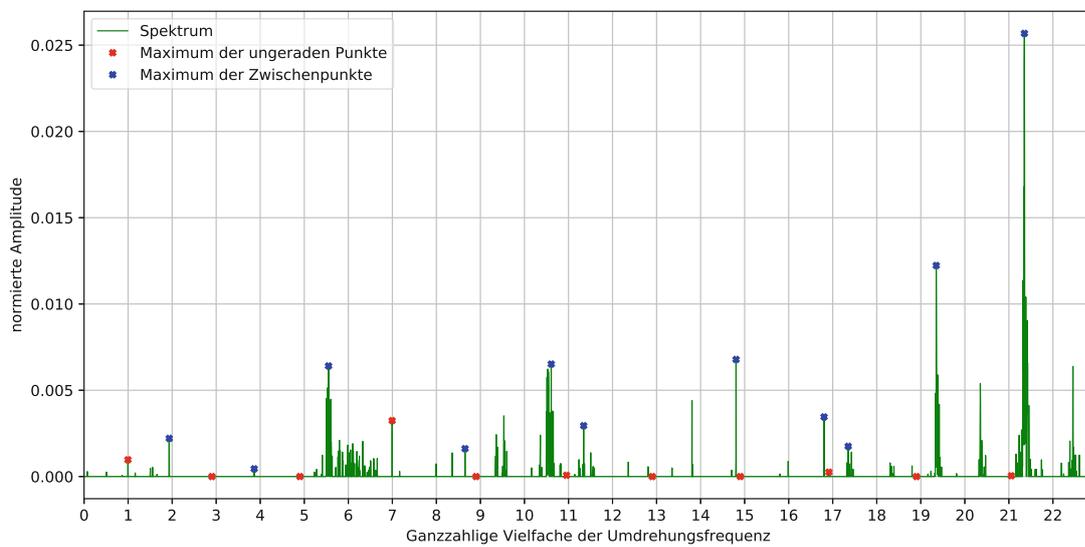
ABBILDUNG 3.10: Amplitudenspektren eines stabilen Prozesses



(A) Amplitudenspektrum des Versuchs 2020 (x-Richtung)



(B) Amplitudenspektrum des Versuchs 2020 (y-Richtung)



(C) Amplitudenspektrum des Versuchs 2020 (z-Richtung)

ABBILDUNG 3.11: Amplitudenspektren eines instabilen Prozesses

## Kapitel 4

# Erstellung und Auswahl von Stabilitätsindikatoren

Die in Kapitel 5 verwendete Form des Machine Learnings, das Supervised (Machine) Learning, benötigt Features, welche den Prozess anhand von Kennzahlen charakterisieren. Features ermöglichen in weiterer Folge Quantifizierung, Beurteilung und Klassifikation. Eine Ausnahme wäre Deep Learning. Hier entfällt der manuelle Feature Extraction Teil. Dazu mehr in Kapitel 5.

Für die Feature Extraction gibt es Standard-Features aus der Stochastik, welche ohne viel Aufwand analysiert werden können. Dieser Vorgang wird unter anderem auch in [Esmael u. a., 2013] beschrieben. Weiters werden konkret für diesen Anwendungsfall relevante Features aus dem Frequenzbereich geschaffen und analysiert. Alle Auswertungen beziehen sich dabei auf den Bereich der Schnittversuche, in denen das Schnittbreiten zu Fräserdurchmesser Verhältnis gleich 20mm/25mm ist. Beispielhaft ist dieser Bereich in Abbildung 3.8 und 3.9 grün gekennzeichnet. Auch die Frequenzspektren wurden mit den Daten aus dieser Zeitspanne gebildet. Das heißt, dass das Eindringen in das Werkstück beziehungsweise das Ausdringen aus dem Werkstück nicht berücksichtigt wird, da der eingeschwungene Zustand untersucht werden soll.

### 4.1 Feature Extraction im Zeitbereich

Die Feature Extraction wird in vielen Disziplinen der Signalverarbeitung ähnlich und nach dem gleichen Schema vollführt. Gerade bei Zeitreihendaten ist dies der Fall. Hier wird sich an statistischen Standardmaßen bedient, welche bei nahezu allen Zeitreihen-Analysen zur Anwendung kommen. Aus diesem Grund wird der Aufbau des Abschnittes 4.1 von [Esmael u. a., 2013] übernommen und sind in Abbildung 4.1 aufgelistet. All die, in diesem Abschnitt angeführten statistischen Features werden jedoch unter anderem auch in [Suhm, 2018] verwendet. Dieser Schritt ist für die Erstellung von Stabilitätsindikatoren und für die Erstellung von Features für das Machine Learning enorm wichtig. Die kursiv, in Klammern geschriebenen Namen, bezeichnen die Variablen in Python. Auf diese Benennung wird auch bei den später folgenden Feature Selection Methoden zurückgegriffen.

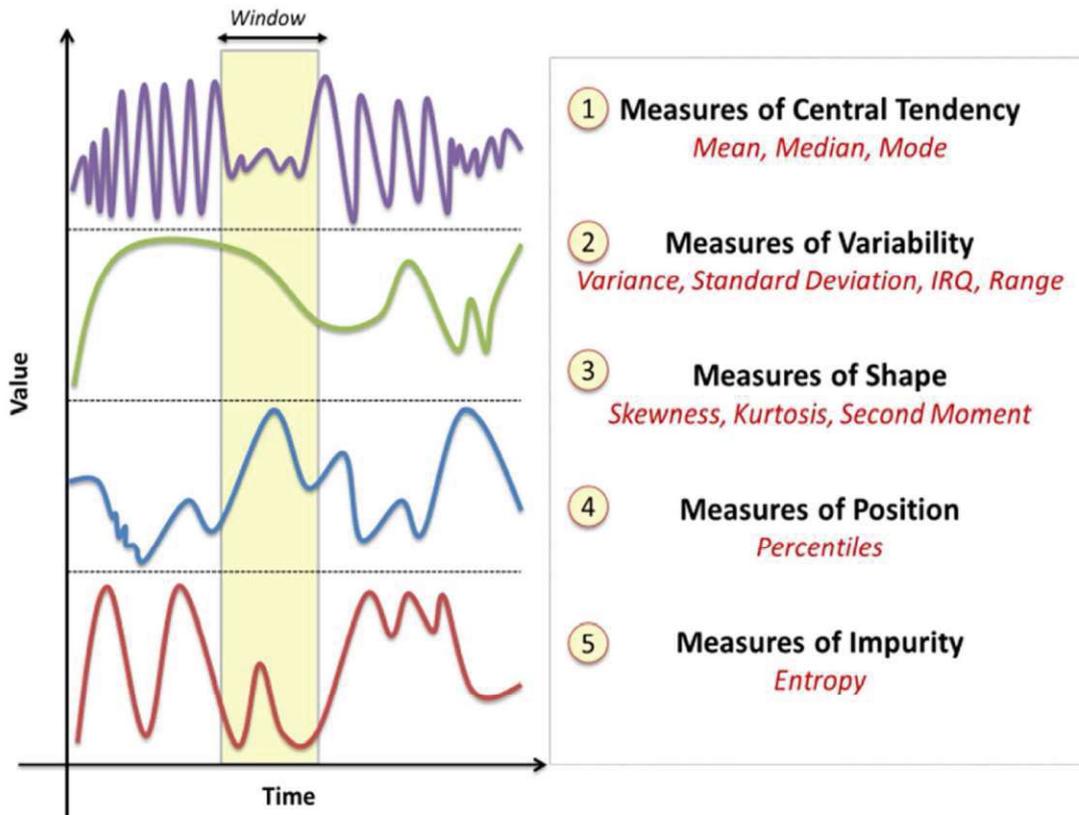


ABBILDUNG 4.1: Statistische Standard-Features [Esmael u. a., 2013]

#### 4.1.1 Zentrale Tendenz: Arithmetisches Mittel, Median, Quantilen

- Arithmetisches Mittel (*feature\_mean*):  
Der arithmetische Mittelwert ist das intuitivste und am einfachsten zu implementierende Feature.

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (4.1)$$

- Median (*feature\_median*):  
Der Median ist genau der Wert, welcher genau zwischen 50% der größten und 50% der kleinsten Werte liegt. Er wird oft statt des Mittelwertes herangezogen, da er unempfindlicher gegen Ausreißer in den Messdaten ist.
- Obere (*feature\_q75*) und untere (*feature\_q25*) Quartile:  
Die untere Quartile ist der Wert, welcher die 25% der kleinsten und 75% der größten Werte teilt. Die obere Quartile ist genau der gegenteilige Wert. Dieser teilt die Werte zwischen 75% der kleinsten und 25% der größten Werte.

#### 4.1.2 Streuungsmaße: Varianz, Interquartilsabstand

Die Streuungsmaße sagen nichts über das Potential des Signales aus, sondern über die zeitliche Änderung. Da die Varianz nur das Quadrat der Standardabweichung ist, wird die Standardabweichung nicht als Feature verwendet.

- Standardabweichung:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \quad (4.2)$$

- Varianz (*feature\_variance*):

$$\mu_2 = \sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \quad (4.3)$$

- Interquartilsabstand (*feature\_iqr*):

Der Interquartilsabstand ist die Differenz zwischen oberer und unterer Quartile, sprich der Wertebereich größer als 25% der kleinsten Werte und kleiner als 25% der größten Werte.

### 4.1.3 Gestaltmaße: Schiefe, Wölbung

Schiefe und Wölbung können als statistische Maße höherer Ordnung betrachtet werden.

- Schiefe (*feature\_skewness*):

$$\mu_3 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^3 \quad (4.4)$$

- Wölbung (*feature\_kurtosis*):

$$\mu_4 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^4 \quad (4.5)$$

### 4.1.4 Störmaße: Entropie

Der Begriff Entropie wird in der Statistik ähnlich gebraucht wie in der Thermodynamik. Damit kann auf Störstellen im Signal aufmerksam gemacht werden. Die Funktion P ist dabei eine Wahrscheinlichkeitsfunktion (probability mass function) [Esmael u. a., 2013].

- Signal-Entropie (*feature\_shannon*):

$$H = - \sum_{i=1}^n P(x_i) \cdot \log_2(P(x_i)) \quad (4.6)$$

## 4.2 Feature Extraction im Frequenzbereich

Wie in Abbildung 3.10 und 3.11 zu sehen ist, unterscheiden sich die Frequenzspektren stabiler und instabiler Klassifikation in vielen Schnitten deutlich. Wie schon zuvor angesprochen, sollten laut Modulationseffekt und einem Werkzeug mit vier Schneiden, die ungeraden Vielfachen der Umdrehungsfrequenz stark ausgeprägt sein. Kommt

es jedoch zu Instabilitäten im Prozess, verschieben sich diese Ausprägungen und liegen zwischen den ungeraden Vielfachen. In weiterer Folge werden die Punkte auf den ungeraden Vielfachen der Umdrehungsfrequenz, in den Spektren rot dargestellt, ungerade Punkte (oder ONP) genannt. Die Ausprägungen zwischen den ungeraden Vielfachen, in den Spektren blau dargestellt, werden Zwischenpunkte (oder ISP) genannt. Die erstellten Features sollen Zusammenhänge zwischen den ungeraden Punkten und den Zwischenpunkten darstellen. In einigen Features wird auch der gesamte Signalgehalt des Spektrum berücksichtigt.

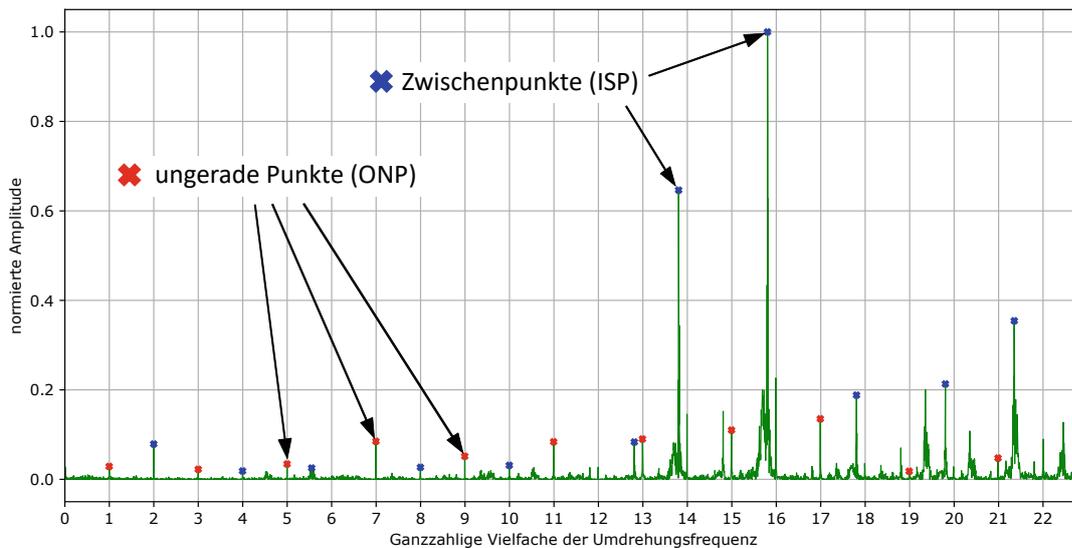


ABBILDUNG 4.2: Darstellung der ungeraden Punkte (ONP) und der Zwischenpunkte (ISP) im Amplitudenspektrum

Bei den unterschiedlichen frequenznormierten Spektren fällt auf, dass der Wertebereich der x-Achse variiert. Dies liegt daran, dass die Versuche mit unterschiedlichen Drehzahlen durchgeführt wurden. Der im Frequenzspektrum darstellbare Bereich wird einerseits durch das Nyquistkriterium auf die halbe Abtastrate beschränkt und andererseits durch die Normierung der Umdrehungsfrequenz noch einmal normiert. Um aussagekräftige Features, vor allem für höhere Drehzahlen, zu generieren, ist es daher wichtig, dass die Abtastrate dementsprechend hoch ist.

- Feature Nr. 0 (*feature\_0*):  
ist das Verhältnis aus der Summe aller Zwischenpunkte und der Summe aller ungeraden Punkte

$$f_0 = \frac{\sum_{i=1}^n ISP_i}{\sum_{i=1}^n ONP_i} \quad (4.7)$$

- Feature Nr. 1 (*feature\_1*):  
ist das Verhältnis vom größten Zwischenpunkt zum größten ungeraden Punkt

$$f_1 = \frac{ISP_{max}}{ONP_{max}} \quad (4.8)$$

- Feature Nr. 2 (*feature\_2*):  
ist das Verhältnis von der Summe der zwei größten Zwischenpunkte zur Summe der zwei größten ungeraden Punkte

$$f_2 = \frac{\sum_{i=1}^{n=2} ISP_{max_i}}{\sum_{i=1}^{n=2} ONP_{max_i}} \quad (4.9)$$

- Feature Nr. 3 (*feature\_3*):  
ist das Verhältnis von der Summe der drei größten Zwischenpunkte zur Summe der drei größten ungeraden Punkte

$$f_3 = \frac{\sum_{i=1}^{n=3} ISP_{max_i}}{\sum_{i=1}^{n=3} ONP_{max_i}} \quad (4.10)$$

- Feature Nr. 4 (*feature\_4*):  
ist das Verhältnis von der Summe der zwei größten Zwischenpunkte zum größten ungeraden Punkt

$$f_4 = \frac{\sum_{i=1}^{n=2} ISP_{max_i}}{ONP_{max}} \quad (4.11)$$

- Feature Nr. 5 (*feature\_5*):  
ist der das Verhältnis vom gesamten Signalgehalt des Spektrums zur Summe der ungeraden Punkte

$$f_5 = \frac{\sum_{i=1}^m y_{S_i}}{\sum_{i=1}^n ONP_i} \quad (4.12)$$

- Feature Nr. 6 (*feature\_6*):  
ist der das Verhältnis vom gesamten Signalgehalt des Spektrums, ohne den ungeraden Punkten, zur Summe der ungeraden Punkte

$$f_6 = \frac{\sum_{i=1}^m y_{S_i} - \sum_{i=1}^n ONP_i}{\sum_{i=1}^n ONP_i} \quad (4.13)$$

- Feature Nr. 7 (*feature\_7*):  
ist das Verhältnis des Mittelwertes der Zwischenpunkt zum Mittelwert der ungeraden Punkte

$$f_7 = \frac{\frac{1}{n} \sum_{i=1}^n ISP_i}{\frac{1}{n} \sum_{i=1}^n ONP_i} \quad (4.14)$$

- Feature Nr. 8 (*feature\_8*):  
ist das Verhältnis vom Mittelwert des gesamten Signalgehalts des Spektrums zum Mittelwert der ungeraden Punkte

$$f_8 = \frac{\frac{1}{m} \sum_{i=1}^m y_{S_i}}{\frac{1}{n} \sum_{i=1}^n ONP_{max_i}} \quad (4.15)$$

Um die Features zu bewerten und zu selektieren, werden für jedes Feature eigene Feature-Karten geplottet. Als Beispiel wird hier die Feature-Karte für Feature 1 gezeigt. Dabei werden die Wertebereiche der einzelnen Features auf den Wertebereich von 0 bis 1 normiert um, so untereinander verglichen zu werden.

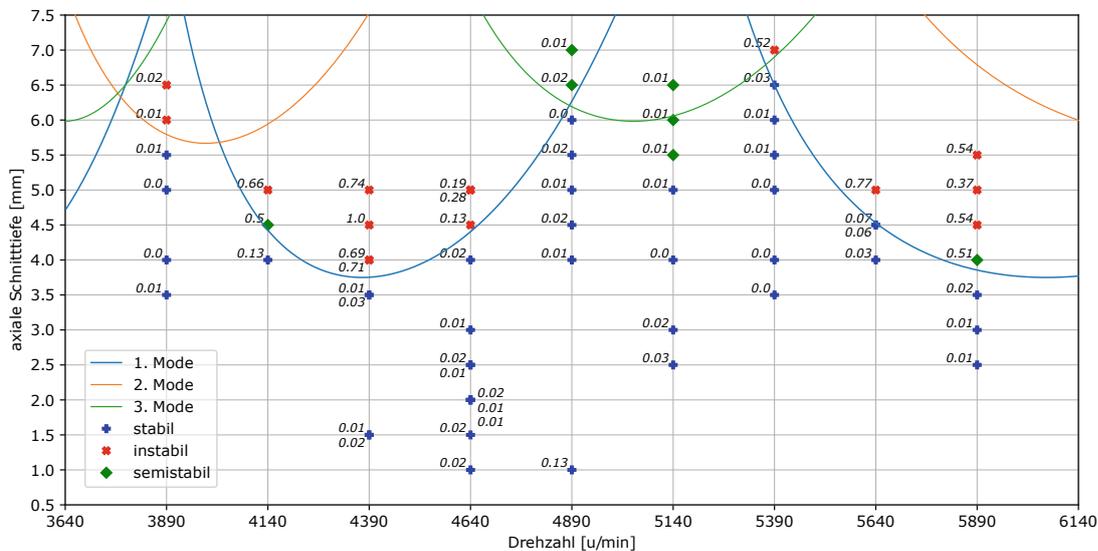


ABBILDUNG 4.3: Feature 1 für Versuchsreihe I mit Stabilitätskarte

Hier fällt auf, dass die beiden Punkte (3890/6.5) und (3890/6.0) nicht ins Schema passen. Auch beim Machine Learning in Kapitel 5 werden diese Punkte als stabil klassifiziert. Dies kann dem Umstand geschuldet sein, dass diese Punkte nicht wegen des ersten Eigenmodes, sondern wegen des zweiten Eigenmodes instabil sind und somit eventuell die gemessene Beschleunigung vom Sensorischen Werkzeughalters anders wahrgenommen wird.

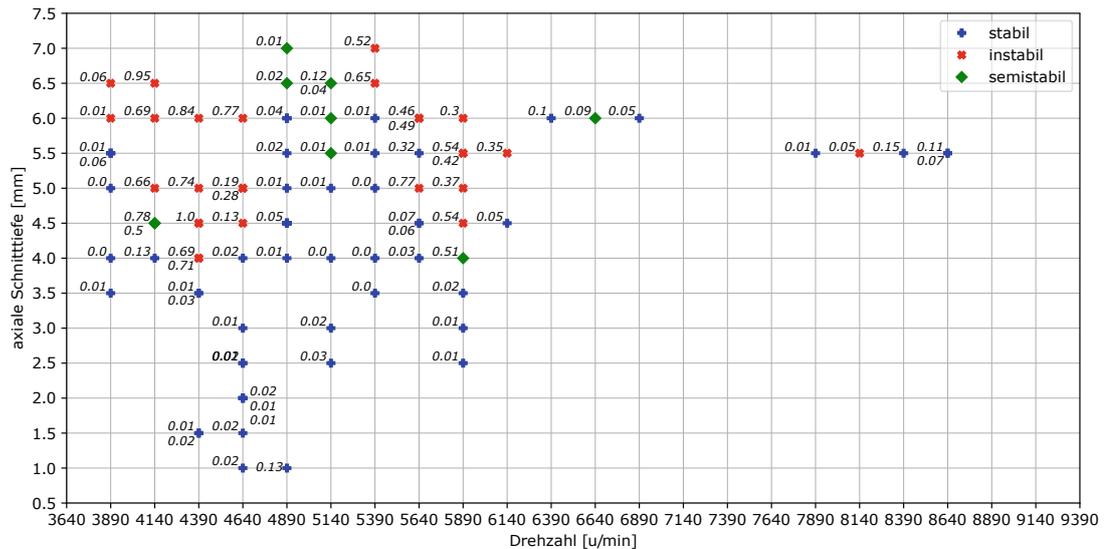


ABBILDUNG 4.4: Feature 1 für Versuchsreihe I & II

Es wird deutlich sichtbar, dass Feature 1 eine gute Aussagekraft über den Übergang zwischen stabilen und instabilen Bereich hat, da die Feature-Werte stabiler Schnitte eher nahe 0 sind und die instabiler Schnitte tendenziell nahe 1 sind.

### 4.3 Feature Selection

Nachdem eine Vielzahl von Features sowohl aus den Zeitbereich als auch dem Frequenzbereich erstellt wurden, müssen nun diese auf eine überschaubare Anzahl reduziert werden (dimensionality reduction). Eine Aufzählung von möglichen Reduktionsmethoden ist in [Sharma, 2018] oder in [Pedregosa u. a., 2011] zu finden. Daraus wurden einige Möglichkeiten herangezogen und auf das hier stehende Problem angewandt. Es werden auch andere Methoden, wie zum Beispiel die Neighborhood Components Analysis, aus [Suhm, 2018], angewandt.

- Fehlende Feature-Werte:

Ein mögliches Auswahlkriterium für Features ist jene auszusortieren, welche die meisten fehlerhaften beziehungsweise fehlenden Datensätze aufweisen. Im vorliegenden Fall werden die Features aus eigens durchgeführten Versuchen berechnet und dementsprechend treten keine Lücken in den Messdaten auf. Aus diesem Grund wird dieses Verfahren nicht angewandt.

- Geringe Varianz-Kriterium:

Bei diesem Verfahren werden die Features mit der größten Varianz innerhalb des Features selbst ausgewählt. In Python ist die Funktion *VarianceThreshold* aus dem Modul *scikit-learn* [Pedregosa u. a., 2011] abrufbar.

1.	feature_1
2.	feature_2
3.	feature_variance
4.	feature_iqr
5.	feature_shannon
6.	feature_3
7.	feature_7
8.	feature_0

TABELLE 4.1: Reihung anhand der Varianz der Features

- Hohe Korrelations-Kriterium:  
Eine weitere Feature Selection Methode ist eine Korrelationsanalyse mittels  $\chi^2$ -Test. Auch diese Funktion, namens *SelectKBest(chi2)* stammt aus *scikit-learn* [Pedregosa u. a., 2011].

1.	feature_1
2.	feature_variance
3.	feature_2
4.	feature_3
5.	feature_4
6.	feature_7
7.	feature_0
8.	feature_iqr

TABELLE 4.2: Reihung anhand eines Korrelations-Kriterium

- Random Forest Verfahren:  
Dieses Verfahren, gibt den einzelnen Features Gewichtungen. Die zuvor gezeigten Methoden hingegen, geben nur die n-besten Features an, weshalb nur eine Reihung ohne genauere Aussage möglich ist. Das Random Forest Verfahren (oder auch randomized trees) wird in [Geurts, Ernst und Wehenkel, 2006] erläutert und ausgeführt nach welchen Gesichtspunkten die Gewichtung erfolgt. Mit Hilfe dieser Publikation wurde die Funktion *ExtraTreesClassifier* in [Pedregosa u. a., 2011] implementiert und ist mit Python ebenfalls ausführbar.
- False-Positiv-Kriterium:  
Hier wird anhand der False-Positive-Rate (kurz: FPR) und einem gewissen Schwellwert entschieden, welches Features zu den aussagekräftigsten gehören. Auch diese Funktion ist im Python Modul *scikit-learn* [Pedregosa u. a., 2011] zu finden.

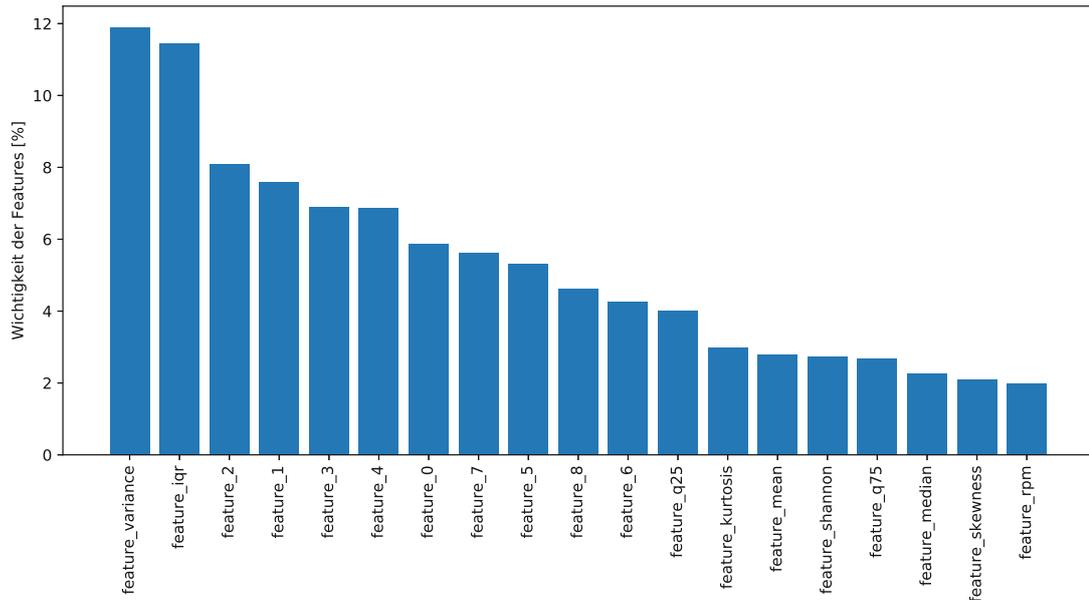


ABBILDUNG 4.5: Feature Selection mit dem Random Forest Verfahren

1.	feature_1
2.	feature_variance
3.	feature_2
4.	feature_3
5.	feature_4
6.	feature_7
7.	feature_0
8.	feature_iqr

TABELLE 4.3: Reihung anhand des FPR-Kriteriums

- Neighborhood Components Analysis (NCA):  
Die Neighborhood Components Analysis ist im Vergleich zu anderen Feature Selection Methoden ein relativ junges Verfahren. Der Grundgedanke dieses Verfahrens beruht auf dem k-nearest-neighbor (kNN) Klassifikationsverfahren und wird unter anderem in [Yang, Wang und Zuo, 2012] erklärt. Für die hier vorhandene Aufgabenstellung wird die MATLAB Funktion *fscnca()* verwendet.

1.	feature_variance
2.	feature_shannon

TABELLE 4.4: Reihung durch die NCA

Zusammenfassend liefern viele Verfahren also die Aussage, dass sich *feature\_variance* und *feature\_1* als sehr aussagekräftig erweisen. Wenig aussagekräftig sind hingegen beispielsweise *feature\_median*, *feature\_kurtosis*, *feature\_skewness* oder *feature\_6*.



## Kapitel 5

# Machine Learning

In diesem Kapitel wird zu Beginn ein Einblick in das Thema künstliche Intelligenz gegeben und in weiterer Folge wird gezeigt, wie Machine Learning umgesetzt werden kann. Machine Learning ist in der technischen Informatik ein weit verbreiteter Begriff, jedoch im Maschinenbau noch etwas unbekannt. Mit dieser Einführung soll es dem Leser beziehungsweise der Leserin leichter fallen, die weiteren Teile dieses Kapitels zu verstehen.

### 5.1 Einführung zu künstlicher Intelligenz

Ein System mit künstlicher Intelligenz kann die Umgebung in gewisser Form wahrnehmen, auf Einflüsse reagieren und das Systemverhalten anpassen. All diese Aufgabenbereiche hat auch jedes klassische Regelungssystem. Der Unterschied jedoch ist, dass ein System mit künstlicher Intelligenz nicht denen von Menschen implementierten Regelungsstrategien folgt, sondern anhand von Daten trainiert wird und eigene Strategien entwickelt. Künstliche Intelligenz, in weiterer Folge kurz KI genannt, wird aufgrund von immer größer werdenden Speicherkapazitäten und kürzeren Rechenzeiten stets populärer.

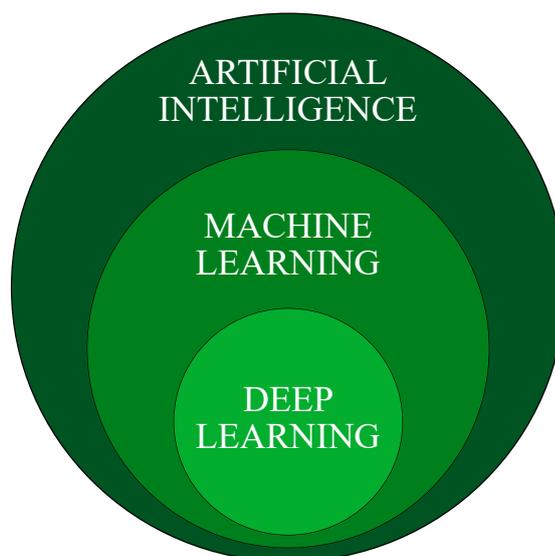


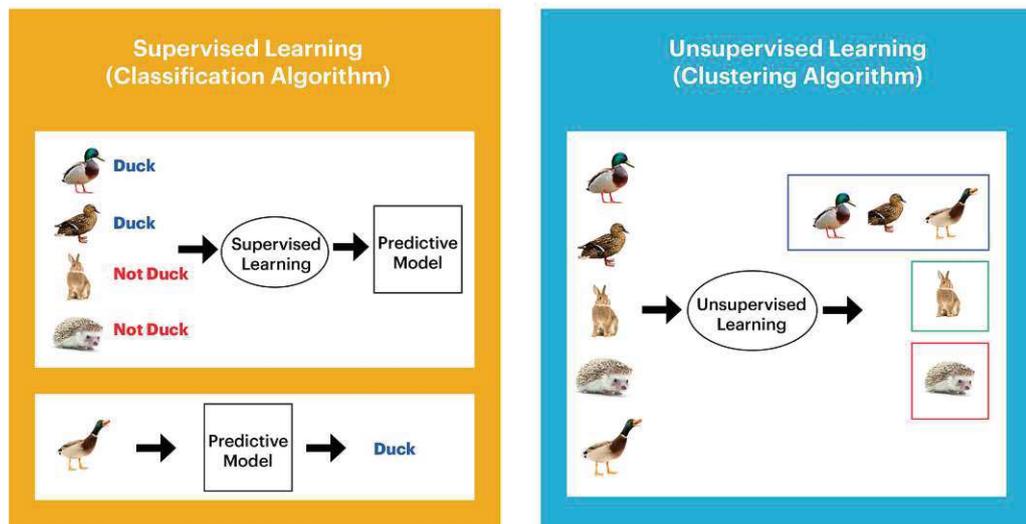
ABBILDUNG 5.1: Ebenen von KI [MathWorks, 2018]

### 5.1.1 Machine Learning

Der Grundgedanke beim Machine Learning ist das mehr oder weniger selbstständige Lernen des Algorithmus mit der Möglichkeit, sollten mehr beziehungsweise aussagekräftigere Daten vorhanden sein, durch erneutes Training des Algorithmus die Performance zu verbessern. Es haben sich eine Vielzahl an Methoden für unterschiedlichste Anwendungen etabliert. Die folgende Aufzählung ist lediglich ein Auszug aus möglichen Verfahren und Methoden und wurde aus [Fumo, 2017, MathWorks, 2016] zusammengefasst.

- Unsupervised Learning
  - Clustering
    - \* k-means
    - \* k-medoids
    - \* Fuzzy C-Means
    - \* Hierarchical
    - \* Gaussian Mixture
    - \* Hidden Markov Model
- Supervised Learning
  - Classification
    - \* Support Vector Machines
    - \* Discriminant Analysis
    - \* Native Bayes
    - \* k-Nearest Neighbor
    - \* Neural Networks
  - Regression
    - \* Linear Regression
    - \* Generalized Linear Model
    - \* Support Vector Regression
    - \* Gaussian Process Regression
    - \* Ensemble Methods
    - \* Decision Trees
    - \* Neural Networks
- Semi-Supervised Learning
- Reinforcement Learning

Der Unterschied zwischen den beiden Hauptkategorien „Unsupervised Learning“ und „Supervised Learning“ ist jener, dass beim Supervised Learning gekennzeichneten Datenreihen beziehungsweise vorklassifizierte Daten vorhanden sein müssen. Diese Klassifikation kann binär sein, wie zum Beispiel (True/False), (1/0), (gut/schlecht) oder auch aus mehreren Klassen bestehen wie zum Beispiel (gut/neutral/schlecht), (Hund/Ente/Katze/Igel). Dies ist eine Notwendigkeit, da das fertige Modell Voraussagen treffen soll und als Output genau eine dieser Klassen ausgegeben werden soll. Beim Unsupervised Learning werden die Datensätze lediglich in eine endliche Anzahl an Gruppen unterteilt. Die einzelnen Gruppen teilen ähnliche Eigenschaften, welche aus den Features hervorgehen. Diese Methode wird unter anderem zur Mustererkennung verwendet. Hier gibt es jedoch kein Modell mit dem Voraussagen getroffen werden können.



Western Digital.

ABBILDUNG 5.2: Unterschied von Supervised Learning und Unsupervised Learning [Zhou, 2018]

### 5.1.2 Supervised Machine Learning

In Abbildung 5.2, auf der linken Seite, wird die Grundidee des Supervised Learnings dargestellt. Als Trainingsdaten werden vier Bilder von Tieren herangezogen - zwei Enten, ein Hase und ein Igel. Diese werden jedoch binär klassifiziert mit den Klassen *Ente* oder *keine Ente*. Mit diesen Eingangsgrößen wird ein Modell erstellt, welches in späterer Folge ein weiteres Bild einer Ente, welches nicht Bestandteil der Trainingsdaten ist, erfolgreich als Ente klassifiziert. Ein genauerer Ablauf des Supervised Learnings wird in Abbildung 5.3 dargestellt. Ausgangsbasis hierfür sind die Zeitreihendaten welche

in diesem Fall die Beschleunigungsdaten des Sensorischen Werkzeughalters sind. Diese müssen auf Vollständigkeit geprüft werden und in die gewünschten Klassen (stabil, instabil) unterteilt werden. Weiters werden aus dem Beschleunigungssignal Features extrahiert. Dieser Vorgang wird in dieser Arbeit in Kapitel 4 gezeigt. Danach werden diese Daten strukturiert in sogenannte Feature Tables abgelegt. Für den weiteren Verlauf werden die Daten in Trainings- und Testdaten gesplittet. All diese Daten sind einem bestimmten Versuch und einer bestimmten Klasse zugeordnet. Im nächsten Schritt wird der Machine Learning Algorithmus trainiert. Ergebnis des Trainings ist ein Modell, mit dem Vorhersagungen getroffen werden können. Um die Qualität des Modells zu quantifizieren, werden nun die Testdaten herangezogen und das Modell trifft die Vorhersagungen. Um nun die Fehlerrate des Modells zu erhalten werden die vorhergesagten Klassen der Testdaten mit den realen Klassen verglichen und das Ergebnis wird in Confusion Matrices dargestellt.

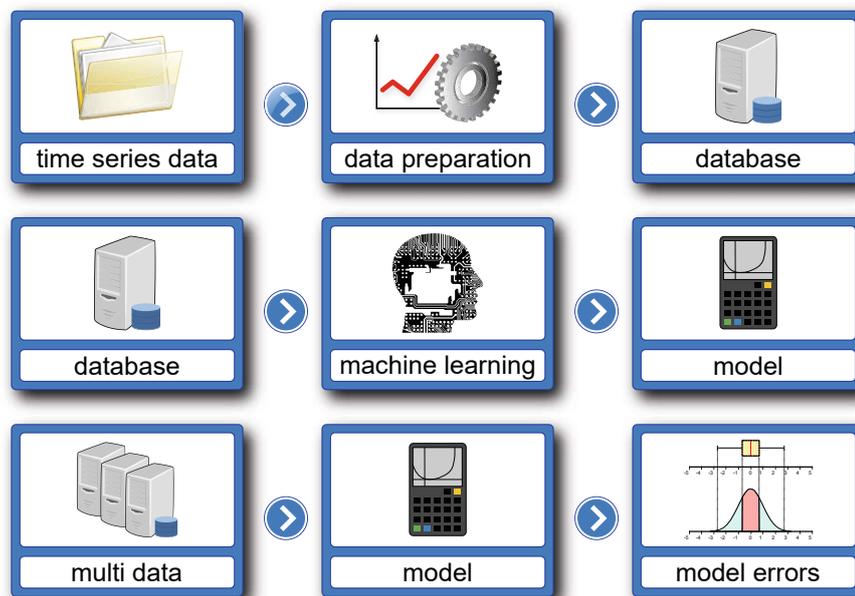


ABBILDUNG 5.3: Ablauf des Supervised Learnings [Bleicher, 2019a]

### 5.1.3 Deep Learning

Deep Learning beruht auf den Prinzipien von neuronalen Netzen und funktioniert in der Regel nur wenn eine sehr große Datenbasis an Trainingsdaten vorhanden ist. Der Grundgedanke ist, dass Feature Extraction und Feature Selection entfallen und das Modell als Input nur die Rohdaten erhält. Die beiden erwähnten Schritte werden durch eigene Layer im neuronalen Netzwerk ersetzt. In [Dettmers, 2015] gibt der Grafikprozessor-Hersteller NVIDIA eine gute Einführung zum Thema Deep Learning. In Abbildung 5.4 wird ein neuronales Netz für Deep Learning beispielhaft dargestellt. Es sind nicht zwingenderweise Zeitreihen-Daten als Input erforderlich, denn gerade neuronale Netze funktionieren hervorragend bei Pixelgrafiken. In diesem Beispiel geht es um die Erkennung von Tempobeschränkungen. Als Input wird eine Pixelgrafik von

einer 60mph Tafel gegeben und direkt an den Feature-Extraction Layer und unter Umständen danach auch an einen Feature-Selection Layer, übergeben. Nachgeschaltet an diese(n) Layer, ist das klassische neuronale Netzwerk, welches auch für die Klassifikation benutzt wird. Der Output bei diesem Beispiel ist zum einen der Begriff *sign* und die Zahl 60.

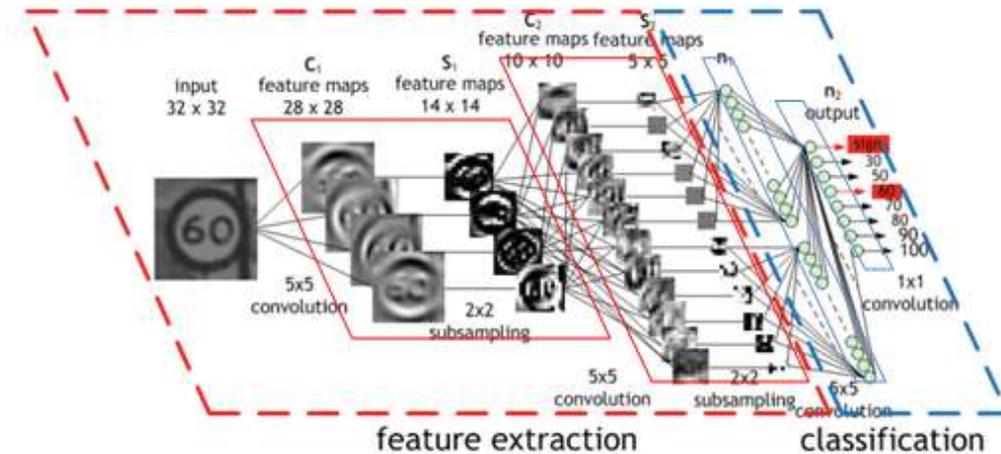


ABBILDUNG 5.4: Aufbau eines neuronalen Netzes für DL [Dettmers, 2015]

In dieser Arbeit ist die Verwendung von Deep Learning Methoden aufgrund der beschränkt kleinen Datenmengen nicht zielführend. In späteren Arbeiten mit dem Sensorischen Werkzeughalter kann dieses Thema sehrwohl von Relevanz sein.

## 5.2 Anwendung von ML: Herzton-Klassifizierung

Mit dem MATLAB-Skript *HeartSoundClassifier* [Suhm, 2018] versucht MathWorks die in MATLAB implementierten Machine Learning Methoden an einem Beispiel zu demonstrieren. Grundsätzlich sollen Herzöne als .wav-Dateien eingelesen werden, welche als normal und anormal klassifiziert wurden, und mithilfe von Machine Learning soll ein Algorithmus so trainiert werden, dass dieser in Zukunft selbst diese Unterscheidung treffen kann. Bei der Feature Extraction werden die in Abschnitt 4.1 gezeigten Features verwendet. Ebenfalls werden die in der Spracherkennung verbreiteten Mel Frequency Cepstral Coefficients (MFCC) als Features verwendet. Bei der Feature Selection wird dann ausschließlich auf die Neighborhood Component Analysis zurückgegriffen. Da diese Methode in der Bibliothek *scikit-learn* nicht in der Form implementiert ist, wird diese als einzige Feature Selection Methode aus MATLAB übernommen. Mit Hilfe der MATLAB Applikation *ClassificationLearner* können folgende Schritte des Machine Learnings umgesetzt werden (Abbildung 5.5):

- Splitten von Trainings- und Testdaten

Das Teilen des gesamten Datensatzes ist notwendig, da das Modell einerseits trainiert und andererseits auch validiert werden muss. Geschieht dies nicht beziehungsweise nur unzureichend, so besteht die Gefahr des Overfittings. In MATLAB ist das Verhältnis von Test- zu Gesamtdaten zwischen 5% und 50% frei wählbar. Als maximaler Grenzwert wird häufig 50% gewählt, da die Trainingsdaten ausreichend genug vorhanden sein müssen um das Modell mit akzeptabler Genauigkeit zu trainieren. Aus diesem Grund wird auch beim selbst erstellten Machine Learning Algorithmus auf diese Grenze geachtet.

- Features paarweise in einem Koordinatensystem vergleichen  
Um lineare Abhängigkeiten oder andere Effekte der einzelnen Features zu erkennen, können diese in der Applikation *ClassificationLearner* mit der Funktion *Scatter Plot* gegenseitig verglichen werden (siehe Abbildung 5.5). Im besten Fall heben sich beiden Klassen deutlich voneinander ab.
- Testen von Machine Learning Verfahren auf deren Effizienz und Genauigkeit  
Es können eine Vielzahl von Klassifizierungsmethoden (kNN, Decision Trees, Support-Vector-Machines, etc.) getestet und verglichen werden. In Abbildung 5.5 werden unterschiedliche Arten von Support-Vector-Machines (kurz: SVM) verglichen und als klarer Favorit wird die kubische Support-Vektor-Machine ausgewählt.
- Erzeugen von Confusion Matrices  
Anhand der trainierten Modelle können unterschiedliche Confusion Matrices erzeugt werden. Diese können auch normiert anhand der True-Positive-/False-Negativ-Rate oder der Positive-Predictive/False-Discovery-Rate dargestellt werden.

Der große Vorteil bei der in diesem Abschnitt gezeigten Anwendung von Machine Learning im Vergleich zur Rattererkennung in dieser Arbeit, ist der, dass dank der Open-Source Datenbank PhysioNet (<https://physionet.org/>) eine große Anzahl an Test- und Trainingsdaten für die Herzton-Klassifizierung zur Verfügung stehen. Dadurch wird die Genauigkeit des Algorithmus stark verbessert.

### 5.3 Erstellung des neuronalen Netzes

Das neuronale Netzwerk wird mit der Python Bibliothek *scikit-learn* [Pedregosa u. a., 2011] erstellt. Die Grundfunktion nennt sich *MLPClassifier()*. MLP ist eine Abkürzung für multilayer perceptron und beschreibt den allgemeinen Aufbau eines neuronalen Netzes, welches anhand der zwei Grundfunktionen *feedforward* und *backpropagation* trainiert wird. [Loy, 2018]. Beim *feedforward* wird mit vorausgewählten Gewichtungen ein Ergebnis berechnet und der Fehler zum eigentlichen Ergebnis ausgewertet. Danach werden bei der *backpropagation* die neuen Gewichtungen so gewählt dass der Fehler ein Minimum wird. Dieser Vorgang wird so lange wiederholt, bis der Fehler

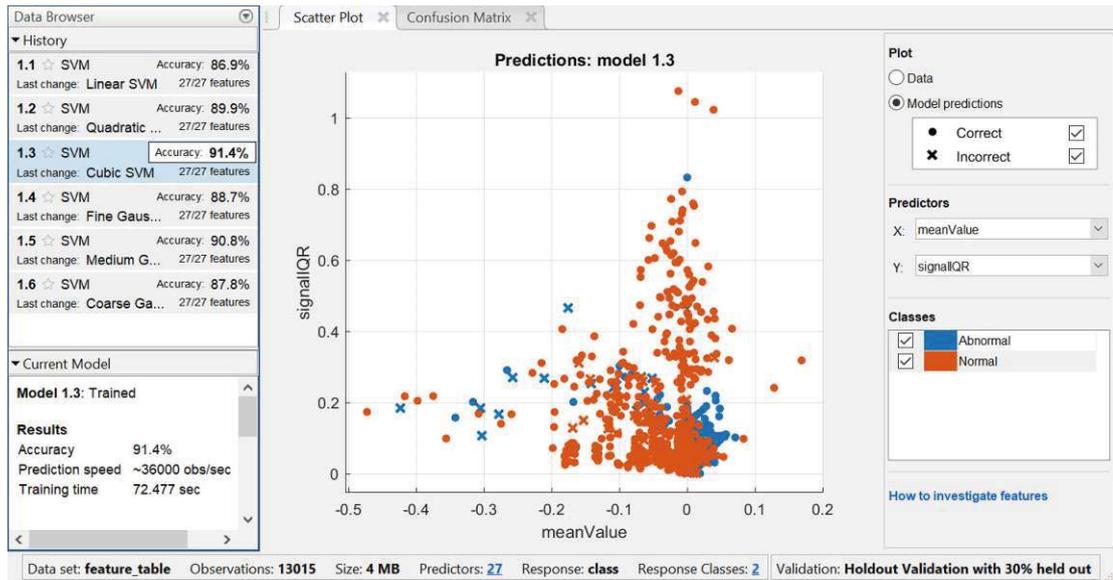


ABBILDUNG 5.5: MATLAB App ClassificationLearner mit Daten aus [Suhm, 2018]

unter einem gewissen Grenzwert ist. Das Ziel des Trainings ist die Gewichtungen der einzelnen Neuronen so zu wählen, sodass unterschiedliche Features optimal zur richtigen Klassifizierung führen.

Als erstes muss der Feature-Table, die Tabelle die alle Features und die dazugehörige Klassifikation (stabil/instabil) enthält, erstellt werden. Hier sind alle Parameter, welche für das Machine Learning erforderlich sind enthalten und daher werden die Zeitreihendaten nicht mehr benötigt. Das hier entwickelte neuronale Netzwerk, wie in Abbildung 5.6 gezeigt, besteht aus einem Input-Layer, zwei Zwischenlayer (auch: Hidden-Layer) und einem Output-Layer. Der Input-Layer beinhaltet 5 Features. Die Anzahl der Features ist frei wählbar, jedoch sollte bedacht werden, dass die Rechenzeit mit jedem Feature anwächst, die Genauigkeit des Modells nicht zwangsläufig steigen muss. Die beiden Zwischen-Layer bestehen jeweils aus 100 Neuronen. Die Anzahl der Neuronen werden durch den Default-Wert, welcher von [Pedregosa u. a., 2011] in der Funktion `MLPClassifier()` vorgeschlagen wird definiert. Die Anzahl der Zwischenlayer wird deshalb auf zwei Stück beschränkt, da ein weiterer Zwischenlayer die Genauigkeit in diesem Fall nicht mehr verbessert, es jedoch zu einem Anstieg der Rechenzeit kommt. Der Output-Layer besteht aus den beiden Klassen, welche als stabil oder instabil deklariert sind.

### 5.3.1 Training und Validierung

Der erste Schritt des Machine Learnings, welcher unabhängig von der gewählten Methode ist, ist das Normieren der Features. Die für das Normieren verwendete Funktion in Python der Bibliothek `scikit-learn` lautet `StandardScaler()`. Diese Funktion normiert

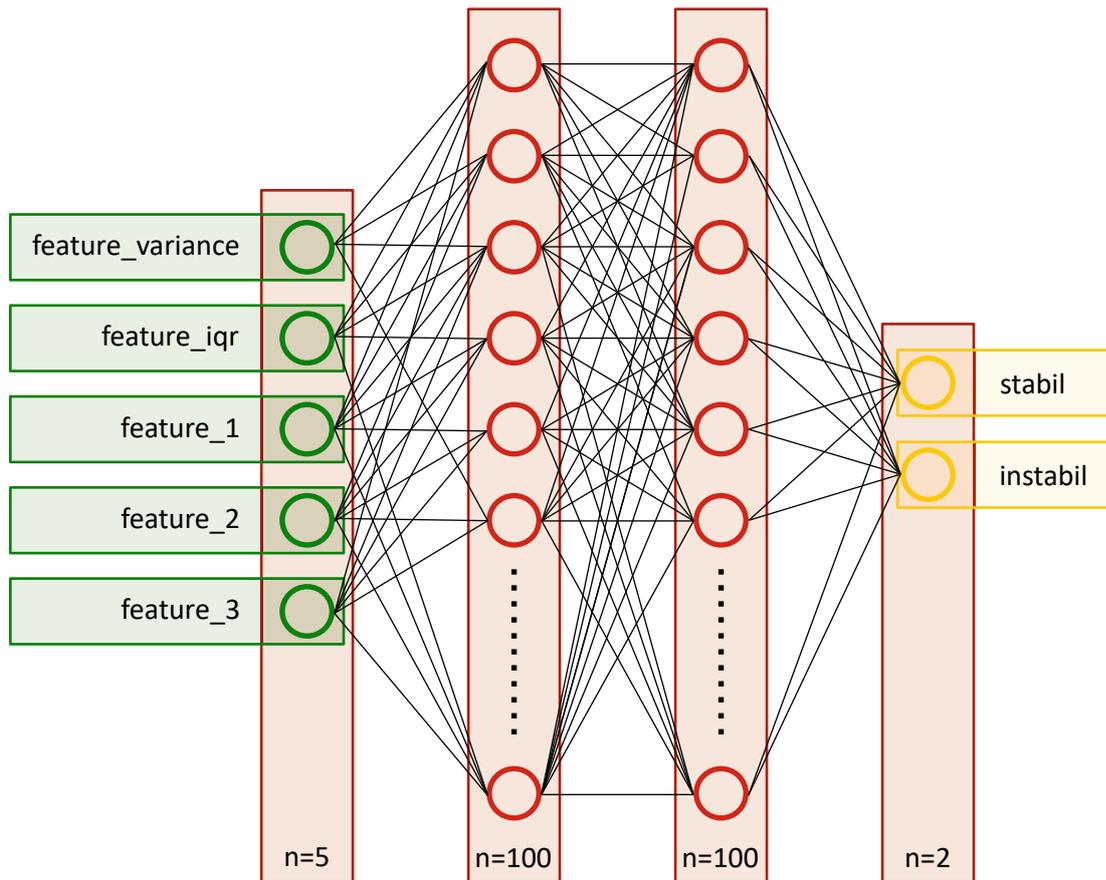


ABBILDUNG 5.6: Aufbau des neuronalen Netzes

die Features so, dass der Mittelwert jedes Features Null wird und die Standardabweichung gleich 1. Es gibt auch andere Normierungsarten wie zum Beispiel *MinMaxScaler()*, welcher die Features zwischen zwei Zahlenwerte (meist zwischen 0 und 1) legt. Das Splitten von Trainings- und Testdaten ist der nächste Schritt in der Reihenfolge und wird mit der Funktion *train\_test\_split()* vollzogen. Das Verhältnis von Trainings- und Testdaten wird 1:1 gewählt. Um gewisse fehlerverursachende Effekte zu vermeiden, werden nicht die ersten 50% der Gesamtdaten zum Trainieren und die letzten 50% zum Testen herangezogen, sondern es wird randomisiert gesplittet. Um eine wissenschaftliche Nachvollziehbarkeit zu gewährleisten, ist jede Random-Funktion mit einer Zählnummer hinterlegt und kann jederzeit genauso aufgerufen werden. In diesem Fall handelt es sich um *random\_state = 42*. Nach dem Trainieren des neuronalen Netzwerks werden die Testdaten zum Validieren herangezogen. Die Performance hierbei kann in Confusion Matrices (Abbildung 5.7) dargestellt werden. In dem hier vorliegenden Fall werden drei von 50 Testreihen falsch klassifiziert. Es handelt sich dabei um die in Tabelle 5.1 dargestellten Versuchsreihen.

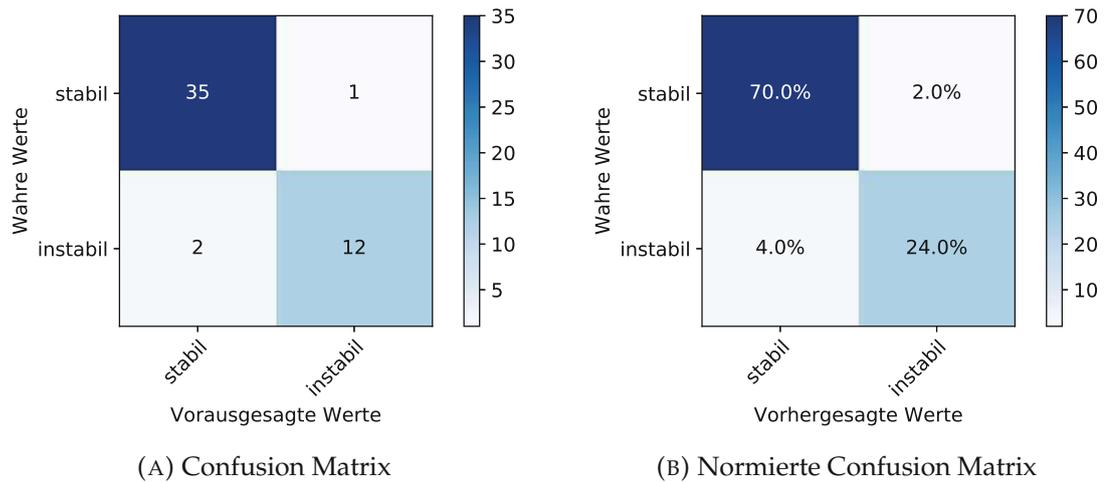


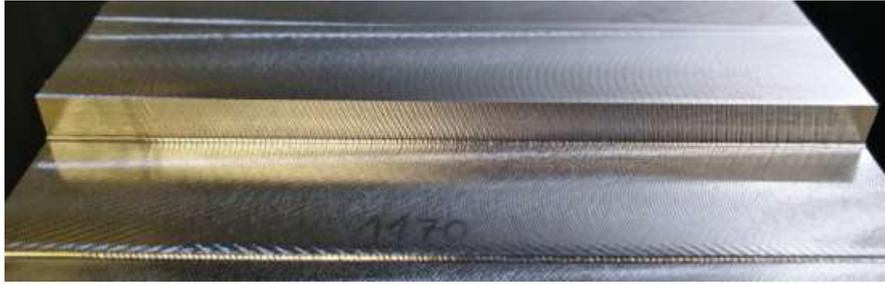
ABBILDUNG 5.7: Beurteilung des Modells durch Confusion Matrices

Versuche	Manuelle Klassifikation	Automatische Klassifikation	Art des Fehlers
1040	instabil (semistabil)	stabil	false-positive
1170	instabil	stabil	false-positive
2060	stabil	instabil	true-negative

TABELLE 5.1: Fehlerhaft klassifizierte Messreihen

## 5.4 Analyse der falsch klassifizierten Messreihen

In Tabelle 5.1 in Abschnitt 5.3.1 werden die vom Machine Learning Algorithmus fehlerhaft klassifizierten Messreihen gelistet. In diesem Abschnitt soll ausgewertet werden, wieso diese Messreihen als fehlerhaft klassifiziert wurden. Diese Versuche sind ein Beispiel dafür, dass auch eine durch Machine Learning geschaffene KI eine endliche Genauigkeit hat.



(A) Oberfläche des Versuches 1170



(B) Oberfläche des Versuches 2060

ABBILDUNG 5.8: Oberflächen der falsch klassifizierten Versuche

### 5.4.1 Versuch 1040 (false-positive / Fehler 1.Art)

Bei dem Versuch 1040 ist im Spektrum die Stabilität eindeutig erkennbar. Da bei der Versuchsreihe I es leider teilweise verabsäumt wurde, jede Oberfläche jedes Versuchs zu protokollieren, liegen hier leider keine aussagekräftigen Abbildungen vor. Aus diesem Grund kann nicht ausgeschlossen werden, dass die manuelle Klassifikation teilweise fehlerbehaftet ist und deshalb das neuronale Netz diese Fehler reproduziert.

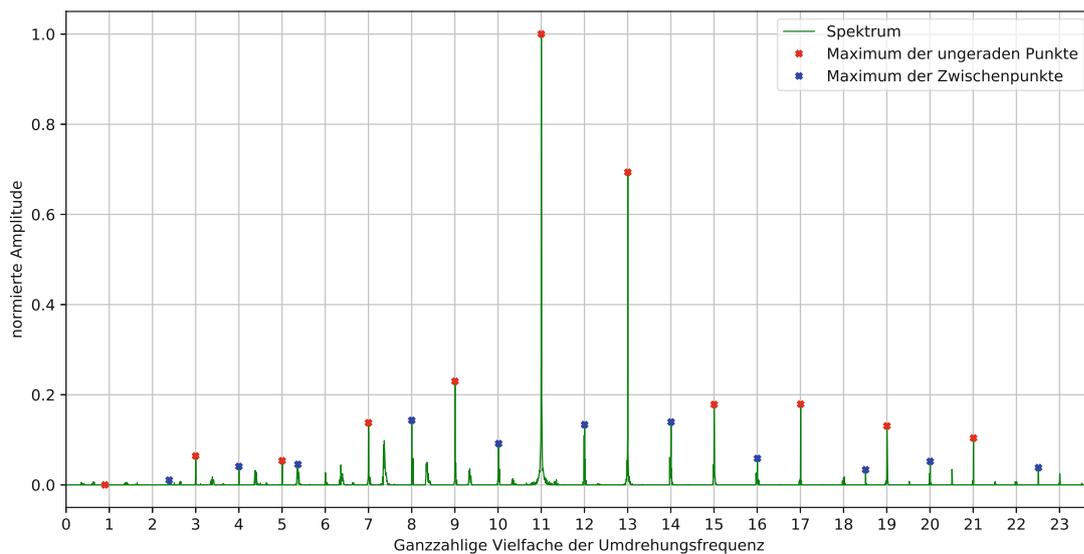


ABBILDUNG 5.9: Spektrum des Versuches 1040 (radial)

### 5.4.2 Versuch 1170 (false-positive / Fehler 1.Art)

Ähnlich wie beim vorherigen Versuch (1040) deutet auch beim Versuch 1170 das Spektrum klar auf einen stabilen Schnitt hin. Jedoch weist hier die Oberfläche in Abbildung 5.8a eindeutige und klar zu erkennende Rattermarken auf, auch wenn diese nicht unbedingt typisch sind und sich von den Rattermarken anderer Versuche durch ihre geometrische Erscheinung unterscheiden. Nichtsdestotrotz wurde dieser Versuch vom entwickelten Algorithmus falsch klassifiziert. Wie schon in Kapitel 4 erwähnt, wurde dieses Rattern durch die zweite Eigenfrequenz hervorgerufen. Das kann eine Erklärung dafür sein, dass die Beschleunigungsmessung des Sensorischen Werkzeughalters beziehungsweise die hier entwickelten Features nicht ausreichend sind um Rattern von höherwertigen Eigenformen zu erkennen.

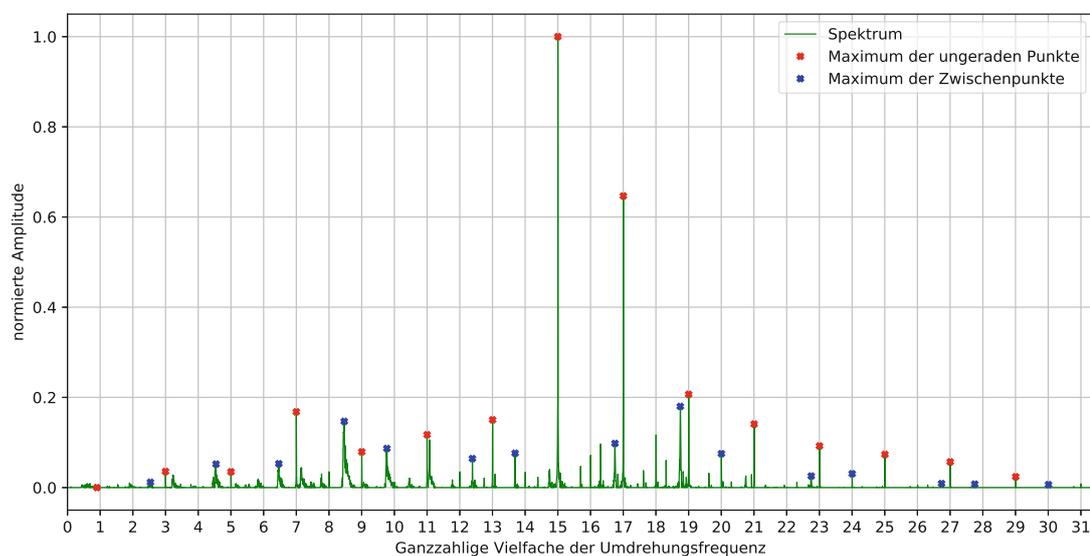


ABBILDUNG 5.10: Spektrum des Versuchs 1170 (radial)

### 5.4.3 Versuch 2060 (true-negative / Fehler 2.Art)

Beim Versuch 2060 liegt ein Fehler 2. Art vor. Dieser Versuch wird als stabil angesehen, vom Algorithmus aber als instabil klassifiziert. Das Spektrum deutet ebenfalls auf einen instabilen Prozess hin, da die Zwischenpunkte deutlich größere Amplituden als die ungeraden Punkte haben. Die Oberfläche hingegen ist jedoch frei von Rattermarken.

Aufgrund der Analyse der fehlerhaft klassifizierten Versuche kann die Genauigkeit des Modells mit 94% - 96% angegeben werden. Dies gilt jedoch für eine Anzahl von 99 Versuchen von denen 49 zum Trainieren und 50 zum Testen verwendet werden.

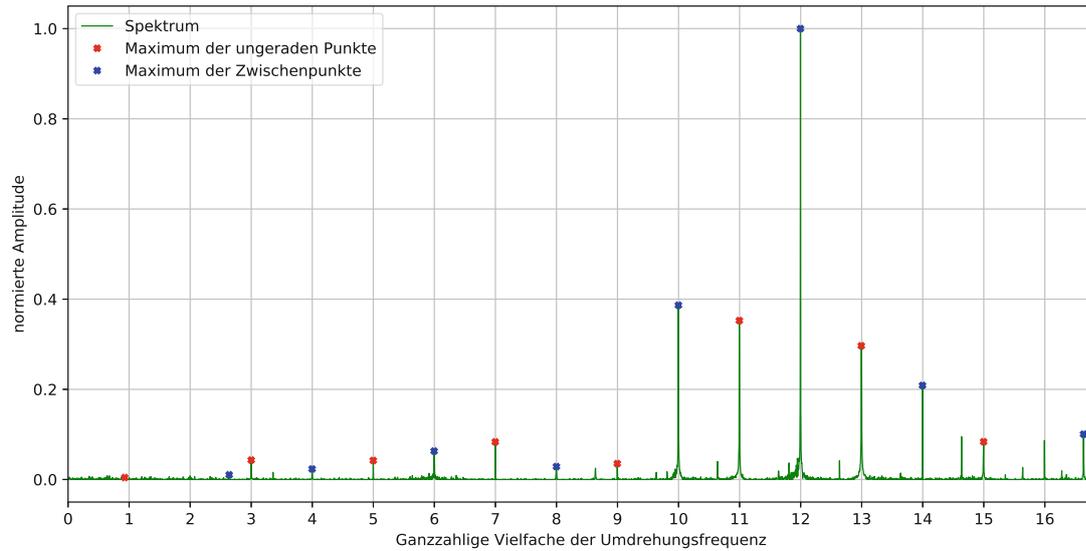


ABBILDUNG 5.11: Spektrum des Versuchs 2060 (radial)

## 5.5 Verwendung des Modells

Das vollständige neuronale Netzwerk kann wie in Anhang A.8 gezeigt, verwendet werden. Hierzu wird mit dem Python-Modul *joblib* eine *.pkl*-Datei erstellt. Diese Datei kann in weiterer Folge in ein System eingespeist werden, welches die Stabilität von Fräsprozessen dieser Art, mit Hilfe des Sensorischen Werkzeughalters und dem Feature-Generation-Skript (Anhang A.3 und A.5), klassifizieren kann. Wichtig hierbei ist die Verwendung der gleichen Features die beim Trainieren des neuronalen Netzwerks verwendet wurden. Der Input-Layer muss genau wie in Abbildung 5.6 bestückt werden. Die Normierung muss ebenfalls mit dem *StandardScaler()* vollführt werden. Auf diese Art und Weise liefert das neuronale Netzwerk die richtigen Ergebnisse.

## Kapitel 6

# Conclusio und Ausblick

In diesem Kapitel werden die Forschungsfragen, welche in der Einleitung der Arbeit gestellt wurden, beantwortet. In späterer Folge wird ein Ausblick auf weitere Anwendungsfelder des Sensorischen Werkzeughalters in Verbindung mit dieser Arbeit, gegeben.

### 6.1 Beantwortete Forschungsfragen

- Ist es möglich mit Hilfe der Sensordaten, eine analytisch berechnete Stabilitätskarte zu verifizieren? Wie genau stimmen die ausgewerteten Daten mit den theoretischen Funktionen überein?

Es wurde durch Versuche gezeigt, dass es möglich ist die Richtigkeit von analytisch erstellten Stabilitätskarten durch Versuche zu beweisen. Die Genauigkeit hängt stark von den geschätzten Parametern ab und kann nicht trivial quantifiziert werden. Es wird angenommen, dass kleine Parameteränderungen, wie zum Beispiel die von Versuch zu Versuch stetig weiter auskragenden Spindel der Fräsmaschine, das exakte Aufzeichnen von Stabilitätskarten beeinflusst.

- Kann anhand eines mechanischen Modells des Fertigungsprozesses, in Verbindung mit Daten des Werkzeughalters, das auftretende Rattern prognostiziert und wie in Abbildung 1.1 mit Senkung der Drehzahl und der Vorschubgeschwindigkeit der Prozess stabilisiert werden?

Ja, die Versuche haben gezeigt, dass Testschnitte die vorerst instabil waren, später bei geringerer Drehzahl und Vorschubgeschwindigkeit stabil abgelaufen sind. (siehe Tabelle 3.5)

- Kann diese Stabilisierung des Prozesses auch bei einer Steigerung der Drehzahl und Vorschubgeschwindigkeit erreicht werden, um in weiterer Folge die Bearbeitungszeit unter stabilen Schnittverhältnissen zu minimieren?

In Tabelle 3.6 wird im günstigsten Fall dieses Versuches deutlich gezeigt, dass ein instabiler Prozess durch Erhöhung der Drehzahl und Vorschubgeschwindigkeit durchaus stabil ablaufen kann und die Hauptzeit um mehr als die Hälfte verringert werden kann. Ein Vergleich des Werkzeugverschleißes bei einem instabilen

Prozess bei niedriger Drehzahl und einem stabilen Prozess bei hoher Drehzahl wird nicht getätigt.

- Ist es möglich anhand von Qualitätsindikatoren einen Rückschluss zu ziehen, in welchem Maß die Bearbeitungsparameter Vorschub und Drehzahl verschoben werden sollten, um höheres Zeitspannvolumen bei gleichzeitig stabilen Schnittverhältnissen zu realisieren?

Anhand der Qualitätsindikatoren ist dies alleine nicht möglich. Um die Eigenfrequenzen der Maschinenstruktur zu ermitteln, ist zum Beispiel eine Untersuchung mit dem Impulshammer oder eine Finite Elemente Simulation notwendig. Mit den gefundenen Eigenfrequenzen ist es möglich in Richtung der am ehesten stabilen Drehzahlbereiche zu regeln. Sollte die Regelung keine Verbesserung des Prozesses bewirken, kann durch den Sensorischen Werkzeughalter die immer noch vorliegende Instabilität erkannt werden und eine weitere Regelungsstrategie, wie das Reduzieren der Drehzahl, angewandt werden.

### 6.1.1 Gefundene Stabilitätsindikatoren

Es wird in dieser Arbeit gezeigt, dass grundsätzlich Streuungsmaße, wie Varianz beziehungsweise Standardabweichung und Interquartilsabstand, sehr gute Indikatoren im Zeitbereich zum Klassifizieren des Ratterns sind. Durch die Diskrete Fourier-Transformation (DFT) können Frequenzspektren erstellt werden, anhand derer weitere Indikatoren beziehungsweise Features erstellt werden können. Diese Features, vergleichen die Vielfachen der Umdrehungsfrequenz auf unterschiedlichste Art und Weise. Hier geht es um den Vergleich von ganzzahligen Vielfachen der Umdrehungsfrequenz, abhängig von der Anzahl der Schneiden des Werkzeuges.

### 6.1.2 Implementierter Algorithmus

Der in Python programmierte Machine Learning Algorithmus kann als Pickle-Datei (.pkl) abgespeichert werden und so von Systemen, welche Python unterstützen genutzt werden. Die Dateigröße des Neuronalen Netzwerks beträgt lediglich 312KB. Mit Hilfe dieser KI kann auch auf falsch klassifizierte Versuche aufmerksam gemacht werden. Bei den hier vorliegenden Daten beträgt die Genauigkeit 94% bis 96%.

## 6.2 Ausblick

In weiterer Folge kann das hier neu entwickelte System einer künstlichen Intelligenz zur Erkennung von Instabilitäten im Fräsprozess in die Richtung verbessert beziehungsweise erweitert werden, dass einerseits der Algorithmus die Anzahl an Trainingsdaten laufend erweitert und andererseits die Anzahl der Sensoren (Temperatur-, Schall-, Kraft-, Wegmessungen, etc.) erhöht wird. Damit könnte auch der Fräsprozess besser erfasst werden und die Genauigkeit der Rattererkennung deutlich verbessert

werden. Weiters könnte man auf andere Unstimmigkeiten einer Fräsmaschine aufmerksam machen und vor etwaigen Ermüdungs- und Verschleißformen warnen, wie zum Beispiel Lagerschäden. In Abbildung 6.1 wird gezeigt, dass auch der Zustand von Lagereinheiten mittels Schwingungsmessungen analysiert werden kann.

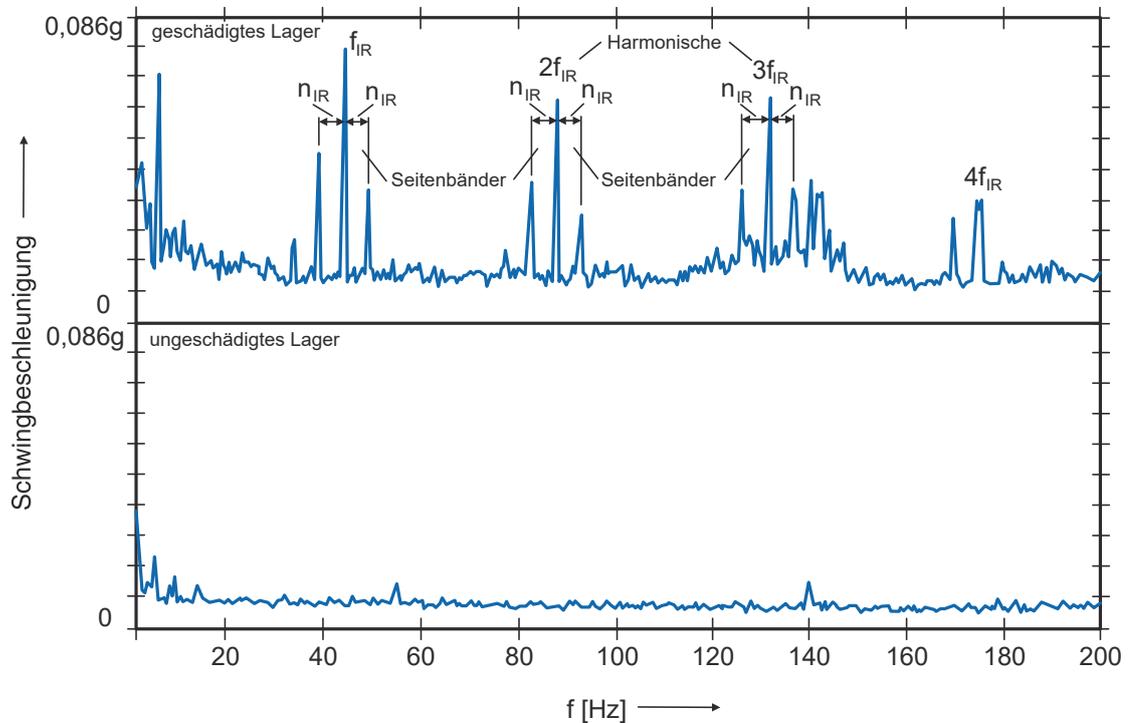


ABBILDUNG 6.1: Detektion von Lagerschäden als mögliches zukünftiges Anwendungsgebiet des STHs [Schlecht, 2009]

### 6.2.1 Automatisiertes Machine Learning

Das Modell, wie es zu diesem Zeitpunkt vorliegt, wird einmal mit Messdaten trainiert und hat eine gewisse fixe Genauigkeit. Ein weiterer Schritt wäre den Algorithmus so zu gestalten, dass dieser laufend während des Fräsprozesses aufgezeichnete Daten einliest und diese zum Machine Learning nutzt. Es muss hierbei jedoch ein Verfahren implementiert werden, welches die Testdaten automatisch klassifiziert, da die Klassifizierung bis jetzt manuell erfolgt.

### 6.2.2 Kognitive Systeme

Eine weitere Möglichkeit um die Detektion von Instabilitäten im Fräsprozess zu verbessern, wäre die Implementierung eines ganzen kognitiven Systems. Ein kognitives System kann mit Hilfe von mehreren Sensoren die Umgebung wahrnehmen, mit Machine Learning Algorithmen und gemessenen Daten Regelungsstrategien entwickeln und aktiv in das System eingreifen. Die Universität Ulm fasst diese Fähigkeiten unter den folgenden Gruppen zusammen [Ortmanns, 2019]:

- Perzeption und Kognition
- Lernen und Gedächtnis
- Planung, Entscheidungsfindung und Schlussfolgerung
- Interaktion

Um dieses Ziel zu erreichen, können mehrere Sensoren gekoppelt und die Gesamtheit aller Daten gemeinsam ausgewertet werden. Eine Zusammenfassung sensorunterstützter Detektion von Instabilitäten im Fräsprozess sind in [Kuljanic, Totis und Sortino, 2009] aufgelistet. Dabei ist entscheidend welche Technologie für welche Art von Messung verwendet wird und wo diese Sensoren platziert werden. So können zum Beispiel Beschleunigungssensoren auf der rotierenden Spindel oder am Spindelgehäuse befestigt werden oder Vibrationen können durch induktive Positionsmessung ermittelt werden.

Mit dieser Menge an Sensoren und dadurch gemessenen Daten sollen nicht nur die Instabilitäten durch auftretendes Rattern sondern auch Instabilitäten durch Schneidenbruch oder auch Schäden in der Spindellagerung genauestens detektiert und analysiert werden.

# Literatur

- Altintas, Yusuf und Erhan Budak (1995). „Analytical Prediction of Stability Lobes in Milling“. In: *CIRP Annals* 44.1, S. 357–362. DOI: [https://doi.org/10.1016/S0007-8506\(07\)62342-7](https://doi.org/10.1016/S0007-8506(07)62342-7).
- Altintas, Yusuf, Serafettin Engin und Erhan Budak (1999). „Analytical Stability Prediction and Design of Variable Pitch Cutters“. In: *Journal of Manufacturing Science and Engineering* 121.2, S. 173. DOI: [doi:10.1115/1.2831201](https://doi.org/10.1115/1.2831201).
- Altintas, Yusuf und M. Weck (2004). „Chatter Stability of Metal Cutting and Grinding“. In: *CIRP Annals* 53.2, S. 619–642. DOI: [10.1016/s0007-8506\(07\)60032-8](https://doi.org/10.1016/s0007-8506(07)60032-8).
- Baur, Matthias (2013). „Aktives Dämpfungssystem zur Ratterunterdrückung an spannenden Werkzeugmaschinen – Entwurf und experimentelle Validierung –“. Diss. Technische Universität München.
- Bleicher, Friedrich (2019a). „Cognitive machining by sensor integration in tooling system“. In: *MTTRF and EiAM-CNC - Annual Meeting 2019*.
- (2019b). *Skriptum: Auslegung von Werkzeugmaschinen VO*. Hrsg. von Friedrich Bleicher. Institut für Fertigungstechnik und Photonische Technologien.
- Brecher, Christian und Manfred Weck (2006). *Werkzeugmaschinen 5*. Springer Berlin Heidelberg. DOI: [10.1007/978-3-540-32951-0](https://doi.org/10.1007/978-3-540-32951-0).
- (2017). *Werkzeugmaschinen Fertigungssysteme 2*. Springer Berlin Heidelberg. DOI: [10.1007/978-3-662-46567-7](https://doi.org/10.1007/978-3-662-46567-7).
- Budak, Erhan (1994). „Mechanics and dynamics of milling thin walled structures“. Diss. University of British Columbia.
- Budak, Erhan und Yusuf Altintas (1998a). „Analytical Prediction of Chatter Stability in Milling—Part I: General Formulation“. In: *Journal of Dynamic Systems, Measurement, and Control* 120.1, S. 22. DOI: [10.1115/1.2801317](https://doi.org/10.1115/1.2801317).
- (1998b). „Analytical Prediction of Chatter Stability in Milling—Part II: Application of the General Formulation to Common Milling Systems“. In: *Journal of Dynamic Systems, Measurement, and Control* 120.1, S. 31. DOI: [10.1115/1.2801318](https://doi.org/10.1115/1.2801318).
- Dettmers, Tim (2015). *Deep Learning in a Nutshell*. URL: <https://devblogs.nvidia.com/deep-learning-nutshell-core-concepts/> (besucht am 17.07.2019).
- EMCO GmbH (2019). *EMCO UMILL750*. URL: <https://www.emco-world.com/de/produkte/industrie/fraesen/cat/480/d/2/p/1000878%2C480/pr/umill-750.html> (besucht am 03.07.2019).
- Esmael, Bilal u. a. (2013). „A Statistical Feature-Based Approach for Operations Recognition in Drilling Time Series“. In: *International Journal of Computer Information Systems and Industrial Management Applications*. 5, S. 454–461.

- Fumo, David (2017). *Types of Machine Learning Algorithms You Should Know*. URL: <https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861> (besucht am 17.07.2019).
- Geurts, Pierre, Damien Ernst und Louis Wehenkel (2006). „Extremely randomized trees“. In: *Machine Learning* 63.1, S. 3–42. DOI: 10.1007/s10994-006-6226-1.
- Gradišek, Janez u. a. (2005). „On stability prediction for milling“. In: *International Journal of Machine Tools and Manufacture* 45.7-8, S. 769–781. DOI: 10.1016/j.ijmachtools.2004.11.015.
- Insperger, Tamas, David Lehotzky und Gabor Stepan (2015). „Regenerative delay, parametric forcing and machine tool chatter: A review“. In: *IFAC-PapersOnLine* 48.12. 12th IFAC Workshop on Time Delay Systems TDS 2015, S. 322–327. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2015.09.398>. URL: <http://www.sciencedirect.com/science/article/pii/S2405896315015013>.
- Klocke, Fritz und Wilfried König (2008). *Fertigungsverfahren 1*. 8. Auflage. Springer Berlin Heidelberg. DOI: 10.1007/978-3-540-35834-3.
- Kuljanic, E., G. Totis und M. Sortino (2009). „Development of an intelligent multisensor chatter detection system in milling“. In: *Mechanical Systems and Signal Processing* 23.5, S. 1704–1718. ISSN: 0888-3270. DOI: <https://doi.org/10.1016/j.ymsp.2009.01.003>. URL: <http://www.sciencedirect.com/science/article/pii/S0888327009000053>.
- Lacerda, H. B und V. T Lima (2004). „Evaluation of Cutting Forces and Prediction of Chatter Vibrations in Milling“. In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 26. DOI: 10.1590/S1678-58782004000100013.
- Li, Zhongqun, Zhikang Wang und Xiaofang Shi (2017). „Fast prediction of chatter stability lobe diagram for milling process using frequency response function or modal parameters“. In: *The International Journal of Advanced Manufacturing Technology* 89.9-12, S. 2603–2612. DOI: 10.1007/s00170-016-9959-4.
- Loy, James (2018). *How to build your own Neural Network from scratch in Python*. URL: <https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6> (besucht am 11.07.2019).
- MathWorks (2016). *MathWorks - Section 1: Introducing Machine Learning*. URL: <https://www.mathworks.com/campaigns/offers/machine-learning-with-matlab.html>.
- (2018). *Get Ready for AI with MATLAB*. URL: <https://www.mathworks.com/campaigns/offers/ai-with-matlab.html> (besucht am 26.09.2019).
- My Tool IT GmbH (2019). *IC0tronic System*. URL: <https://www.mytoolit.com/IC0tronic/> (besucht am 03.07.2019).
- Ortmanns, Maurits (2019). *Was sind Kognitive Systeme?* URL: <https://www.uni-ulm.de/in/fakultaet/studium/studiengaenge/cognitive-systems/about-cognitive-systems/> (besucht am 29.07.2019).
- Pedregosa, Fabian u. a. (2011). „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12, S. 2825–2830.

- Quintana, Guillem und Joaquim Ciurana (2011). „Chatter in machining processes: A review“. In: *International Journal of Machine Tools and Manufacture* 51.5, S. 363–376. DOI: 10.1016/j.ijmachtools.2011.01.001.
- Ramsauer, Christoph (2019). „Parameteranalyse zur Steuerung komplexer Fräsprozesse“. Magisterarb. Technische Universität Wien.
- Rubeo, Mark A. und Tony L. Schmitz (2016). „Mechanistic force model coefficients: A comparison of linear regression and nonlinear optimization“. In: *Precision Engineering* 45, S. 311–321. DOI: 10.1016/j.precisioneng.2016.03.008.
- Schlecht, Berthold (2009). *Maschinenelemente 2: Getriebe, Verzahnungen und Lagerungen*. Pearson Deutschland GmbH. ISBN: 978-3827371461.
- Sharma, Pulkit (2018). *The Ultimate Guide to 12 Dimensionality Reduction Techniques*. URL: <https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/> (besucht am 08.07.2019).
- Suhm, Bernhard (2018). *Heart Sound Classifier*. URL: <https://www.mathworks.com/matlabcentral/fileexchange/65286-heart-sound-classifier> (besucht am 11.07.2019).
- Tekeli, A. und Erhan Budak (2004). „Increasing Chatter Free Material Removal Rate In Die and Mold Machining“. In: *CIRP Annals* 54.1, S. 353–356. DOI: [https://doi.org/10.1016/S0007-8506\(07\)60121-8](https://doi.org/10.1016/S0007-8506(07)60121-8).
- Tobias, S. A. (1965a). *Machine-Tool Vibration*. Blackie und Son Ltd.
- Tobias, S.A. (1965b). „Vibration of machine tools“. In: *Production Engineer*, S. 599–608. DOI: 10.1049/tpe:19640084.
- Yang, Wei, Kuanquan Wang und Wangmeng Zuo (2012). „Neighborhood Component Feature Selection for High-Dimensional Data“. In: *Journal of Computers* 7.1. DOI: 10.4304/jcp.7.1.161-168.
- Yangui, Hedi u. a. (2010). „Influence of Cutting and Geometrical Parameters on the Cutting Force in Milling“. In: *Scientific Research* 02.10, S. 751–761. DOI: 10.4236/eng.2010.210097.
- Zhou, Linda (2018). *Simplify Machine Learning Pipeline Analysis with Object Storage*. URL: <https://blog.westerndigital.com/machine-learning-pipeline-object-storage/> (besucht am 17.07.2019).



# Abbildungsverzeichnis

1.1	Optimale Prozessregelung . . . . .	2
1.2	Übersicht des ICOTRONIC Systems [My Tool IT GmbH, 2019] . . . . .	4
1.3	ungefiltertes Ausgangssignal des Sensorischen Werkzeughalters für verschiedene Drehzahlen der Spindel . . . . .	5
1.4	Tiefpass-gefiltertes Signal des Sensorischen Werkzeughalters für verschiedene Drehzahlen der Spindel . . . . .	5
1.5	Ausgleichsfunktion des zur Beschleunigung proportionalen Sensorsignales des Sensorischen Werkzeughalters in y-Richtung (radial) . . . . .	6
1.6	Vergleich zweier Fräszentren ähnlicher Größe . . . . .	8
1.7	Aus Erfahrungswerten für modale Parameter berechnete Stabilitätskarte: gelb $\hat{=}$ Hüller Hille / gelb + rot $\hat{=}$ EMCO . . . . .	9
1.8	Experimentelle Stabilitätskarte [Brecher und Weck, 2017] . . . . .	10
2.1	Allgemeine Stabilitätskarte [Quintana und Ciurana, 2011] . . . . .	11
2.2	Übertragungsfunktion für eine Eigenfrequenz von 1000Hz . . . . .	16
2.3	System mit zwei Freiheitsgraden . . . . .	17
2.4	Phasenverschiebung der Spannungsdickenänderung [Quintana und Ciurana, 2011] . . . . .	18
2.5	Beschreibung des Faktors k [Baur, 2013] . . . . .	18
2.6	System mit einem Freiheitsgrad . . . . .	20
2.7	Drei Schleifen des Berechnungsalgorithmus . . . . .	21
2.8	Selbst entwickelte Stabilitätsdiagramme mit Parametern nach [Li, Wang und Shi, 2017] . . . . .	22
2.9	Variation der radialen Schnittbreite ähnlich wie in [Li, Wang und Shi, 2017] . . . . .	23
2.10	Vergleich mit dem Ergebnis von CutPro © . . . . .	24
2.11	Stabilitätskarte mit min./max. Grenzschnittiefen . . . . .	25
3.1	Überblick des Versuchsaufbaus . . . . .	30
3.2	Ermittelte Übertragungsfunktionen des Impulshammerversuches . . . . .	32
3.3	Berechnete und an die Versuchsreihe I angepasste Stabilitätskarte . . . . .	34
3.4	Verwendetes Messmikrofon . . . . .	35
3.5	Darstellung der Versuchsreihen . . . . .	36
3.6	Versuchsreihe I mit Stabilitätskarte . . . . .	37
3.7	Vergleich der Oberfläche mehrerer Schnitte . . . . .	38

3.8	Zeitreihendaten eines stabilen Prozesses . . . . .	40
3.9	Zeitreihendaten eines instabilen Prozesses . . . . .	41
3.10	Amplitudenspektren eines stabilen Prozesses . . . . .	43
3.11	Amplitudenspektren eines instabilen Prozesses . . . . .	44
4.1	Statistische Standard-Features [Esmael u. a., 2013] . . . . .	46
4.2	Darstellung der ungeraden Punkte (ONP) und der Zwischenpunkte (ISP) im Amplitudenspektrum . . . . .	48
4.3	Feature 1 für Versuchsreihe I mit Stabilitätskarte . . . . .	50
4.4	Feature 1 für Versuchsreihe I & II . . . . .	51
4.5	Feature Selection mit dem Random Forest Verfahren . . . . .	53
5.1	Ebenen von KI [MathWorks, 2018] . . . . .	55
5.2	Unterschied von Supervised Learning und Unsupervised Learning [Zhou, 2018] . . . . .	57
5.3	Ablauf des Supervised Learnings [Bleicher, 2019a] . . . . .	58
5.4	Aufbau eines neuronalen Netzes für DL [Dettmers, 2015] . . . . .	59
5.5	MATLAB App ClassificationLearner mit Daten aus [Suhm, 2018] . . . . .	61
5.6	Aufbau des neuronalen Netzes . . . . .	62
5.7	Beurteilung des Modells durch Confusion Matrices . . . . .	63
5.8	Oberflächen der falsch klassifizierten Versuche . . . . .	64
5.9	Spektrum des Versuchs 1040 (radial) . . . . .	64
5.10	Spektrum des Versuchs 1170 (radial) . . . . .	65
5.11	Spektrum des Versuchs 2060 (radial) . . . . .	66
6.1	Detektion von Lagerschäden als mögliches zukünftiges Anwendungsgebiet des STHs [Schlecht, 2009] . . . . .	69

# Tabellenverzeichnis

1.1	Versuchsparameter aus [Brecher und Weck, 2017] . . . . .	9
2.1	Modellierungsarten der Schnittkraft [Yangui u. a., 2010] . . . . .	12
3.1	Versuchsparameter . . . . .	30
3.2	Verwendete Sensorik . . . . .	31
3.3	Verwendete Eigenfrequenzen . . . . .	33
3.4	Durchgeführte Versuche . . . . .	34
3.5	Vergleich des Übergangs zwischen stabilen und instabilen Systemverhaltens aus Versuchsreihe II . . . . .	35
3.6	Größtmögliche Reduktion der Hauptzeit bei gleichzeitiger Stabilisierung des Prozesses in Versuchsreihe I . . . . .	36
4.1	Reihung anhand der Varianz der Features . . . . .	52
4.2	Reihung anhand eines Korrelations-Kriterium . . . . .	52
4.3	Reihung anhand des FPR-Kriteriums . . . . .	53
4.4	Reihung durch die NCA . . . . .	53
5.1	Fehlerhaft klassifizierte Messreihen . . . . .	63



# Abkürzungsverzeichnis und Physikalische Konstanten

<b>BLE</b>	<b>Bluetooth Low Energy</b>
<b>CC</b>	<b>Charging Cradle</b>
<b>DL</b>	<b>Deep Learning</b>
<b>FNR</b>	<b>False Negative Rate</b>
<b>FPR</b>	<b>False Positive Rate</b>
<b>GIMP</b>	<b>GNU Image Manipulation Program</b>
<b>HSK</b>	<b>HohlSchaftKegel</b>
<b>IFT</b>	<b>Institut für FertigungsTechnik und Photonische Technologien</b>
<b>ISP</b>	<b>InterSection Points</b>
<b>IQR</b>	<b>InterQuartil Range</b>
<b>kNN</b>	<b>k - Nearest Neighbor</b>
<b>KI</b>	<b>Künstliche Intelligenz</b>
<b>MFCC</b>	<b>Mel Frequency Cepstral Coefficients</b>
<b>MLP</b>	<b>MultiLayer Perceptron</b>
<b>ML</b>	<b>Machine Learning</b>
<b>NC</b>	<b>Numerical Control</b>
<b>NCA</b>	<b>Neighborhood Component Analysis</b>
<b>NN</b>	<b>Neuronales Netz(werk)</b>
<b>ONP</b>	<b>Odd-Numbered Points</b>
<b>PKL</b>	<b>PicKLe</b>
<b>SPU</b>	<b>Signal ProcessingUnit</b>
<b>STH</b>	<b>Sensory ToolHolder</b>
<b>STU</b>	<b>Stationary TrancieverUnit</b>
<b>SVM</b>	<b>Support Vector Machine</b>
<b>TNR</b>	<b>True Negative Rate</b>
<b>TPR</b>	<b>True Positive Rate</b>

Normalfallbeschleunigung  $1g$   $9.81 \frac{m}{s^2}$



# Symbolverzeichnis

$a_e, b$	(radiale) Schnittbreite	$m$
$a_{lim}$	(axiale) Grenzschnitttiefe	$m$
$a_p$	axiale Schnitttiefe	$m$
$c_m$	modale Dämpfung	$\frac{kg}{s}$
$D$	Fräserdurchmesser	$m$
$f_n$	Eigenfrequenz	$Hz$
$f_s$	Abtastrate	$Hz$
$f_{ZEF}$	Zahneingriffsfrequenz	$Hz$
$F_r$	radiale Schnittkraftkomponente	$N$
$F_t$	tangentiale Schnittkraftkomponente	$N$
$\underline{\underline{G}}$	Übertragungsfrequenzgangmatrix, (2x2)	$\frac{m}{N}$
$h$	Spanungsdicke	$m$
$\underline{\underline{I}}$	Einheitsmatrix, (nxn)	—
$j$	komplexe Zahl	—
$k_m$	modale Steifigkeit	$\frac{m}{N}$
$k$	Ordnung des Ratterverhaltens	—
$K_{c1.1}$	spezifische Schnittkraft	$\frac{N}{m^2}$
$K_{rc}$	radialer Schnittkraftkoeffizient	$\frac{N}{m^2}$
$K_{tc}$	tangentialer Schnittkraftkoeffizient	$\frac{N}{m^2}$
$\Lambda$	Lösung des Eigenwertproblems	—
$m_c$	Werkstoffkennwert	—
$m_m$	modale Masse	$kg$
$\underline{\underline{M}}$	Massenmatrix (2x2)	—
$n$	Spindeldrehzahl	$\frac{1}{min}$
$N$	Schneidenanzahl des Fräasers	—
$T_{ZEF}$	Periodendauer des Zahneingriffs	$s$
$\underline{\underline{\alpha}}$	gerichtete dynamische Fräskraftkoeffizientenmatrix, (2x2)	—
$\epsilon$	Phasenverschiebung der Spandickenänderung	$rad$
$\xi_m$	Dämpfungsparameter	—
$\phi$	Spanungswinkel	$rad$
$\phi_{start}$	Startwinkel des Zahneingriffes	$rad$
$\phi_{exit}$	Austriffswinkel des Zahneingriffes	$rad$
$\omega_c$	Ratterfrequenz	$\frac{rad}{s}$
$\omega_n$	Eigenfrequenz	$\frac{rad}{s}$
$\omega_{ZEF}$	Zahneingriffsfrequenz	$\frac{rad}{s}$



## Anhang A

# Python Code

### A.1 Berechnung und Plotten der Stabilitätskarte

```
# -*- coding: utf-8 -*-

"""
Algorithmus für 2DoF Stabilitätskarten      Oswald 2019
"""

#-----
#Datum:          05.03.2019 (Tuesday)
#Uhrzeit:        16:09:00
#Bearbeiter:     Ralf Oswald
#Projekt:        Diplomarbeit_Oswald
#Datei:          2DOF_stability_lobe.py

#Importieren der Bibliotheken
from math import cos as cos
from math import acos as acos
from math import sin as sin
from math import atan as atan
from math import pi as pi

import numpy as np

import pandas as pd

import matplotlib
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
from matplotlib.font_manager import FontProperties

#Definition der Parameter
```

```

N = 4 #[]
Ktc = 1100e6 #[N/m^2]
Krc = 600e6 #[N/m^2]
Kr = Kr/Kt #[]

w_nx_1 = 1050*2*pi #[rad/s]
w_ny_1 = 1050*2*pi #[rad/s]

xi_x_1 = 0.05 #[]
xi_y_1 = xi_x_1

k_x_1 = 60000000 #[N/m]
k_y_1 = k_x_1 #[N/m]

bD = 20/25 #b/D-Verhältnis [-]

#Start der Berechnung für Eigenmode 1
phi = acos(1-2*bD)

a_xx = 0.5*(cos(2*phi) - 2*Kr*phi + Kr*sin(2*phi))
a_xy = 0.5*(-sin(2*phi) - 2*phi + Kr*cos(2*phi))
a_yx = 0.5*(-sin(2*phi) + 2*phi + Kr*cos(2*phi))
a_yy = 0.5*(-cos(2*phi) - 2*Kr*phi - Kr*sin(2*phi))
print("axx = ", a_xx, "\naxy = ", a_xy,
      "\nayx = ", a_yx, "\nayy = ", a_yy)

k = np.arange(0,20,1)
w_c = np.arange(1,15000,1)

a_lim_L1_1 = list()
a_lim_L2_1 = list()
rpm_L1_1 = list()
rpm_L2_1 = list()

for i in range(0,len(k)):
    a_lim_L1_1.append(np.array([]))
    rpm_L1_1.append(np.array([]))

    for j in range(0, len(w_c)):
        G_xx_1 = (w_nx_1**2/k_x_1)/(w_nx_1**2 - w_c[j]**2
                    + 2*xi_x_1*w_nx_1*w_c[j]*1j)
        G_yy_1 = (w_ny_1**2/k_y_1)/(w_ny_1**2 - w_c[j]**2

```

```

+ 2*xi_y_1*w_ny_1*w_c[j]*1j)

a0_1 = (G_xx_1*G_yy_1)*(a_xx*a_yy-a_xy*a_yx)
a1_1 = a_xx*G_xx_1 + a_yy*G_yy_1

L1_1 = (-1/(2*a0_1))*(a1_1+(a1_1**2 - 4*a0_1)**0.5)
L2_1 = (-1/(2*a0_1))*(a1_1-(a1_1**2 - 4*a0_1)**0.5)

kappa_L1_1 = L1_1.real/L1_1.imag
kappa_L2_1 = L2_1.real/L2_1.imag

a_lim_loop1_1 = (-2*pi*L1_1.imag)*
(kappa_L1_1+1/kappa_L1_1)/(N*Ktc)
a_lim_loop1_1 = a_lim_loop1_1*1000 #m-->mm
a_lim_loop2_1 = (-2*pi*L2_1.imag)*
(kappa_L2_1+1/kappa_L2_1)/(N*Ktc)
a_lim_loop2_1 = a_lim_loop2_1*1000 #m-->mm

if 0 < a_lim_loop1_1 < 10:
    a_lim_L1_1[i] = np.append(a_lim_L1_1[i], a_lim_loop1_1)
    epsilon_loop1_1 = pi - 2*atan(1/kappa_L1_1)
    T_ZEF1 = (2*pi*k[i]+epsilon_loop1_1)/w_c[j]
    rpm_L1_1[i] = np.append(rpm_L1_1[i], 60/(N*T_ZEF1))

if 0 < a_lim_loop2_1 < 10:
    a_lim_L2_1[i] = np.append(a_lim_L2_1[i], a_lim_loop2_1)
    epsilon_loop2_1 = pi - 2*atan(1/kappa_L2_1)
    T_ZEF2 = (2*pi*k[i]+epsilon_loop2_1)/w_c[j]
    rpm_L2_1[i] = np.append(rpm_L2_1[i], 60/(N*T_ZEF2))

    print(str(round(100*i/len(k))+"%") #Fortschrittsanzeige
print("100%")

#Auswahl der Lösung Nr.2 der quadratischen Gleichung für Eigenmode 1
a_lim_1 = a_lim_L2_1
rpm_1 = rpm_L2_1

del(a_lim_L1_1 ,a_lim_L2_1)
del(rpm_L1_1 ,rpm_L2_1)

#Berechnung der minimalen Grenzschnitttiefe
a_grenz_1 = np.empty(shape=(len(k),1))

```

```

rpm_grenz_1 = np.empty(shape=(len(k),1))

for i in range(0,len(a_lim_1)):
    a_grenz_1[i] = np.amin(a_lim_1[i])
    rpm_grenz_1[i] = rpm_1[i][np.argmin(a_lim_1[i])]
    print(i, ", a_grenz_1 = ", a_grenz_1[i],
          " bei n_grenz_1 = ", rpm_grenz_1[i])

#Berechnung der maximalen Grenzschnitttiefe / Schnittpunkte der Rattersäcke
a_schnitt_1 = np.empty(shape=(len(k)-1),1)
rpm_schnitt_1 = np.empty(shape=(len(k)-1),1)
a_schnitt_2 = np.empty(shape=(len(k)-1),1)
rpm_schnitt_2 = np.empty(shape=(len(k)-1),1)
a_schnitt_3 = np.empty(shape=(len(k)-1),1)
rpm_schnitt_3 = np.empty(shape=(len(k)-1),1)

for i in range(0,len(k)-1):
    rpm_schnitt_1[i] = (60*w_nx_1/((k[i]+1)*N*2*pi))
    a_schnitt_1[i] = a_lim_1[i+1][np.argmin(np.square(rpm_1[i+1]
-rpm_schnitt_1[i])))]
    print(i, ", a_schnitt_1 = ", a_schnitt_1[i],
          " bei rpm_schnitt_1 = ", rpm_schnitt_1[i])

#Löschen der Punkte oberhalb der Schnittpunkte der Rattersäcke
lower = 1.02 #Hilfsgrößen zum Anpassen der Schnittpunkte
upper = 0.99 #Hilfsgrößen zum Anpassen der Schnittpunkte

a_lim_1cut = list()
rpm_1cut = list()

for i in range(0,len(k)):
    a_lim_1cut.append([])
    rpm_1cut.append([])
    for j in range(0, len(a_lim_1[i])):
        if i==0:
            if rpm_1[i][j] >= lower*rpm_schnitt_1[i]:
                rpm_1cut[i].append(rpm_1[i][j])
                a_lim_1cut[i].append(a_lim_1[i][j])
        if 1<=i<=len(k)-2:
            if lower*rpm_schnitt_1[i] <= rpm_1[i][j]
            <= upper*rpm_schnitt_1[i-1]:
                rpm_1cut[i].append(rpm_1[i][j])

```

```

        a_lim_1cut[i].append(a_lim_1[i][j])
    if i==len(k)-1:
        if rpm_1[i][j] <= upper*rpm_schnitt_1[i-1]:
            rpm_1cut[i].append(rpm_1[i][j])
            a_lim_1cut[i].append(a_lim_1[i][j])

#Aufbereiten der Daten zum Plotten der Diagramme
rpm_1_plot = []
a_lim_1_plot = []

for i in range(0,len(k)):
    for j in range(0,len(a_lim_1cut[i])):
        a_lim_1_plot.append(a_lim_1cut[i][j])
        rpm_1_plot.append(rpm_1cut[i][j])

rpm_1_plot = np.asarray(rpm_1_plot)
a_lim_1_plot = np.asarray(a_lim_1_plot)

ind_1 = rpm_1_plot.argsort()
rpm_1_plot = rpm_1_plot[ind_1]
a_lim_1_plot = a_lim_1_plot[ind_1]

#Einlesen der Versuchdaten (stabile, semistabile und instabile Punkte)
data = pd.read_csv("data.csv", sep=";")

points_stable = [], []
points_unstable = [], []
points_semistable = [], []

for i in range(0,data.shape[0]):
    if data["stability"][i]=="stabil":
        points_stable[0].append(data["adoc"][i])
        points_stable[1].append(data["rpm"][i])
    if data["stability"][i]=="instabil":
        points_unstable[0].append(data["adoc"][i])
        points_unstable[1].append(data["rpm"][i])
    if data["stability"][i]=="semistabil":
        points_semistable[0].append(data["adoc"][i])
        points_semistable[1].append(data["rpm"][i])

#Plotten der Stabilitätskarten
matplotlib.rcParams.update({'font.size': 10}) #normal=10

```

```
fig = plt.figure(figsize = (12,6), dpi=300, facecolor='w')
plt.plot(rpm_1_plot, a_lim_1_plot, linewidth=1.0, label="1. Mode")

plt.plot(points_stable[1], points_stable[0], 'bP',
         markersize=5, label="stabil")
plt.plot(points_unstable[1], points_unstable[0], 'rX',
         markersize=5, label="instabil")
plt.plot(points_semistable[1], points_semistable[0], 'gD',
         markersize=5, label="semistabil")

font0 = FontProperties()
font0.set_style('oblique')
font0.set_size('smaller')

for i in range(0, data.shape[0]):
    plt.text(data["rpm"][i], data["adoc"][i],
            str(data["testrun"][i])+" ",
            fontproperties=font0, horizontalalignment='right',
            verticalalignment='center', fontsize=8)

plt.xlim(xmin=3640, xmax=9390)
plt.ylim(ymin=0.5, ymax=7.5)
plt.xticks(np.arange(3640,9391,250))
plt.yticks(np.arange(0.5,7.51,0.5))

plt.legend()

plt.xlabel("Drehzahl [u/min]", fontsize=10)
plt.ylabel("axiale Schnitttiefe [mm]", fontsize=10)
plt.grid(linestyle='-', linewidth=0.2)

fig.savefig("Stabilitaetskarte.pdf", bbox_inches='tight', dpi=fig.dpi)
plt.close()
```

## A.2 Brechnung und Plotten des Übertragungsfrequenzgangs

```
# -*- coding: utf-8 -*-

"""
Berechnung und Plotten des Übertragungsfrequenzgangs   Oswald 2019
"""

#-----
#Datum:          24.07.2019 (Wednesday)
#Uhrzeit:        22:52:00
#Bearbeiter:     Ralf Oswald
#Projekt:        Diplomarbeit_Oswald
#Datei:          Uebertragungsfrequenzgang.py

# Importieren der Bibliotheken
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
from math import pi as pi

#Start der Berechnung
G_jwc = list()
G_jwc_real = list()
G_jwc_imag = list()

w_c = np.arange(1,15000,1) #[rad/s]
w_nx_1 = 1000*2*pi #[rad/s]
xi_x_1 = 0.05 #[]
k_x_1 = 60000000 #[N/m]

for i in range(0, len(w_c)):
    G_jwc_1 = (w_nx_1**2/k_x_1)/(w_nx_1**2
        - w_c[i]**2 + 2*xi_x_1*w_nx_1*w_c[i]*1j)
    G_jwc.append(G_jwc_1)

for i in range(0, len(G_jwc)):
    G_jwc_real.append(G_jwc[i].real)
    G_jwc_imag.append(G_jwc[i].imag)

G_jwc_real = np.asarray(G_jwc_real)
G_jwc_imag = np.asarray(G_jwc_imag)
```

```
#Plotten des Übertragungsfrequenzgangs
matplotlib.rcParams.update({'font.size': 10})
fig = plt.figure(figsize = (12,6), dpi=300, facecolor='w')

plt.plot(w_c/(2*pi), G_jwc_real*(10**6), color="#1f77b4", linewidth=1.2,
         label="Realteil der Übertragungsfunktion")
plt.plot(w_c/(2*pi), -G_jwc_imag*(10**6), color="#ff7f0e", linewidth=1.2,
         label="Imaginärteil der Übertragungsfunktion")

plt.xlim(xmin=0, xmax=2000)
plt.xticks(np.arange(0,2001,250))

plt.legend()
plt.xlabel("Frequenz " + r"$\omega_{c}$" + " [Hz]", fontsize=12)
plt.ylabel("Amplitude " + r"$H_{ij}$" + " [\u03bcm/N]", fontsize=12)

plt.grid(linestyle='-', linewidth=0.2)
fig.savefig("Uebertragungsfrequenzgang.pdf", bbox_inches='tight', dpi=fig.dpi)
plt.close()
```

## A.3 Selbsterstellte Funktionen

```

# -*- coding: utf-8 -*-

"""
Selbsterstellte Funktionen    Oswald 2019
"""

#-----
#Datum:          24.07.2019 (Wednesday)
#Uhrzeit:        23:09:00
#Bearbeiter:     Ralf Oswald
#Projekt:        Diplomarbeit_Oswald
#Datei:          Funktionen.py

#Importieren der Bibliotheken
import numpy as np
from scipy import fftpack
from scipy.fftpack import fft, fftfreq, fftshift
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd

#Einlesen des Zeitsignals (altes Schema)
def get_signal_old(folder, filename, extension):
    data = pd.read_csv(folder+filename+"."+extension, header=None, sep=';',
                       skiprows=1, engine='python', skipfooter=0,
                       error_bad_lines=False)

    data_clean = data
    col_names = ['accx']
    data_clean.columns = col_names
    return(data_clean)

#Einlesen des Zeitsignals (neues Schema)
def get_signal_new(folder, filename, extension):
    data = pd.read_csv(folder+"/"+filename+"."+extension, header=None,
                       sep='\(|ms\)|s+|;', skiprows=10, engine='python',
                       skipfooter=50, error_bad_lines=False)
    data_clean = data.drop([0,1,2,3,4,5,6,8,9,11,12,14], axis=1)
    col_names = ['accx', 'accy', 'accz']
    data_clean.columns = col_names

```

```
return(data_clean)
```

```
#Berechnung der Zeiten
```

```
def timevalues(signal, fs, factor, rpm, vf=0):
    for i in range(0,len(signal)-1):
        if signal[i+1] > factor*signal[i]:
            time0 = (i/fs) #t0 = erster Kontakt
            break
        else:
            time0 = 0
    if vf==0:
        vf=rpm*0.1*4
    time1 = time0 + (12.5/vf)*60 #t1 = erste volle Schnittbreite
    time2 = time0 + (100/vf)*60 #t2 = letzte volle Schnittbreite
    time3 = time0 + (112.5/vf)*60 #t3 = letzter Kontakt
    return([time0,time1,time2,time3])
```

```
#Schneiden des Signales
```

```
def get_signal_cut(signal, fs, t1, t2):
    signal_cut = signal.iloc[int(round(t1*fs)):int(round(t2*fs))]
    return(signal_cut)
```

```
#Plotten der Zeitreihendaten
```

```
def timeplot_cut(signal, signal_cut, savename, fs, rpm, adoc,
                 stability, t0, t1, t2, t3):
    fig = plt.figure(figsize = (12,6), dpi=300, facecolor='w')
    xplot = (np.arange(0, len(signal))/fs)
    yplot = signal
    xplot_cut = (np.arange(0, len(signal_cut))/fs)+t1
    yplot_cut = signal_cut

    plt.plot(xplot, yplot, "k-", linewidth=0.001,
             label="Sensorsignal")
    plt.plot(xplot_cut, yplot_cut, "g-", linewidth=0.01,
             label="Sensorsignal im Eingriff")

    plt.plot([t0, t0], [min(signal), max(signal)], "r-", linewidth=0.9,
             label="erster Kontakt")
    plt.plot([t1, t1], [min(signal), max(signal)], "b-", linewidth=0.9,
```

```

        label="erste volle Schnittbreite")
plt.plot([t2, t2], [min(signal), max(signal)], "r-", linewidth=0.9,
        label="letzte volle Schnittbreite")
plt.plot([t3, t3], [min(signal), max(signal)], "b-", linewidth=0.9,
        label="letzter Kontakt")

plt.xlim(xmin=0, xmax=np.max(xplot))
plt.ylim(ymin=26000, ymax=42000)
plt.legend()
plt.xlabel("Zeit [s]", fontsize=10)
plt.ylabel("Sensorsignal in bits [-]", fontsize=10)
plt.grid(True)
fig.savefig("Zeitdaten_" + str(savename) + "_cut.pdf",
        bbox_inches='tight', dpi=fig.dpi)
plt.close()

```

*#Berechnung und Plotten des Spektrums*

```

def spectrumplot(signal, fs, rpm, a_lim, filename, stability):
    signal_norm = signal - np.mean(np.array(signal))
    N = len(signal_norm)
    T = 1/fs

    x = np.linspace(0.0, N*T, N)
    xf = np.linspace(0.0, 1.0/(2.0*T), N/2)
    xf = xf/(rpm/60)

    y = signal_norm
    yf = fftpack.fft(y)
    yf = 2.0/N * np.abs(yf[:N//2]) -5 #beseitigen des Messrauschens
    yf[yf < 0] = 0

    highestint = int((fs/2)/(rpm/60))

    list_x_oddinteger = list()
    list_y_oddinteger = list()
    list_x_interspace = list()
    list_y_interspace = list()

#loop für Berechnung der ungeraden Vielfachen im fft
    for i in range(1, highestint, 2):
        #Bereich von ungeraden Vielfachen

```

```

section_oddinteger = np.argwhere((xf>(i-0.10)) & (xf<i+0.10))
#Suchen des größten Y-Wertes in oberen Bereich
y_max = np.amax(yf[section_oddinteger])
#Suchen des Index des größten Y-Wertes
y_max_index = np.argmax(yf[section_oddinteger])
#Speichern des zum größten Y-Wert dazugehörigen x-Wert
x_max = xf[section_oddinteger[y_max_index]]
list_y_oddinteger.append(y_max)
list_x_oddinteger.append(x_max)
#if i % 2 == 0: #Speichern der gerade(even) Punkte
#if i % 2 == 1: #Speichern der ungeraden (odd) Punkte

#loop für Berechnung der Spitzenwerte zwischen ungeraden Vielfachen im fft
for i in range(1, highestint, 2):
    #Bereich zwischen den ungeraden Vielfachen
    section_interspace = np.argwhere((xf>(i+0.10)) & (xf<(i+2-0.10)))
    y_max = np.amax(yf[section_interspace])
    #Suchen des Index des größten Y-Wertes
    y_max_index = np.argmax(yf[section_interspace])
    #Speichern des zum größten Y-Wert dazugehörigen x-Wert
    x_max = xf[section_interspace[y_max_index]]
    list_y_interspace.append(y_max)
    list_x_interspace.append(x_max)

    #Ploten des Spektrums
fig = plt.figure(figsize = (12,6), dpi=300, facecolor='w')
plt.plot(xf, yf, "g-", linewidth=0.8, label="Spektrum")
plt.plot(list_x_oddinteger, list_y_oddinteger, 'rX', markersize=4,
         label="max. oddinteger points")
plt.plot(list_x_interspace, list_y_interspace, 'bX', markersize=4,
         label="max. intersection points")
plt.xlim(xmin=0, xmax=np.max(xf))
plt.xticks(np.arange(0, int(np.max(xf))+1, 1))
plt.legend()
plt.xlabel("Ganzzahlige Vielfache der Umdrehungsfrequenz", fontsize=10)
plt.ylabel("Amplitude", fontsize=10)
plt.grid(True)
fig.savefig("intSpektrum" + str(filename) + ".pdf",
          bbox_inches='tight', dpi=fig.dpi)
plt.close()

```

```

#Berechnung der Features
def myfeatures(inputsignal, samplingrate, rpm):
    signal_norm = signal - np.mean(np.array(signal))
    N = len(signal_norm)
    T = 1/fs

    x = np.linspace(0.0, N*T, N)
    xf = np.linspace(0.0, 1.0/(2.0*T), N/2)
    xf = xf/(rpm/60)

    y = signal_norm
    yf = fftpack.fft(y)
    yf = 2.0/N * np.abs(yf[:N//2]) -5 #beseitigen des Messrauschens
    yf[yf < 0] = 0

    highestint = int((fs/2)/(rpm/60))

    list_x_oddinteger = list()
    list_y_oddinteger = list()
    list_x_interspace = list()
    list_y_interspace = list()

    #loop für Berechnung der ungeraden Vielfachen im fft
    for i in range(1, highestint, 2):
        #Bereich von ungeraden Vielfachen
        section_oddinteger = np.argwhere((xf>(i-0.10)) & (xf<i+0.10))
        #Suchen des größten Y-Wertes in oberen Bereich
        y_max = np.amax(yf[section_oddinteger])
        #Suchen des Index des größten Y-Wertes
        y_max_index = np.argmax(yf[section_oddinteger])
        #Speichern des zum größten Y-Wert dazugehörigen x-Wert
        x_max = xf[section_oddinteger[y_max_index]]
        list_y_oddinteger.append(y_max)
        list_x_oddinteger.append(x_max)
        #if i % 2 == 0: #Speichern der gerade(even) Punkte
        #if i % 2 == 1: #Speichern der ungeraden (odd) Punkte

    #loop für Berechnung der max. Werte zwischen ungeraden Vielfachen
    for i in range(1, highestint, 2):
        #Bereich zwischen den ungeraden Vielfachen
        section_interspace = np.argwhere((xf>(i+0.10)) & (xf<(i+2-0.10)))
        y_max = np.amax(yf[section_interspace])

```

```

    #Suchen des Index des größten Y-Wertes
    y_max_index = np.argmax(yf[section_interspace])
    #Speichern des zum größten Y-Wert dazugehörigen x-Wert
    x_max = xf[section_interspace[y_max_index]]
    list_y_interspace.append(y_max)
    list_x_interspace.append(x_max)

list_y_oddinteger = np.asarray(list_y_oddinteger)
list_y_interspace = np.asarray(list_y_interspace)

feature0 = np.sum(list_y_interspace)/np.sum(list_y_oddinteger)
feature1 = np.max(list_y_interspace)/np.max(list_y_oddinteger)
feature2_Z = np.sum(list_y_interspace[np.argsort(list_y_interspace)[-2:]])
feature2_N = np.sum(list_y_oddinteger[np.argsort(list_y_oddinteger)[-2:]])
feature2 = feature2_Z/feature2_N
feature3_Z = np.sum(list_y_interspace[np.argsort(list_y_interspace)[-3:]])
feature3_N = np.sum(list_y_oddinteger[np.argsort(list_y_oddinteger)[-3:]])
feature3 = feature3_Z/feature3_N
feature4_Z = np.sum(list_y_interspace[np.argsort(list_y_interspace)[-2:]])
feature4_N = np.sum(list_y_oddinteger[np.argsort(list_y_oddinteger)[-1:]])
feature4 = feature4_Z/feature4_N
feature5 = np.sum(yf)/np.sum(list_y_oddinteger)
feature6 = (np.sum(yf)-np.sum(list_y_oddinteger))/np.sum(list_y_oddinteger)
feature7 = np.mean(list_y_interspace)/np.mean(list_y_oddinteger)
feature8 = np.mean(yf)/np.mean(list_y_oddinteger)
return([feature0, feature1, feature2, feature3, feature4,
        feature5, feature6, feature7, feature8])

```

## A.4 Datenauswertung: Zeitreihenplots und Spektrumplots

```
# -*- coding: utf-8 -*-

"""
Datenauswertung: Plotten der Graphen      Oswald 2019
"""

#-----
#Datum:          24.07.2019 (Wednesday)
#Uhrzeit:        23:30:00
#Bearbeiter:     Ralf Oswald
#Projekt:        Diplomarbeit_Oswald
#Datei:          Datenauswertung.py

#Importieren der Bibliotheken
import numpy as np
import pandas as pd
from OwnFeatures import get_signal_old
from OwnFeatures import get_signal_new
from OwnFeatures import timevalues
from OwnFeatures import get_signal_cut
from OwnFeatures import timeplot
from OwnFeatures import timeplot_cut
from OwnFeatures import spectrumpLOT

# Einlesen der Versuchsdaten
data = pd.read_csv("data.csv", sep=";")

#Berechnung der Zeiten (nur einmal notwendig - bzw. wird überschrieben)
times = []
for i in range(0, data.shape[0]):
    signal = get_signal(folder=folder, filename=str(data["testrun"][i]),
                       extension="txt")
    times.append(timevalues(signal=signal["accx"], fs=data["fs"][i],
                           factor=1.03, rpm=data["rpm"][i]))
    print(str(round(100*i/data.shape[0]))+"%") #Fortschrittsanzeige
print("100%")
times = np.transpose(np.asarray(times))

data['t0']=times[0]
data['t1']=times[1]
```

```

data['t2']=times[2]
data['t3']=times[3]

data.to_csv("data.csv", sep=';', index=False)

#Zu plottende Graphen auswählen (True/False)
ask_timeplot = False
ask_timeplot_cut = True
ask_spectrumplot = True

#Auswahl der zu plottenden Versuchsreihe
index = int(data.loc[data['testrun']==2020].index.values)

#Einlesen der Rohdaten
signal = get_signal(folder=folder, filename=str(data["testrun"][index]),
                    extension="txt")

#Zeitreihen plotten
if ask_timeplot == False:
    timeplot(signal["accx"], str(data["testrun"][index])+"x",
             data["fs"][index], data["rpm"][index], data["adoc"][index],
             data["stability"][index], data["t0"][index],
             data["t1"][index], data["t2"][index], data["t3"][index])
    timeplot(signal["accy"], str(data["testrun"][index])+"y",
             data["fs"][index], data["rpm"][index], data["adoc"][index],
             data["stability"][index], data["t0"][index],
             data["t1"][index], data["t2"][index], data["t3"][index])
    timeplot(signal["accz"], str(data["testrun"][index])+"z",
             data["fs"][index], data["rpm"][index], data["adoc"][index],
             data["stability"][index], data["t0"][index],
             data["t1"][index], data["t2"][index], data["t3"][index])

#Geschnittene Zeitreihen plotten
signal_cut = get_signal_cut(signal, data["fs"][index],
                             data["t1"][index], data["t2"][index])
if ask_timeplot_cut == True:
    timeplot_cut(signal["accx"], signal_cut["accx"],
                 str(data["testrun"][index])+"x", data["fs"][index],
                 data["rpm"][index], data["adoc"][index],
                 data["stability"][index], data["t0"][index],
                 data["t1"][index], data["t2"][index], data["t3"][index])
    timeplot_cut(signal["accy"], signal_cut["accy"],
                 str(data["testrun"][index])+"y", data["fs"][index],

```

```
data["rpm"][index], data["adoc"][index],
data["stability"][index], data["t0"][index],
data["t1"][index], data["t2"][index], data["t3"][index])
timeplot_cut(signal["accz"], signal_cut["accz"],
str(data["testrun"][index])+"z", data["fs"][index],
data["rpm"][index], data["adoc"][index],
data["stability"][index], data["t0"][index],
data["t1"][index], data["t2"][index], data["t3"][index])
#Spektrum plotten
if ask_spectrumplot == True:
    spectrumplot(signal_cut["accx"], data["fs"][index], data["rpm"][index],
data["adoc"][index], str(data["testrun"][index])+"x",
data["stability"][index])
    spectrumplot(signal_cut["accy"], data["fs"][index], data["rpm"][index],
data["adoc"][index], str(data["testrun"][index])+"y",
data["stability"][index])
    spectrumplot(signal_cut["accz"], data["fs"][index], data["rpm"][index],
data["adoc"][index], str(data["testrun"][index])+"z",
data["stability"][index])
```

## A.5 Feature Extraction

```
# -*- coding: utf-8 -*-

"""
Erstellen und Abspeichern der Features      Oswald 2019
"""

#-----
#Datum:          24.07.2019 (Wednesday)
#Uhrzeit:        23:30:00
#Bearbeiter:     Ralf Oswald
#Projekt:        Diplomarbeit_Oswald
#Datei:          Feature_Extraction.py

#Importieren der Bibliotheken
import numpy as np
import pandas as pd
from scipy.stats import kurtosis
from scipy.stats import skew
from OwnFeatures import myfeatures
from OwnFeatures import get_signal_old
from OwnFeatures import get_signal_new
from OwnFeatures import timevalues
from OwnFeatures import get_signal_cut

feature_names = ["feature_mean", "feature_median", "feature_variance",
                 "feature_q75", "feature_q25", "feature_iqr",
                 "feature_kurtosis", "feature_skewness", "feature_shannon",
                 "feature_0", "feature_1", "feature_2", "feature_3", "feature_4",
                 "feature_5", "feature_6", "feature_7", "feature_8"]

#Auswahl der zu berechnenden Datenreihen
index_start = int(data_new.loc[data_new['testrun']==1840].index.values)
index_end = int(data_new.loc[data_new['testrun']==2200].index.values)

features = pd.DataFrame(columns=(feature_names),
                        index=range(index_start, index_end+1))

for i in range(index_start, index_end+1):
    #Einlesen des Signals
    signal = get_signal(folder=folder, filename=str(data["testrun"][i]),
```

```

        extension="txt")
    signal_cut = get_signal_cut(signal, data["fs"][i],
                               data["t1"][i], data["t2"][i])
    #Berechnen der Features im Zeitbereich
    features["feature_mean"][i] = np.mean(signal_cut["accx"])
    features["feature_median"][i] = np.median(signal_cut["accx"])
    features["feature_variance"][i] = np.var(signal_cut["accx"])
    features["feature_q75"][i] = np.quantile(signal_cut["accx"], 0.75)
    features["feature_q25"][i] = np.quantile(signal_cut["accx"], 0.25)
    features["feature_iqr"][i] = np.quantile(signal_cut["accx"], 0.75)
    -np.quantile(signal_cut["accx"], 0.25)
    features["feature_kurtosis"][i] = kurtosis(signal_cut["accx"])
    features["feature_skewness"][i] = skew(signal_cut["accx"])*(-1)
    features["feature_shannon"][i] = shannon(signal_cut["accx"])

    features["feature_0"][i] = myfeatures(signal_cut["accx"],
                                         data["fs"][i], data["rpm"][i])[0]
    features["feature_1"][i] = myfeatures(signal_cut["accx"],
                                         data["fs"][i], data["rpm"][i])[1]
    features["feature_2"][i] = myfeatures(signal_cut["accx"],
                                         data["fs"][i], data["rpm"][i])[2]
    features["feature_3"][i] = myfeatures(signal_cut["accx"],
                                         data["fs"][i], data["rpm"][i])[3]
    features["feature_4"][i] = myfeatures(signal_cut["accx"],
                                         data["fs"][i], data["rpm"][i])[4]
    features["feature_5"][i] = myfeatures(signal_cut["accx"],
                                         data["fs"][i], data["rpm"][i])[5]
    features["feature_6"][i] = myfeatures(signal_cut["accx"],
                                         data["fs"][i], data["rpm"][i])[6]
    features["feature_7"][i] = myfeatures(signal_cut["accx"],
                                         data["fs"][i], data["rpm"][i])[7]
    features["feature_8"][i] = myfeatures(signal_cut["accx"],
                                         data["fs"][i], data["rpm"][i])[8]
    print(str(round(100*i/index_end)+"%") #Fortschrittsanzeige)
    print("100%")

    features.insert(0, "testrun", data["testrun"])
    features.insert(1, "rpm", data["rpm"])

    features.to_csv("features.csv", sep=';', index=False)

```

## A.6 Feature Selection

```
# -*- coding: utf-8 -*-

"""
Auswahl der Features      Oswald 2019
"""

#-----
#Datum:          24.07.2019 (Wednesday)
#Uhrzeit:        23:30:00
#Bearbeiter:     Ralf Oswald
#Projekt:        Diplomarbeit_Oswald
#Datei:          Feature_Selection.py

#Importieren der Bibliotheken
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from sklearn.datasets import make_classification
from sklearn.ensemble import ExtraTreesClassifier

from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.feature_selection import SelectFpr, chi2
from sklearn.feature_selection import VarianceThreshold

#Einlesen der relevanten Daten
features = ["feature_mean", "feature_median", "feature_variance",
           "feature_q75", "feature_q25", "feature_iqr",
           "feature_kurtosis", "feature_skewness", "feature_shannon",
           "feature_0", "feature_1", "feature_2", "feature_3", "feature_4",
           "feature_5", "feature_6", "feature_7", "feature_8"]

X = pd.read_csv("features.csv", sep=";", usecols=features)
y = pd.read_csv("data.csv", sep=";", usecols=["stability"])

#Geringe Varianz-Kriterium
df = pd.DataFrame(X, columns=features)
```

```

scaler = MinMaxScaler()
MinMaxScaler(copy=True, feature_range=(0, 1))
scaler.fit(X)
X_new = scaler.transform(X)
p = 0.95
feature_selector = VarianceThreshold(threshold=(p*(1-p)))
X_new = feature_selector.fit_transform(X_new)
bestfeatures = df.columns[feature_selector.get_support()]
for i in range(len(bestfeatures)):
    print(str(i+1) + ". " + str(bestfeatures[i]))

#Hohe Korrelations-Kriterium
df = pd.DataFrame(X, columns=features)
feature_selector = SelectKBest(chi2, k=8)
scaler = MinMaxScaler()
MinMaxScaler(copy=True, feature_range=(0, 1))
scaler.fit(X)
X_new = scaler.transform(X)
X_new = feature_selector.fit_transform(X_new, y)
X_new.shape
bestfeatures = df.columns[feature_selector.get_support()]
for i in range(len(bestfeatures)):
    print(str(i+1) + ". " + str(bestfeatures[i]))

#Random Forest Verfahren
forest = ExtraTreesClassifier(n_estimators=250,
                              random_state=0)

forest.fit(X, y)
importances = forest.feature_importances_
importances = 100*importances/np.sum(importances)
std = np.std([tree.feature_importances_ for tree in forest.estimators_],
             axis=0)
indices = np.argsort(importances)[::-1]

print("Feature ranking:")
features_sorted = list()
for i in range(X.shape[1]):
    features_sorted.append(features[indices[i]])
    print(features[indices[i]], importances[indices[i]])

```

```
#False-Positiv-Kriterium
df = pd.DataFrame(X, columns=features)
feature_selector = SelectFpr(chi2, alpha=0.001)
scaler = MinMaxScaler()
MinMaxScaler(copy=True, feature_range=(0, 1))
scaler.fit(X)
X_new = scaler.transform(X)
X_new = feature_selector.fit_transform(X_new, y)
X_new.shape
bestfeatures = df.columns[feature_selector.get_support()]
for i in range(len(bestfeatures)):
    print(str(i+1) + ". " + str(bestfeatures[i]))
```

## A.7 Erstellen des Neuronalen Netzwerks

```
# -*- coding: utf-8 -*-

"""
Erstellen des Neuronalen Netzwerks      Oswald 2019
"""

#-----
#Datum:          24.07.2019 (Wednesday)
#Uhrzeit:        23:30:00
#Bearbeiter:     Ralf Oswald
#Projekt:        Diplomarbeit_Oswald
#Datei:          NN_sklearn.py

#Importieren der Bibliotheken
import pandas as pd

from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

#Einlesen der Daten
testrun = pd.read_csv("data.csv", sep=";", usecols=["testrun", "stability"])
input_data = pd.read_csv("features.csv", sep=";",
                        usecols=["feature_variance", "feature_iqr",
                                "feature_1", "feature_2", "feature_3"])
output_data = pd.read_csv("data.csv", sep=";", usecols=["stability"])

for i in range(0, output_data.shape[0]):
    if output_data["stability"][i] == "semistabil":
        output_data["stability"][i] = "instabil"

#Splitten der Daten in Test- und Trainingsdaten
input_train, input_test, output_train, output_test = train_test_split(
    input_data, output_data, test_size= 0.50, random_state=42)

#Normieren der Daten
scaler = StandardScaler()
scaler.fit(input_train)
input_train = scaler.transform(input_train)
input_test = scaler.transform(input_test)
```

```
#Erstellen des Neuronalen Netzwerks
NN = MLPClassifier(hidden_layer_sizes=(100,100), max_iter=1000, alpha=0.0001,
                   solver='sgd', verbose=True, random_state=1,tol=0.00000001)

#Trainieren
NN.fit(input_train, output_train)

#Testen
y_pred = NN.predict(input_test)

#Printen der Genauigkeit des Modells
print(accuracy_score(output_test, y_pred))

#Ausgabe der falsch prädiktierten Versuchsreihen
index_output_test = output_test.index.tolist()
output_pred = pd.DataFrame(y_pred, columns = ["stability"],
                           index = index_output_test)

for i in index_output_test:
    if output_test["stability"][i] == output_pred["stability"][i]:
        None
    else:
        print(testrun["testrun"][i], " ", testrun["stability"][i])
```

## A.8 Nutzen des Neuronalen Netzwerks

```
# -*- coding: utf-8 -*-

"""
Speichern, Laden und Verwenden des Neuronalen Netzwerks  Oswald 2019
"""

#-----
#Datum:          24.07.2019 (Wednesday)
#Uhrzeit:        22:43:00
#Bearbeiter:     Ralf Oswald
#Projekt:        Diplomarbeit_Oswald
#Datei:          using_NN.py

#Importieren der Bibliotheken
import numpy as np
from sklearn.externals import joblib
import pickle

#Abspeichern des Neuronalen Netzwerks
joblib.dump(myNN, 'myNN.pkl')

#Laden des Neuronalen Netzwerks
NN_from_joblib = joblib.load('myNN.pkl')

#Nutzen des Neuronalen Netzwerks
input_layer = [-0.205576, 0.25966, -1.01365, 0.461212, 0.540874]
test = np.array(input_layer)
prediction = str(NN_from_joblib.predict(test))
print(prediction)
```



## Anhang B

# Versuchsprotokoll

Testlauf	X	Z	a <sub>e</sub>	a <sub>p</sub>	N	n	v <sub>F</sub>	f <sub>z</sub>	Kühlung	Achsen	f <sub>s</sub>	Stabilität
[-]	[mm]	[mm]	[mm]	[mm]	[-]	[U/min]	[mm/min]	[mm/Z.]	[-]	[-]	[Hz]	[-]
<b>Impulshammer, Fräser vermessen an allen 4 Schneiden</b>												
<b>1. Versuchstag, am 26.03.2019</b>												
0010	10	-1.5	10	1.5	4	1650	660	0.1000	NEIN	XYZ+stat	1357	stabil
0011	20	-1.5	10	1.5	4	1650	660	0.1000	NEIN	XYZ+stat	1357	stabil
0020	20	-4	20	2.5	4	1650	660	0.1000	JA	XYZ	1357	stabil
0030	20	-6.5	20	2.5	4	1650	660	0.1000	JA	X	4070	stabil
0040	20	-9	20	2.5	4	1650	660	0.1000	JA	X	4070	stabil
0050	40	-1.5	20	1.5	4	1650	1320	0.2000	JA	X	4070	stabil
0060	40	-4	20	2.5	4	1650	1320	0.2	JA	X	4070	stabil
0070	40	-6.5	20	2.5	4	1650	1320	0.2	JA	X	4070	stabil
0080	40	-9	20	2.5	4	1650	1320	0.2	JA	X	4070	stabil
0090	42	-9	2	9	4	1650	1320	0.2	JA	X	4070	stabil
0100	44	-9	2	9	4	1650	1320	0.2	JA	X	4070	stabil
0110	46	-9	2	9	4	1650	1320	0.2	JA	X	4070	stabil
0120	48	-9	2	9	4	1650	1320	0.2	JA	XYZ	1357	stabil
0130	50	-9	2	9	4	1650	1320	0.2	JA	XYZ	1357	stabil
<b>2. Versuchstag, am 27.03.2019</b>												
0140	52	-9	2	9	4	1650	1980	0.3	JA	X	4070	stabil
0150	54	-9	2	9	4	1650	1980	0.3	JA	X	4070	stabil
0160	56	-9	2	9	4	1650	1980	0.3	JA	X	4070	stabil
0170	58	-9	2	9	4	1650	1980	0.3	JA	Y	4070	stabil
0180	60	-9	2	9	4	1650	1980	0.3	JA	Y	4070	stabil
0190	62	-9	2	9	4	1650	1980	0.3	JA	Y	4070	stabil
0200	64	-9	2	9	4	1650	1980	0.3	JA	Z	4070	stabil
0210	66	-9	2	9	4	1650	1980	0.3	JA	Z	4070	stabil
0220	68	-9	2	9	4	1650	1980	0.3	JA	Z	4070	stabil
0230	70	-9	2	9	4	1650	1980	0.3	JA	XYZ	1357	stabil
0240	72	-9	2	9	4	1650	1980	0.3	JA	XYZ	1357	stabil
0250	74	-9	2	9	4	1650	1980	0.3	JA	XYZ	1357	stabil
0260	76	-9	2	9	4	1650	1980	0.3	JA	XY	2035	stabil
0270	78	-9	2	9	4	1650	1980	0.3	JA	XY	2035	stabil
0280	80	-9	2	9	4	1650	1980	0.3	JA	XY	2035	stabil
0290	82	-9	2	9	4	2475	1980	0.2	JA	X	4070	stabil
0300	84	-9	2	9	4	2475	1980	0.2	JA	X	4070	stabil
0310	86	-9	2	9	4	2475	1980	0.2	JA	X	4070	stabil
0320	88	-9	2	9	4	2475	1980	0.2	JA	X	4070	stabil
0330	90	-9	2	9	4	2475	1980	0.2	JA	X	4070	stabil
0340	92	-9	2	9	4	2475	1980	0.2	JA	X	4070	stabil
0350	94	-9	2	9	4	2475	1980	0.2	JA	X	4070	stabil
0360	96	-9	2	9	4	2475	1980	0.2	JA	X	4070	stabil
0370	98	-9	2	9	4	2475	1980	0.2	JA	X	4070	stabil
0380	100.5	-9	2	9	4	2475	1980	0.2	JA	X	4070	stabil
0390	2	-18	2	9	4	3300	2640	0.2	JA	X	4070	stabil
0400	4	-18	2	9	4	3300	2640	0.2	JA	X	4070	stabil
0410	6	-18	2	9	4	3300	2640	0.2	JA	X	4070	stabil
0420	8	-18	2	9	4	4125	3300	0.2	JA	X	4070	stabil
0430	10	-18	2	9	4	4125	3300	0.2	JA	X	4070	stabil
0440	12	-18	2	9	4	4125	3300	0.2	JA	X	4070	stabil

0450	14	-18	2	9	4	4950	3960	0.2	JA	X	4070	stabil
0460	16	-18	2	9	4	4950	3960	0.2	JA	X	4070	stabil
0470	18	-18	2	9	4	4950	3960	0.2	JA	X	4070	stabil
0480	20	-18	2	9	4	5775	4620	0.2	JA	X	4070	stabil
0490	22	-18	2	9	4	5775	4620	0.2	JA	X	4070	stabil
0500	24	-18	2	9	4	5775	4620	0.2	JA	X	4070	stabil
0510	28	-18	4	9	4	1650	1320	0.2	JA	X	4070	stabil
0520	32	-18	4	9	4	1650	1320	0.2	JA	X	4070	stabil
0530	36	-18	4	9	4	1650	1320	0.2	JA	X	4070	stabil
0540	40	-18	4	9	4	2475	1980	0.2	JA	X	4070	stabil
0550	44	-18	4	9	4	2475	1980	0.2	JA	X	4070	stabil
0560	48	-18	4	9	4	2475	1980	0.2	JA	X	4070	stabil
0570	52	-18	4	9	4	3300	2640	0.2	JA	X	4070	stabil
0580	56	-18	4	9	4	3300	2640	0.2	JA	X	4070	stabil
0590	60	-18	4	9	4	3300	2640	0.2	JA	X	4070	stabil
0600	64	-18	4	9	4	4125	3300	0.2	JA	X	4070	stabil
0610	68	-18	4	9	4	4125	3300	0.2	JA	X	4070	stabil
0620	72	-18	4	9	4	4125	3300	0.2	JA	X	4070	stabil
0630	76	-18	4	9	4	4950	3960	0.2	JA	X	4070	stabil
0640	80	-18	4	9	4	4950	3960	0.2	JA	X	4070	stabil
0650	84	-18	4	9	4	5775	4620	0.2	JA	X	4070	stabil
0660	88	-18	4	9	4	5775	4620	0.2	JA	X	4070	stabil
0670	92	-18	4	9	4	5775	4620	0.2	JA	X	4070	stabil
0680	96	-18	4	9	4	5775	4620	0.2	JA	X	4070	stabil
0690	100.7	-18	4	9	4	5775	4620	0.2	JA	X	4070	stabil
0700	20	-19	20	1	4	4640	1856	0.1	JA	X	4070	stabil
0710	20	-21	20	1.5	4	4640	1856	0.1	JA	X	4070	stabil
0720	20	-23	20	2	4	4640	1856	0.1	JA	X	4070	stabil
0730	20	-25	20	2.5	4	4640	1856	0.1	JA	X	4070	stabil
0740	40	-21	20	3	4	4640	1856	0.1	JA	X	4070	stabil
0750	40	-25	20	4	4	4640	1856	0.1	JA	X	4070	stabil
0760	60	-23	20	5	4	4640	1856	0.1	JA	X	4070	instabil
0770	60	-25	20	2	4	4640	1856	0.1	JA	X	4070	stabil
0780	80	-23	20	4.5	4	4640	1856	0.1	JA	X	4070	instabil
0790	80	-25	20	2.5	4	4640	1856	0.1	JA	X	4070	stabil
0800	100.5	-23	20	5	4	4640	1856	0.1	JA	X	4070	instabil
0810	100.5	-25	20	2	4	4640	1856	0.1	JA	X	4070	stabil
0820	20	-30	20	5	4	4140	1656	0.1	JA	X	4070	instabil
0830	20	-35	20	4.5	4	4140	1656	0.1	JA	X	4070	semistabil
0840	40	-24	20	4	4	4140	1656	0.1	JA	X	4070	stabil
0850	40	-33	20	4	4	4390	1756	0.1	JA	X	4070	instabil
0860	40	-35	20	1.5	4	4390	1756	0.1	JA	X	4070	stabil
0870	60	-30	20	4.5	4	4390	1756	0.1	JA	X	4070	instabil
0880	60	-33	20	3.5	4	4390	1756	0.1	JA	X	4070	stabil
0890	60	-35	20	1.5	4	4390	1756	0.1	JA	X	4070	stabil
0900	80	-30	20	4.5	4	4890	1956	0.1	JA	X	4070	stabil
0910	80	-35	20	5	4	4890	1956	0.1	JA	X	4070	stabil
0920	101	-31	20	5.5	4	4890	1956	0.1	JA	X	4070	semistabil
0930	101	-35	20	4	4	4890	1956	0.1	JA	X	4070	stabil
0940	20	-39	20	4	4	5140	2056	0.1	JA	X	4070	stabil

0950	20	-44	20	5	4	5140	2056	0.1	JA	X	4070	stabil
0960	40	-39	20	4	4	5390	2156	0.1	JA	X	4070	stabil
0970	40	-44	20	5	4	5390	2156	0.1	JA	X	4070	stabil
0980	60	-40	20	5.5	4	5390	2156	0.1	JA	X	4070	stabil
0990	60	-44	20	3.5	4	5390	2156	0.1	JA	X	4070	stabil
1000	80	-39	20	4	4	5640	2256	0.1	JA	X	4070	stabil
1010	80	-44	20	5	4	5640	2256	0.1	JA	X	4070	instabil
1020	101	-39	20	4.5	4	5640	2256	0.1	JA	X	4070	stabil
1030	101	-44	20	4.5	4	5640	2256	0.1	JA	X	4070	stabil
1040	20	-50	20	6	4	5140	2056	0.1	JA	X	4070	semistabil
1050	20	-53	20	3	4	5140	2056	0.1	JA	X	4070	stabil
1060	40	-50	20	6.5	4	5140	2056	0.1	JA	X	4070	semistabil
1070	40	-53	20	2.5	4	5140	2056	0.1	JA	X	4070	stabil
1080	60	-49	20	5.5	4	5140	2056	0.1	JA	X	4070	semistabil
1090	60	-53	20	3.5	4	4390	1756	0.1	JA	X	4070	stabil
1100	80	-48	20	4	4	4390	1756	0.1	JA	X	4070	instabil
1110	80	-53	20	5	4	4390	1756	0.1	JA	X	4070	instabil
1120	101	-48	20	4	4	3890	1556	0.1	JA	X	4070	stabil
1130	101	-53	20	5	4	3890	1556	0.1	JA	X	4070	stabil
1140	20	-58	20	5.5	4	3890	1556	0.1	JA	X	4070	stabil
1150	20	-62	20	3.5	4	3890	1556	0.1	JA	X	4070	stabil
1160	40	-59	20	6.5	4	3890	1556	0.1	JA	X	4070	instabil
1170	60	-59	20	6	4	3890	1556	0.1	JA	X	4070	instabil
1180	80	-59	20	6	4	4890	1956	0.1	JA	X	4070	stabil
1190	101	-59	20	6	4	5390	2156	0.1	JA	X	4070	stabil
1200	20	-68	20	6.5	4	5390	2156	0.1	JA	X	4070	stabil
1210	40	-66	20	7	4	5390	2156	0.1	JA	X	4070	instabil
1220	60	-63	20	4	4	5890	2356	0.1	JA	X	4070	semistabil
1230	60	-66	20	3.5	4	5890	2356	0.1	JA	X	4070	stabil
1240	80	-63	20	4.5	4	5890	2356	0.1	JA	X	4070	instabil
1250	80	-66	20	3	4	5890	2356	0.1	JA	X	4070	stabil
1260	101	-64	20	5	4	5890	2356	0.1	JA	X	4070	instabil
1270	101	-66	20	2.5	4	5890	2356	0.1	JA	X	4070	stabil
1280	20	-74	20	5.5	4	5890	2356	0.1	JA	X	4070	instabil
1290	40	-73	20	6.5	4	4890	1956	0.1	JA	X	4070	semistabil
1300	40	-74	20	1	4	4890	1956	0.1	JA	X	4070	stabil
1310	60	-73	20	7	4	4890	1956	0.1	JA	X	4070	semistabil
3. Versuchstag, am 27.05.2019 (ERROR: zu niedrige Abtastrate)												
1410	40	7	20	2	4	4640	1856	0.1	JA	XYZ		stabil
1420	40	5	20	2	4	4640	1856	0.1	JA	XYZ		stabil
1430	40	3	20	2	4	4640	1856	0.1	JA	XYZ		stabil
1440	80	14.5	20	2	4	4640	1856	0.1	JA	XYZ		stabil
1450	80	12.5	20	2	4	4640	1856	0.1	JA	XYZ		stabil
1460	80	10.5	20	2	4	4640	1856	0.1	JA	XYZ		stabil
1470	100	14.5	20	2	4	4640	1856	0.1	JA	XYZ		stabil
1480	100	12.5	20	2	4	4640	1856	0.1	JA	XYZ	870	stabil
1490	100	10.5	20	2	4	4640	1856	0.1	JA	XYZ	870	stabil
1500	60	7	20	2.5	4	4640	1856	0.1	JA	XYZ	870	stabil
1510	80	7	20	3.5	4	4640	1856	0.1	JA	XYZ	870	stabil
1520	100	7	20	3.5	4	4640	1856	0.1	JA	XYZ	870	stabil

1530	60	3	20	4	4	5640	2256	0.1	JA	XYZ	870	stabil
1540	80	3	20	4	4	5890	2356	0.1	JA	XYZ	870	instabil
1550	100	3	20	4	4	5890	2356	0.1	JA	XYZ	870	stabil
1560	20	-1	20	4	4	6140	2456	0.1	JA	XYZ	870	
1570	40	-1	20	4	4	6140	2456	0.1	JA	XYZ	870	stabil
1580	60	-1	20	4	4	6390	2556	0.1	JA	XYZ	870	stabil
1590	80	-1	20	4	4	4390	1756	0.1	JA	XYZ	870	stabil
1600	100	-1	20	4	4	4390	1756	0.1	JA	XYZ	870	stabil
1610	20	-6	20	5	4	4890	1956	0.1	JA	XYZ	870	stabil
1620	40	-6	20	5	4	4640	1856	0.1	JA	XYZ	870	semistabil
1630	60	-6	20	5	4	4390	1756	0.1	JA	XYZ	870	instabil
1640	80	-6	20	5	4	4140	1656	0.1	JA	XYZ	870	instabil
1650	100	-6	20	5	4	4140	1656	0.1	JA	XYZ	870	stabil
1660	20	-11	20	5	4	5140	2056	0.1	JA	XYZ	870	semistabil
1670	40	-11	20	5	4	5390	2156	0.1	JA	XYZ	870	stabil
1680	60	-11	20	5	4	5640	2256	0.1	JA	XYZ	870	instabil
1690	80	-11	20	5	4	5890	2356	0.1	JA	XYZ	870	instabil
1700	100	-11	20	5	4	5890	2356	0.1	JA	XYZ	870	stabil
1710	20	-16	20	5	4	6890	2756	0.1	JA	XYZ	870	stabil
1720	40	-16	20	5	4	7140	2856	0.1	JA	XYZ	870	stabil
1730	60	-16	20	5	4	6640	2656	0.1	JA	XYZ	870	instabil
1740	80	-16	20	5	4	6390	2556	0.1	JA	XYZ	870	stabil
1750	100	-16	20	5	4	6390	2556	0.1	JA	XYZ	870	stabil
1760	20	-21	20	5	4	8390	3356	0.1	JA	XYZ	870	stabil
1770	40	-21	20	5	4	8640	3456	0.1	JA	XYZ	870	stabil
1780	60	-21	20	5	4	8890	3556	0.1	JA	XYZ	870	stabil
1790	80	-21	20	5	4	9140	3656	0.1	JA	XYZ	870	instabil
1800	100	-21	20	5	4	9140	3656	0.1	JA	XYZ	870	stabil
1810	20	-27	20	6	4	4890	1956	0.1	JA	XYZ	870	stabil
1820	40	-27	20	6	4	4640	1856	0.1	JA	XYZ	870	instabil
1830	60	-27	20	6	4	4390	1756	0.1	JA	XYZ	870	instabil
4. Versuchstag, am 28.05.2019												
1840	80	-27	20	6	4	4890	1956	0.1	JA	XYZ	3160	stabil
1850	100	-27	20	6	4	4890	1956	0.1	JA	XYZ	3160	stabil
1860	20	-32	20	4.5	4	4890	1956	0.1	JA	XYZ	3160	stabil
1870	40	-32	20	4.5	4	4640	1856	0.1	JA	XYZ	3160	stabil
1880	60	-32	20	4.5	4	4390	1756	0.1	JA	XYZ	3160	instabil
1890	80	-32	20	4.5	4	4140	1656	0.1	JA	XYZ	3160	semistabil
1900	100	-32	20	4.5	4	6140	2456	0.1	JA	XYZ	3160	stabil
1910	20	-38	20	6	4	3890	1556	0.1	JA	XYZ	3160	stabil
1920	40	-38	20	6	4	4140	1656	0.1	JA	XYZ	3160	instabil
1930	60	-38	20	6	4	4390	1756	0.1	JA	XYZ	3160	instabil
1940	80	-38	20	6	4	4640	1856	0.1	JA	XYZ	3160	instabil
1950	100	-38	20	6	4	5640	2256	0.1	JA	XYZ	3160	stabil
1960	20	-44	20	6	4	5640	2256	0.1	JA	XYZ	3160	instabil
1970	40	-44	20	6	4	5890	2356	0.1	JA	XYZ	3160	instabil
1980	60	-44	20	6	4	6890	2756	0.1	JA	XYZ	3160	stabil
1990	80	-44	20	6	4	6640	2656	0.1	JA	XYZ	3160	semistabil
2000	100	-44	20	6	4	6390	2556	0.1	JA	XYZ	3160	stabil
2010	20	-50	20	6.5	4	3890	1556	0.1	JA	XYZ	3160	stabil

2020	40	-50	20	6.5	4	4140	1656	0.1	JA	XYZ	3160	instabil
2030	60	-50	20	6.5	4	5390	2156	0.1	JA	XYZ	3160	instabil
2040	80	-50	20	6.5	4	5140	2056	0.1	JA	XYZ	3160	instabil
2050	100	-50	20	6.5	4	5140	2056	0.1	JA	XYZ	3160	stabil
2060	20	-56	20	5.5	4	5640	2256	0.1	JA	XYZ	3160	stabil
2070	40	-56	20	5.5	4	5890	2356	0.1	JA	XYZ	3160	instabil
2080	60	-56	20	5.5	4	6140	2456	0.1	JA	XYZ	3160	instabil
2090	80	-56	20	5.5	4	3890	1556	0.1	JA	XYZ	3160	stabil
2100	100	-56	20	5.5	4	3890	1556	0.1	JA	XYZ	3160	stabil
2110	20	-61	20	5.5	4	8140	3256	0.1	JA	XYZ	3160	instabil
2120	40	-61	20	5.5	4	8390	3356	0.1	JA	XYZ	3160	stabil
2130	60	-61	20	5.5	4	7890	3156	0.1	JA	XYZ	3160	stabil
2140	80	-61	20	5.5	4	8640	3456	0.1	JA	XYZ	3160	stabil
2150	100	-61	20	5.5	4	8640	3456	0.1	JA	XYZ	3160	stabil
2160	20	-66	20	4.5	4	4890	1956	0.1	JA	XYZ	3160	stabil
2170	40	-66	20	4.5	4-1	4890	1956	0.1	JA	XYZ	3160	stabil
2180	60	-66	20	4.5	4-1	4890	1956	0.1	JA	XYZ	3160	stabil
2190	80	-66	20	4.5	4-1	4890	1956	0.1	JA	XYZ	3160	stabil
2200	100	-66	20	4.5	4-1	4890	1956	0.1	JA	XYZ	3160	stabil

## Anhang C

# Fräserdatenblatt

# „Wave Mill“- Serie WEX - Typ

## Erweiterung



- Die neuen Sorten ACM200/ACM300 speziell geeignet für den ISO **M** und **S** Bereich
- Spanbrecher vom Typ E/EH für rostfreien Stahl und hitzebeständige Legierungen
- Hocheffiziente Bearbeitung
- Präzise Zerspanung und exzellente Oberflächengüte

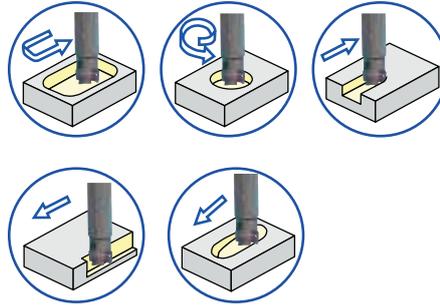


# „Wave Mill“- Serie WEX - Typ



## Allgemeine Merkmale

### Großer Anwendungsbereich



## Schrägeintauchen

Fräser- durchmesser	Max. Schrägeintauchwinkel	
	Typ 2000	Typ 3000
Ø14	5°	
Ø16	4°	
Ø20	4°	
Ø25	2°	5°
Ø32	1°30'	3°
Ø40	1°	2°
Ø50	0°30'	1°
Ø63	0°30'	0°30*
Ø80		0°30'
Ø100~		Nicht möglich

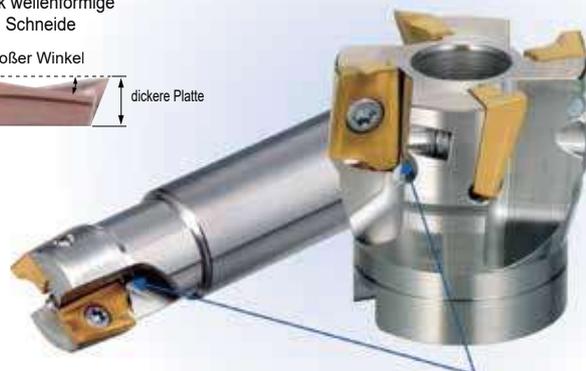
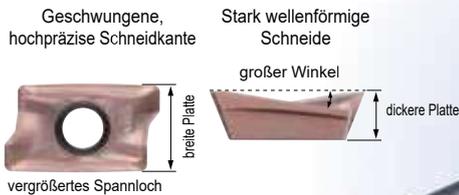
Der maximale Eintauchwinkel ist vom Durchmesser des Werkzeuges abhängig.

### Präzisionsplatte mit stabiler Schneidkante und verbesserten Schnittkräften

Einzigartig geschwungene Schneidkante zur Verbesserung der Schnittkräfte und erhöhter Schneidenstabilität.

Höchste Oberflächenqualität durch hochpräzise Schneiden.

Weicher Schnitt auch bei Nuten und labileren Maschinen.



### Innenkühlung

Verbesserte Spanabfuhr durch Druckluft- oder Emulsionszufuhr.

### Große Auswahl an Platten

6 verschiedene Schneidengeometrien (L, G, H, E, EH und S)

9 neue Fräsorten für einen großen Anwendungsbereich

- ACP100, ACP200, ACP300

(zum Fräsen von Stahl)

- ACK200, ACK300

(zum Fräsen von Grauguss)

- ACM200, ACM300

(für rostfreien Stahl und hitzebeständige Legierungen)

- DL1000, H1

(zur Bearbeitung von Aluminium)

### Hochstabiler Fräskörper

Verbesserte Haltbarkeit gegen Abrieb und Korrosion durch spezielle Oberflächenbehandlung des Fräskörpers.

Größere Schrauben verbessern die Klemmkraft und die Stabilität.

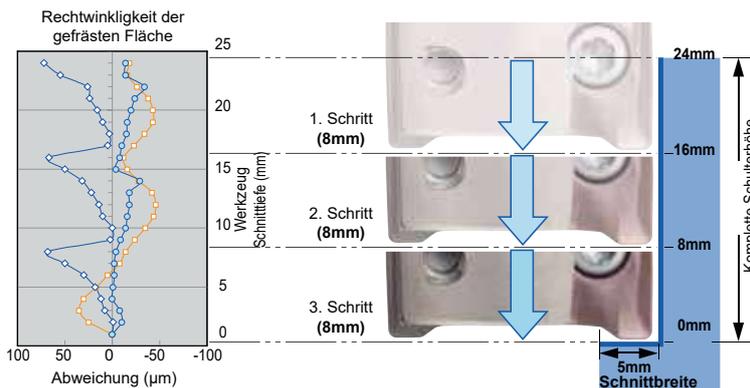
## Produktpalette

Fräser Typ	Kat. Nr.	Ausführungen	Durchmesserbereich (mm)				Fräser
			Ø10	Ø20	Ø40	Ø60	
Schaftfräser	WEX 2000E	Standard	14			63	
	WEX 2000EL	Lange Ausführung	14			40	
	WEX 2000EW	Standard mit Weldon Schaft	16			20	
	WEX 3000E	Standard	25			63	
	WEX 3000EL	Lange Ausführung	25			40	
	WEX 3000EW	Standard mit Weldon Schaft	25			32	
Aufsteckfräser	WEX 2000F	Aufsteckfräser	40				
	WEX 3000F	Aufsteckfräser	40				
Modularfräser	WEX 2000M	Modularfräser	16			40	
	WEX 3000M	Modularfräser	25			40	

## Schnittleistung

### Schulterfräsen

Schulterfräser mit guter Rechtwinkligkeit.  
Hochpräzisionsplatten für sehr geringe Absatzmarken. (50% weniger als der Wettbewerb)

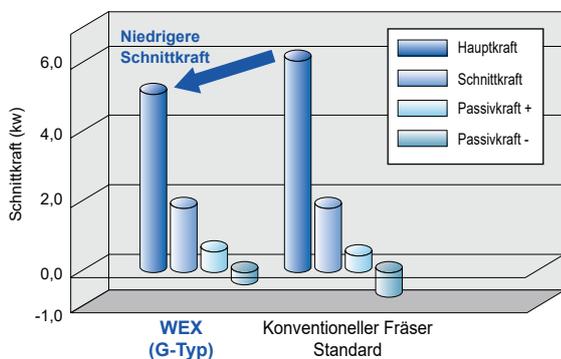


Schnittbedingungen:  
Werkstückstoff: St 42-3  
Fräser: WEX 3032 E ( $\varnothing 32$ )  
Platte: AXMT170508PEER-G  
Sorte: ACP200

$v_c = 150\text{m/min}$   
 $f_z = 0,15\text{mm/Zahn}$ , ( $v_f = 675\text{m/min}$ )  
 $a_e = 5\text{mm}$   
 $a_p = 8\text{mm} \times 3\text{ Schnitte}$   
 trocken

### Schnittkraft

Schnittkraft durchschnittlich 15% niedriger zu herkömmlichen Werkzeugen.

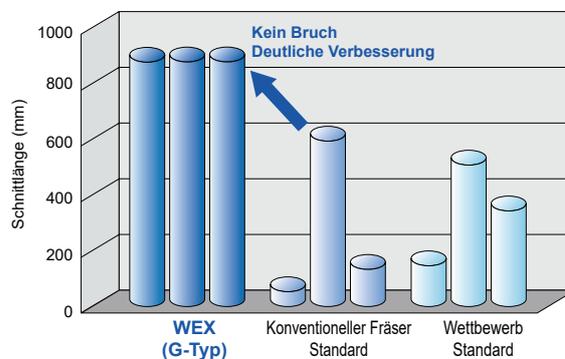


Schnittbedingungen:  
Werkstückstoff: C50  
Fräser: WEX 3032 E ( $\varnothing 32$ )  
Platte: AXMT170508PEER-G  
Sorte: ACP200

$v_c = 200\text{m/min}$   
 $f_z = 0,2\text{mm/Zahn}$ , ( $v_f = 1.200\text{m/min}$ )  
 $a_e = 8\text{mm}$   
 $a_p = 10\text{mm}$   
 trocken

### Zähigkeit

Deutlich verbesserte Bruchfestigkeit durch höhere Schneidkantenstabilität.

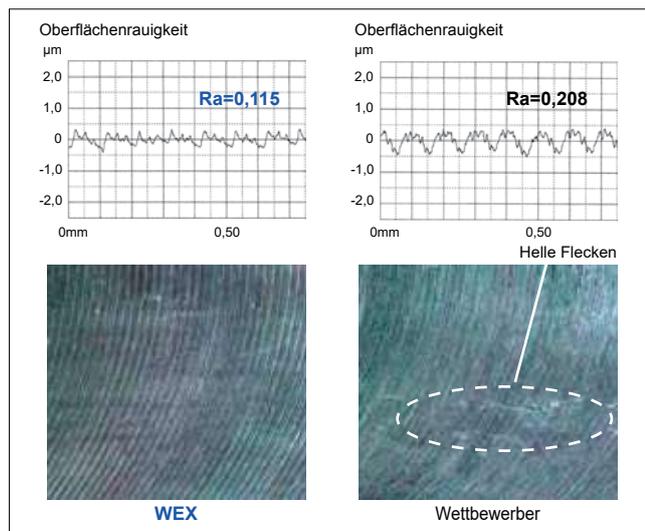


Schnittbedingungen:  
Werkstückstoff: 42CrMo4  
Fräser: WEX 3032 E ( $\varnothing 32$ )  
Platte: AXMT170508PEER-G  
Sorte: ACP200

$v_c = 100\text{m/min}$   
 $f_z = 0,4\text{mm/Zahn}$ , ( $v_f = 1.260\text{m/min}$ )  
 $a_e = 25\text{mm}$   
 $a_p = 3\text{mm}$   
 trocken

### Oberflächenrauigkeit (Fräsen von Al-Legierungen)

Glatte Oberfläche frei von hellen Flecken.

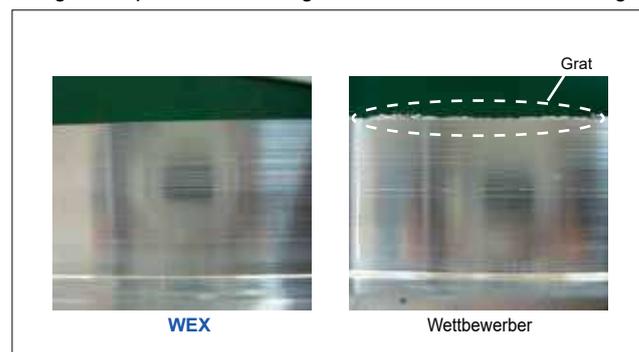


Schnittbedingungen:  
Werkstückstoff: A5052  
Fräser: WEX 3032 E ( $\varnothing 32$ )  
Platte: AXET170504PEFR-S  
Sorte: (H1)

$v_c = 800\text{m/min}$   
 $f_z = 0,1\text{mm/Zahn}$ , ( $v_f = 2.400\text{m/min}$ )  
 $a_e = 16\text{mm}$   
 $a_p = 10\text{mm}$   
 trocken

### Schnittleistung

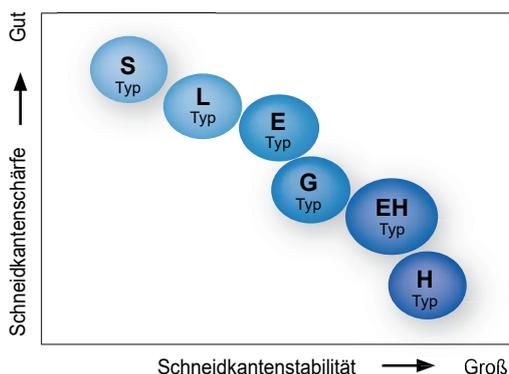
Der große Spanwinkel erzeugt eine deutliche Gratminderung.



Schnittbedingungen:  
Werkstückstoff: A5052  
Fräser: WEX 3032 E ( $\varnothing 32$ )  
Platte: AEMT170504PEFR-S  
Sorte: (H1)

$v_c = 800\text{m/min}$   
 $f_z = 0,1\text{mm/Zahn}$ , ( $v_f = 2.400\text{m/min}$ )  
 $a_e = 5\text{mm}$   
 $a_p = 12\text{mm}$   
 trocken

## Auswahl Spanbrecher



## Produktenerweiterung

Spanbrecher E- Typ für hitzebeständige Legierungen

Für die 2000er Serie  
R0,4 / R0,8 / R1,2

Für die 3000er Serie  
R0,4 / R0,8 / R1,2 /  
R1,6 / R2,0 / R3,0



## Spanbrecherauswahl

Werkstückstoff	Stahl, Grauguss			Rostfreier Stahl, hitzebeständige Legierungen		Aluminium
	L	G	H	E	EH	S
Spanbrecher						
Leistung	Niedrige Schnittkraft	Allgem. Anwendung	Verstärkte Schneidkante	Allgem. Anwendung	Verstärkte Schneidkante	Scharfe Schneide
Spanwinkel der Spanbrecher (2000er Serie)						
Spanwinkel der Spanbrecher (3000er Serie)						
Anwendungsbereich	Leichte Bearbeitung, Fräsen in labilen Verhältnissen, weniger Gratbildung	<b>Hauptspanbrecher</b> allgemeines Fräsen, unterbrochener Schnitt	Schwere Bearb., unterbr. Schnitt u. stabile Verhältnisse bei härteren Materialien	Leichte Bearbeitung, und allgem. Anwendung	Stark unterbrochene Bearbeitung	Aluminiumlegierung und Nichteisenmetalle

## Spanbrecher für rostfreien Stahl, E / EH-Typ für hitzebeständige Legierungen

### Eigenschaften des E / EH Typs

- Verbesserter Adhäsionswiderstand  
Eine glatte Schneidkante erlaubt einen gleichmäßigen Spanabfluss.
- Verbesserte Schneidleistung  
Optimierter Spanwinkel und Schneidkante.

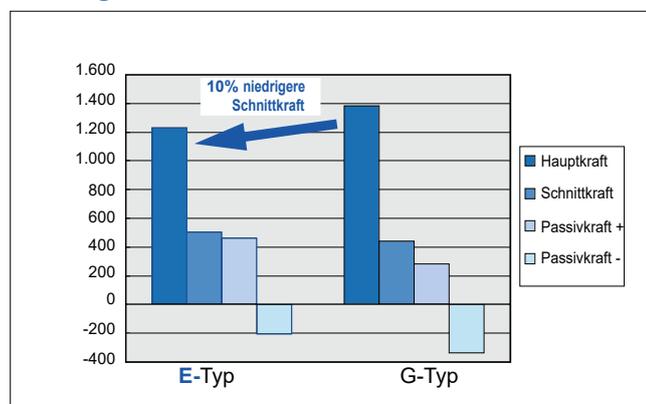


E / EH -Typ



L-Typ / G-Typ

## Vergleich Schnittkraft

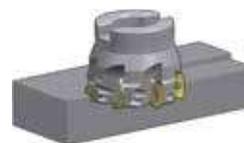


Schnittbedingungen:

Werkstückstoff: X5CrNi1810  
Fräser: WEX 2050 F  
Platte:  
Sorte:

$v_c = 180\text{m/min}$   
 $f_z = 0,15\text{mm/Zahn}$   
 $a_e = 20\text{mm}$   
 $a_p = 3\text{mm}$   
trocken

## Vergleich Standzeit



	E-Typ	G-Typ
Werkzeugverschleiß		
Schnittlänge	1,5m	0,9m

Schnittbedingungen:

Werkstückstoff: X5CrNi1810  
Fräser: WEX 2050 F  
Platte:  
Sorte:

$v_c = 120\text{m/min}$   
 $f_z = 0,15\text{mm/Zahn}$   
 $a_e = 20\text{mm}$   
 $a_p = 3\text{mm}$   
trocken

## Sortenübersicht

ISO	Sorte	Schichten bis leichtes Schruppen	Mittleres Schruppen	Schruppen bis schweres Schruppen
P	Beschichtetes Hartmetall	ACP100		
		ACP200		ACP300
M	Besch. Hartmetall	ACM200		ACM300

ISO	Sorte	Schichten bis leichtes Schruppen	Mittleres Schruppen	Schruppen bis schweres Schruppen
K	Besch. Hartmetall	ACK200		ACK300
N	Hartmetall	DL1000		H1

▽ CVD ▲ PVD

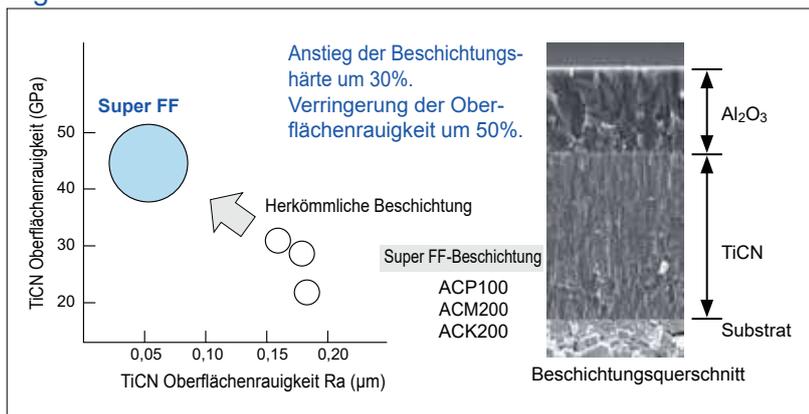
## Eigenschaften der Schichten

### Super FF-Beschichtung

Unser einzigartiger, neuer CVD-Prozess, die Super FF-Beschichtungstechnologie, erzeugt ultraflache Grenzflächen zwischen den Beschichtungslagen sowie ultrafeine Beschichtungspartikel für höhere Zuverlässigkeit und längere Standzeit.

**C** CVD: Chemical Vapor Deposition  
(Chemische Gasphasenabscheidung)

### Eigenschaften



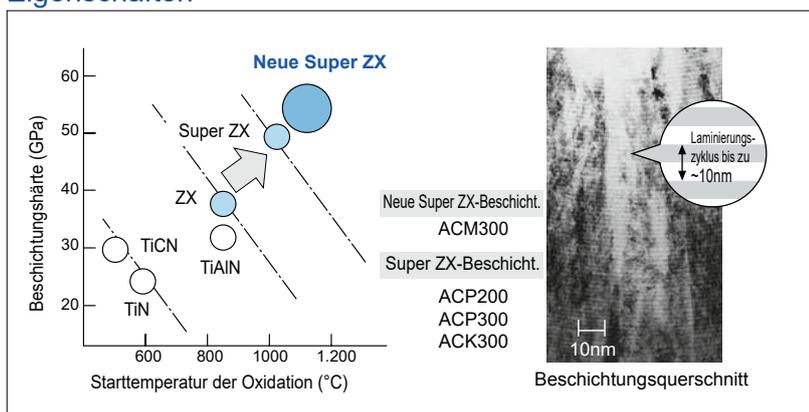
- Die glatte Beschichtungsfläche bietet optimalen Schutz vor Adhäsion und Mikroausbrüchen.
- Verbesserte Schichthftung.
- Härter als herkömmliche Beschichtungen und enorme Verbesserung der Verschleißfestigkeit.
- Mehr als das 1,5-Fache für eine schnellere und effizientere Bearbeitung möglich.
- Unter gleichen Schnittbedingungen kann die Standzeit gegenüber herkömmlichen Schneidstoffen verdoppelt werden.

### Neue Super ZX-Beschichtung / Super ZX-Beschichtung

**P** PVD: Physical Vapor Deposition  
(Physikalische Gasphasenabscheidung)

Unter Einsatz unserer proprietären Dünnschicht-Beschichtungstechnologie und modernster Nanotechnologie hat Sumitomo Electric die neuen Beschichtungen „Neue Super ZX“ und „Super ZX“ entwickelt. Diese Beschichtungen bestehen aus ungefähr 1.000 abwechselnden Schichten im Nanobereich (1 Nanometer = 1 Billionstel eines Meters).

### Eigenschaften



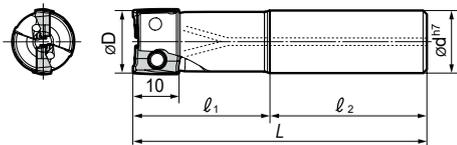
Im Vergleich zu herkömmlichen Beschichtungen:

- 40% erhöhte Beschichtungshärte und um 200°C erhöhte Starttemperatur der Oxidation
- mindestens 1,5-fach schnellere und effizientere Bearbeitung
- bei gleichen Schnittbedingungen doppelte Standzeit

# „Wave Mill“ - Serie WEX 2000 - Typ

## WEX 2000 E/EL

Spanwinkel	Radial	8°-15°
	Axial	16°-24°



### Fräskörper (E-Typ als Standard)

Kat. Nr.	Lager	Abmessungen (mm)					Anzahl Zähne	Gewicht (kg)
		øD	ød	l <sub>1</sub>	l <sub>2</sub>	L		
WEX 2014 E	●	14	16	25	55	80	1	0,10
2016 E	●	16	16	25	75	100	2	0,13
2018 E	○	18	16	25	75	100	2	0,14
2020 E	●	20	20	30	80	110	3	0,22
2022 E	○	22	20	30	80	110	3	0,23
WEX 2025 E	●	25	25	35	85	120	4	0,38
2028 E	○	28	25	35	85	120	4	0,39
2030 E	○	30	25	35	85	120	4	0,40
2032 E	●	32	32	40	90	130	5	0,70
2040 E	○	40	32	30	120	150	6	0,91
WEX 2050 E	○	50	32	30	120	150	7	1,02
2063 E	○	63	32	30	120	150	8	1,22

### Fräskörper (EL-Typ als lange Ausführung)

Kat. Nr.	Lager	Abmessungen (mm)					Anzahl Zähne	Gewicht (kg)
		øD	ød	l <sub>1</sub>	l <sub>2</sub>	L		
WEX 2014 EL	●	14	16	25	95	120	1	0,14
2016 EL	●	16	16	25	120	145	2	0,19
2018 EL	○	18	16	25	120	145	2	0,19
2020 EL	●	20	20	40	110	150	2	0,32
2022 EL	○	22	20	30	120	150	2	0,33
WEX 2025 EL	●	25	25	50	120	170	2	0,55
2028 EL	○	28	25	30	140	170	2	0,59
2030 EL	○	30	25	30	140	170	2	0,60
2032 EL	○	32	32	60	120	180	2	0,99
2040 EL	○	40	32	30	150	180	2	1,12

### Fräskörper (Lange Ausführung mit schlankem Schaft)

Kat. Nr.	Lager	Abmessungen (mm)					Anzahl Zähne	Gewicht (kg)
		øD	ød	l <sub>1</sub>	l <sub>2</sub>	L		
WEX 2016 EL15	●	16	15	25	120	145	2	0,17
2020 EL19	●	20	19	40	110	150	2	0,30
2025 EL24	●	25	24	50	120	170	2	0,53
2025 EL24Z3	○	25	24	50	120	170	3	0,50
3032 EL30	○	32	30	60	120	180	2	0,95

### Fräskörper (Standard mit Weldon-Schaft)

Kat. Nr.	Lager	Abmessungen (mm)					Anzahl Zähne	Gewicht (kg)
		øD	ød	l <sub>1</sub>	l <sub>2</sub>	L		
WEX 2016 EW	●	16	16	25	75	100	2	0,12
2020 EW	●	20	20	30	80	110	3	0,21

Fräsplatten sind nicht im Lieferumfang enthalten.

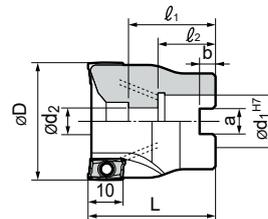
- Eurolager
- Liefertermin auf Anfrage

### Ersatzteile

Schraube	Schlüssel	Empfohlenes Anzugsmoment (N·m)	Geeignete Fräser
		2,0	WEX 2014 ~ WEX 2018 WEX 2020 ~ WEX 2063
BFTX 0305 IP BFTX 0306 IP	TRDR 08 IP	2,0	WEX 2000 F

## WEX 2000 F

Spanwinkel	Radial	16°-18°
	Axial	23°-25°



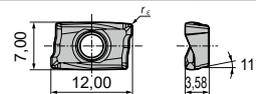
### Fräskörper (F-Typ als Aufsteckfräser)

Kat. Nr.	Lager	Abmessungen (mm)								Anzahl Zähne	Gewicht (kg)
		øD	ød <sub>1</sub>	ød <sub>2</sub>	a	b	L	l <sub>1</sub>	l <sub>2</sub>		
WEX 2040 F	●	40	16	9	8,4	5,6	40	28	18	6	0,19
2050 F	●	50	22	11	10,4	6,3	40	26	20	7	0,29
2063 F	●	63	22	11	10,4	6,3	40	26	20	8	0,51

Fräsplatten sind nicht im Lieferumfang enthalten.

- Eurolager

### Schneidplatten für WEX2000-Typ



Anwendung	Beschichtetes Hartmetall					Hartmet.	DLC	Radius r <sub>ε</sub>	
Hochgeschw./Leichtbearb.	P		K	M <sub>S</sub>	M <sub>S</sub>	K	N		
Allgemeine Anwendung	P		K	M <sub>S</sub>	M <sub>S</sub>		N		
Schruppen	P	P	K	M <sub>S</sub>	M <sub>S</sub>		N		
Kat. Nr.	ACP100	ACP200	ACP300	ACK200	ACK300	ACM200	ACM300	H1	DL1000
AXMT 123504 PEER-G	●	●	●	●	●			-	0,4
123508 PEER-G	●	●	●	●	●			-	0,8
123512 PEER-G	●	●	●	●	●			-	1,2
AXMT 123504 PEER-H	●	●	●	●	●			-	0,4
123508 PEER-H	●	●	●	●	●			-	0,8
123512 PEER-H	●	●	●	●	●			-	1,2
AXMT 123504 PEER-E						●	●	-	0,4
123508 PEER-E						●	●	-	0,8
123512 PEER-E						●	●	-	1,2
AXMT 123508 PEER-EH			●			●	●	-	0,8
AXET 123502 PEFR-S	-	-	-	-	-	-	●	●	0,2
123504 PEFR-S	-	-	-	-	-	-	●	●	0,4
123508 PEFR-S	-	-	-	-	-	-	●	●	0,8

- G - allgemeine Anwendung
- H - verstärkte Schneidkante
- E - weicher Schnitt für rostfreien Stahl / hitzebeständige Legierungen
- EH - verstärkte Ausführung für rostfreien Stahl / hitzebeständige Legier.
- S - für Aluminium
- Eurolager
- nicht möglich

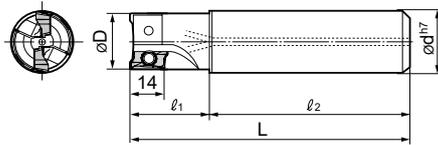
### Ersatzteile

Schraube	Schlüssel	Empfohlenes Anzugsmoment (N·m)	Geeignete Fräser
		2,0	WEX 2000 F
BFTX 0306 IP	TRDR 08 IP	2,0	WEX 2000 F

# „Wave Mill“ - Serie WEX 3000 - Typ

## WEX 3000 E/EL

Spanwinkel	Radial	8°-15°
	Axial	16°-24°



### Fräskörper (E-Typ als Standard)

Kat. Nr.	Lager	Abmessungen (mm)					Anzahl Zähne	Gewicht (kg)
		øD	ød	l <sub>1</sub>	l <sub>2</sub>	L		
WEX 3025 E	●	25	25	35	85	120	2	0,37
3028 E	○	28	25	35	85	120	2	0,39
3030 E	○	30	25	40	90	130	3	0,42
WEX 3032 E	●	32	32	40	90	130	3	0,67
3035 E	○	35	32	40	90	130	3	0,69
3040 E	●	40	32	50	120	170	4	1,01
3050 E	○	50	32	50	120	170	5	1,23
3063 E	○	63	32	50	120	170	6	1,58

### Fräskörper (Standard mit schlankem Schaft)

Kat. Nr.	Lager	Abmessungen (mm)					Anzahl Zähne	Gewicht (kg)
		øD	ød	l <sub>1</sub>	l <sub>2</sub>	L		
WEX 3025 E20	○	25	20	35	85	120	2	0,25
3032 E25	○	32	25	40	90	130	3	0,43

### Fräskörper (EL-Typ als lange Ausführung)

Kat. Nr.	Lager	Abmessungen (mm)					Anzahl Zähne	Gewicht (kg)
		øD	ød	l <sub>1</sub>	l <sub>2</sub>	L		
WEX 3025 EL	●	25	25	50	120	170	2	0,54
3028 EL	○	28	25	50	120	170	2	0,56
3030 EL	○	30	25	60	120	180	2	0,60
3032 EL	●	32	32	60	120	180	2	0,95
3035 EL	○	35	32	60	120	180	2	0,98
WEX 3040 EL	●	40	32	80	140	220	2	1,38

### Fräskörper (Standard mit Weldon-Schaft)

Kat. Nr.	Lager	Abmessungen (mm)					Anzahl Zähne	Gewicht (kg)
		øD	ød	l <sub>1</sub>	l <sub>2</sub>	L		
WEX 3025 EW	●	25	25	35	85	120	2	0,36
3032 EW	●	32	32	40	90	130	3	0,65

Fräsplatten sind nicht im Lieferumfang enthalten.

- Eurolager
- Liefertermin auf Anfrage

\*Achtung: Fräskopf muss modifiziert werden, wenn WSP mit Radius 2,0mm oder größer benutzt wird.

Standard Fase = 0,5mm x 45°.



Fase = 1mm x 45°  
bei Plattenradius 2,0mm  
(AXMT 170520 PEER)

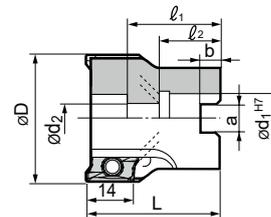
Fase = 1,5mm x 45°  
bei Plattenradius 3,0mm  
(AXMT 170530 PEER)

### Ersatzteile

Schraube	Schlüssel	Empfohlenes Anzugsmoment (N·m)	Geeignete Schaftfräser
BFTX 0407 IP BFTX 0409 IP	TRDR 15 IP	3,0	WEX 3025 - WEX 3030 WEX 3032 - WEX 3063

## WEX 3000 F

Spanwinkel	Radial	12°-15°
	Axial	19°-24°



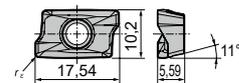
### Fräskörper (F-Typ als Aufsteckfräser)

Kat. Nr.	Lager	Abmessungen (mm)							Anzahl Zähne	Gewicht (kg)	
		øD	ød <sub>1</sub>	ød <sub>2</sub>	a	b	L	l <sub>1</sub>			l <sub>2</sub>
WEX 3040 F	●	40	16	9	8,4	5,6	40	28	18	4	0,16
3050 F	●	50	22	11	10,4	6,3	40	26	20	5	0,25
3063 F	●	63	22	11	10,4	6,3	40	26	20	6	0,48
3080 F	●	80	27	13,5	12,4	7,0	50	31	25	7	1,06
3100 F	●	100	32	32	14,4	8,5	50	32	26	8	1,99

Fräsplatten sind nicht im Lieferumfang enthalten.

- Eurolager

### Schneidplatten für WEX 3000-Typ



Anwendung	Beschichtetes Hartmetall						Hartmet.	DLC	Radius r <sub>ε</sub>
	P	P	P	K	M <sub>S</sub>	M <sub>S</sub>			
Hochgeschw./Leichtbearb.	P			K		M <sub>S</sub>	N	N	
Allgemeine Anwendung		P	P	K		M <sub>S</sub>	N	N	
Schruppen					K	M <sub>S</sub>			
Kat. Nr.	ACP100	ACP200	ACP300	ACK200	ACK300	ACM200	ACM300	H1	DL1000
AXMT 170508 PEER-L	●	●	●	●	●			-	-
AXMT 170504 PEER-G	●	●	●	●	●			-	-
170508 PEER-G	●	●	●	●	●			-	-
170512 PEER-G	●	●	●	●	●			-	-
170516 PEER-G	●	●	●	●	●			-	-
170520 PEER-G*	●	●	●	●	●			-	-
170530 PEER-G*	●	●	●	●	●			-	-
AXMT 170508 PEER-H	●	●	●	●	●			-	-
170512 PEER-H	○	○	○	●	●			-	-
AXMT 170504 PEER-E				●		●	●	-	-
170508 PEER-E				●		●	●	-	-
170512 PEER-E				●		●	●	-	-
170516 PEER-E				●		●	●	-	-
170520 PEER-E*				●		●	●	-	-
170530 PEER-E*				●		●	●	-	-
AXMT 170508 PEER-EH				●		●	●	-	-
AXET 170502 PEFR-S	-	-	-	-	-	-	●	●	0,2
170504 PEFR-S	-	-	-	-	-	-	●	●	0,4
170508 PEFR-S	-	-	-	-	-	-	●	●	0,8

G - allgemeine Anwendung

H - verstärkte Schneidkante

E - weicher Schnitt für rostfreien Stahl / hitzebeständige Legierungen

EH - verstärkte Ausführung für rostfreien Stahl / hitzebeständige Legier.

S - für Aluminium

● Eurolager

○ Liefertermin auf Anfrage

- nicht möglich

\* Fräskörper muss modifiziert werden

### Ersatzteile

Schraube	Schlüssel	Empfohlenes Anzugsmoment (N·m)	Geeignete Schaftfräser
BFTX 0409 IP	TRDR 15 IP	3,0	WEX 3000 F

# „Wave Mill“ - Serie Modularfräser

## Eigenschaften

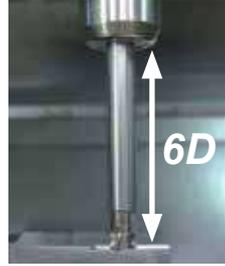
Auskragung bis zu 6 x D beim Modularschafffräser mit Hartmetallverlängerung.

Schnittbedingungen:

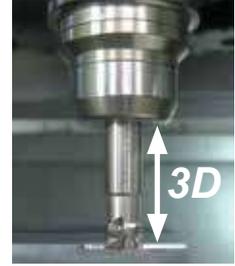
Werkstückstoff: C50  $v_c = 100\text{m/min}$   
 Fräser: WEX 2025 M12Z4 ( $\varnothing 25 \times 4$  Zähne)  $f_z = 0,1\text{mm/Zahn}$   
 Maschine: BAZ BT50  $a_e = 8\text{mm} \times 3\text{Schnitte}$   
 $a_p = 2,0\text{mm}$

Bemerkung: Die mögliche Auskragung des Fräasers hängt von der Werkzeugzusammensetzung und von der Stabilität der Maschine und des Bauteils ab.

Modularkopf + Hartmetallverlängerung



Standardschafffräser

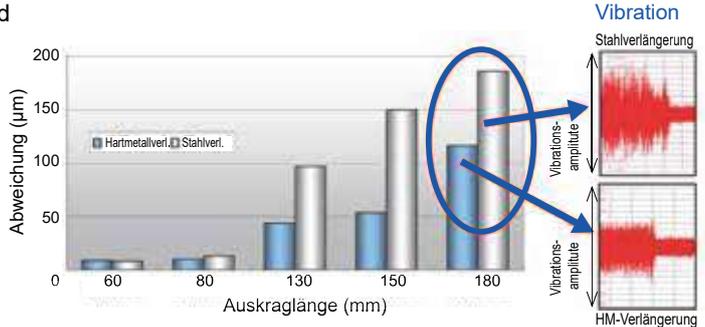
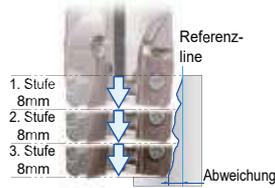


## Leistung

Hartmetallverlängerung erzielt eine höhere Genauigkeit und Standzeit als eine vergleichbare Stahlverlängerung.

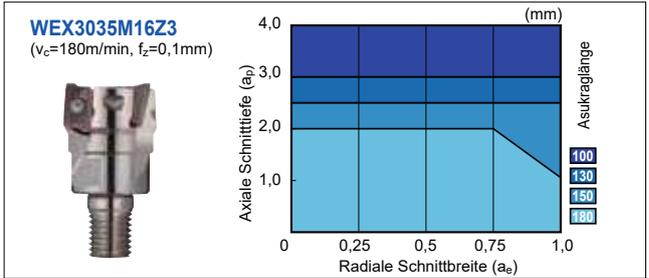
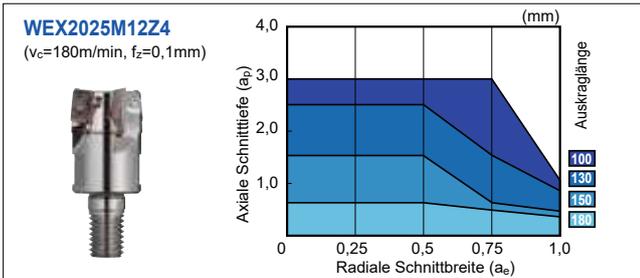
Schnittbedingungen:

Werkstückstoff: C50  
 Fräser: WEX 2025 M12Z4 ( $\varnothing 25 \times 4$  Zähne)  
 Maschine: BAZ BT50  
 $v_c = 100\text{m/min}$   
 $f_z = 0,1\text{mm/Zahn}$   
 $a_e = 8\text{mm} \times 3\text{Schnitte}$   
 $a_p = 2,0\text{mm}$



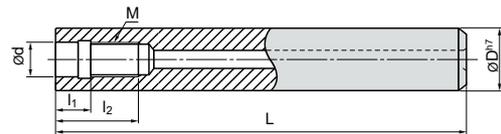
## Anwendungsbereich

Werkstückstoff: C50, Maschine: BAZ BT50 (trocken)



Hinweis: Diese Werte setzen allgem. Bearbeitungsbedingungen voraus. Je nach Maschinenverhältnissen und Werkstückspannungen müssen diese angepasst werden.

## Verlängerungen für Modulare Fräswerkzeuge



## Hartmetallverlängerung

Kat. Nr.	Lager	Abmessungen (mm)						
		M	ød	øD	L	l <sub>1</sub>	l <sub>2</sub>	L <sub>M</sub>
MA 15M08L120C	●	M8	8,5	15	120	10	18	145
15M08L160C	●	M8	8,5	15	160	10	18	185
16M08L120C	●	M8	8,5	16	120	10	18	145
16M08L160C	●	M8	8,5	16	160	10	18	185
MA 18M10L150C	●	M10	10,5	18	150	10	20	180
18M10L200C	●	M10	10,5	18	200	10	20	230
20M10L150C	●	M10	10,5	20	150	10	20	180
20M10L200C	●	M10	10,5	20	200	10	20	230
MA 23M12L200C	●	M12	12,5	23	200	10	22	235
23M12L250C	●	M12	12,5	23	250	10	22	285
25M12L200C	●	M12	12,5	25	200	10	22	235
25M12L250C	●	M12	12,5	25	250	10	22	285
MA 28M16L200C	●	M16	17,0	28	200	10	24	240
28M16L300C	●	M16	17,0	28	300	10	24	340
32M16L200C	●	M16	17,0	32	200	10	24	240
32M16L300C	●	M16	17,0	32	300	10	24	340

## Stahlverlängerung

Kat. Nr.	Lager	Abmessungen (mm)						
		M	ød	øD	L	l <sub>1</sub>	l <sub>2</sub>	L <sub>M</sub>
MA 16M08L120S	●	M8	8,5	16	120	10	18	145
20M10L150S	●	M10	10,5	20	150	10	20	180
25M12L200S	●	M12	12,5	25	200	10	22	235
32M16L200S	●	M16	17,5	32	200	10	24	240

## Identifikation der Katalogbezeichnung



## Empfohlenes Anzugsmoment

Gewinde	Drehmoment (N.m)	Schlüssel	
		W	S
M08	23	8	13
M10	46	8	15
M12	80	10	19
M16	90	10	24

Hinweise zum Anziehen des Fräskopfes:

Entspr. der Katalogbezeichnung erfolgt die Auswahl der Verlängerung nach nebenstehender Tabelle. Im Voraus die Größe des Befestigungsgewindes für den Fräskopf und die Verlängerung prüfen. Das entspr. Anzugsmoment beim Befestigen des Kopfes an der Verlängerung ergibt sich aus der Tabelle.

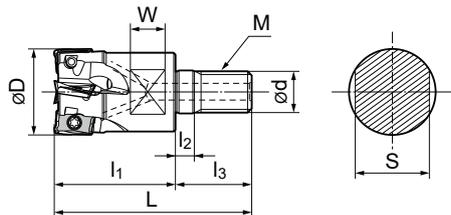
# „Wave Mill“ - Serie Modularfräser

## WEX 2000 M

Spanwinkel	Radial	10°~18°
	Axial	14°~25°

Lager	10mm
	90°

Eckfräser zur hocheffizienten und exakten Bearbeitung



Spanwinkel	Radial	8°~15°
	Axial	16°~24°

Lager	14mm
	90°



### Fräskörper

Kat. Nr.	Lager	Abmessungen (mm)									Anzahl Zähne
		øD	ød	M	L	l <sub>1</sub>	l <sub>2</sub>	l <sub>3</sub>	W	S	
WEX 2016M08Z2	●	16	8,5	M8	42	25	5	17	8	13	2
2018M08Z2	○	18	8,5	M8	42	25	5	17	8	13	2
WEX 2020M10Z3	●	20	10,5	M10	49	30	5	19	8	15	3
2022M10Z3	○	22	10,5	M10	49	30	5	19	8	15	3
WEX 2025M12Z4	●	25	12,5	M12	56	35	5	21	10	19	4
2028M12Z4	○	28	12,5	M12	56	35	5	21	10	19	4
WEX 2030M16Z4	●	30	17,0	M16	63	40	5	23	10	24	4
2032M16Z5	●	32	17,0	M16	63	40	5	23	10	24	5
2040M16Z6	○	40	17,0	M16	63	40	5	23	10	24	6

Fräsplatten sind nicht im Lieferumfang enthalten.

### Fräskörper

Kat. Nr.	Lager	Abmessungen (mm)									Anzahl Zähne
		øD	ød	M	L	l <sub>1</sub>	l <sub>2</sub>	l <sub>3</sub>	W	S	
WEX 3025M12Z2	○	25	12,5	M12	56	35	5	21	10	19	2
3028M12Z2	○	28	12,5	M12	56	35	5	21	10	19	2
WEX 3030M16Z3	○	30	17,0	M16	63	40	5	23	10	24	3
3032M16Z3	○	32	17,0	M16	63	40	5	23	10	24	3
3035M16Z3	○	35	17,0	M16	63	40	5	23	10	24	3
WEX 3040M16Z4	○	40	17,0	M16	63	40	5	23	10	24	4

Fräsplatten sind nicht im Lieferumfang enthalten.

### Verlängerung



### Schneidplatten

Anwendung	Beschichtetes Hartmetall						Hartmet.	DLC		
Hochgeschw./Leichtbearb.	P			K		M S	K N	N		
Allgemeine Anwendung	P			K		M S	K N	N		
Schruppen	P	P		K		M S				
Kat. Nr.	ACP100	ACP200	ACP300	ACK200	ACK300	ACM200	ACM300	H1	DL 1000	Radius r <sub>ε</sub>
AXMT 123504 PEER-G	●	●	●	●	●			-	-	0,4
123508 PEER-G	●	●	●	●	●			-	-	0,8
123512 PEER-G	●	●	●	●	●			-	-	1,2
AXMT 123504 PEER-H	●	●	●	●	●			-	-	0,4
123508 PEER-H	●	●	●	●	●			-	-	0,8
123512 PEER-H	●	●	●	●	●			-	-	1,2
AXMT 123504 PEER-E						●	●	-	-	0,4
123508 PEER-E						●	●	-	-	0,8
123512 PEER-E						●	●	-	-	1,2
AXMT 123508 PEER-EH			●			●	●	-	-	0,8
AXET 123502 PEFR-S	-	-	-	-	-	-	-	●	●	0,2
123504 PEFR-S	-	-	-	-	-	-	-	●	●	0,4
123508 PEFR-S	-	-	-	-	-	-	-	●	●	0,8

G - allgemeine Anwendung  
H - verstärkte Schneidkante  
E - weicher Schnitt für rostfreien Stahl / hitzebeständige Legierungen  
EH - verstärkte Ausführung für rostfreien Stahl / hitzebeständige Legier.  
S - für Aluminium

● Eurolager  
○ Liefertermin auf Anfrage - nicht möglich  
\* Fräskörper muss modifiziert werden

### Schneidplatten

Anwendung	Beschichtetes Hartmetall						Hartmet.	DLC		
Hochgeschw./Leichtbearb.	P			K		M S	K N	N		
Allgemeine Anwendung	P			K		M S	K N	N		
Schruppen	P	P		K		M S				
Kat. Nr.	ACP100	ACP200	ACP300	ACK200	ACK300	ACM200	ACM300	H1	DL 1000	Radius r <sub>ε</sub>
AXMT 170508 PEER-L	●	●	●	●	●			-	-	0,8
AXMT 170504 PEER-G	●	●	●	●	●			-	-	0,4
170508 PEER-G	●	●	●	●	●			-	-	0,8
170512 PEER-G	●	●	●	●	●			-	-	1,2
170516 PEER-G	●	●	●	●	●			-	-	1,6
170520 PEER-G*	●	●	●	●	●			-	-	2,0
170530 PEER-G*	●	●	●	●	●			-	-	3,0
AXMT 170508 PEER-H	●	●	●	●	●			-	-	0,8
170512 PEER-H	○	○	○	●	●			-	-	1,2
AXMT 170504 PEER-E						●	●	-	-	0,4
170508 PEER-E						●	●	-	-	0,8
170512 PEER-E						●	●	-	-	1,2
170516 PEER-E						●	●	-	-	1,6
170520 PEER-E*						●	●	-	-	2,0
170530 PEER-E*						●	●	-	-	3,0
AXMT 170508 PEER-EH			●			●	●	-	-	0,8
AXET 170502 PEFR-S	-	-	-	-	-	-	-	●	●	0,2
170504 PEFR-S	-	-	-	-	-	-	-	●	●	0,4
170508 PEFR-S	-	-	-	-	-	-	-	●	●	0,8

### Ersatzteile

Schraube	Schlüssel	Empfohlenes Anzugs-moment (N·m)	Geeignete Schafffräser
BFTX 0305 IP		2,0	WEX 2016M, WEX 2018M
BFTX 0306 IP	TRDR 08 IP	2,0	WEX 2020M ~ WEX 2040M
BFTX 0407 IP		3,0	WEX 3025M ~ WEX 3030M
BFTX 0409 IP	TRDR 15 IP	3,0	WEX 3032M ~ WEX 3040M

### Identifikation der Katalogbezeichnung

**WEX** Fräser-bezeichnung  
**2** Platten-größe  
**016** Fräser-durchmesser  
**M08** Einschraub-gewinde  
**Z2** Anzahl Zähne

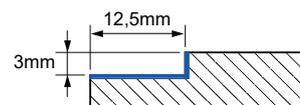
# „Wave Mill“- Serie WEX - Typ

## Empfohlene Schnittdaten

### WEX2000 Serie

Fräser: WEX2025E      Platte: AXMT123508PEER - □

Schnittdaten:  $a_p = 3\text{mm}$ ,  $a_e = 12,5\text{mm}$ , trocken



ISO	Material	HB	Spanbrecher	Hartmetall, beschichtet												DLC							
				ACP100	ACP200	ACP300	ACK200	ACK300	ACM200	ACM300	DL1000												
				Vorschub (mm/Zahn)																			
				0,08	0,15	0,20	0,08	0,15	0,20	0,08	0,15	0,20	0,08	0,15	0,20	0,08	0,15	0,20	0,08	0,15	0,20	0,05	0,15
Schnittgeschwindigkeit $v_c$ (m/min)																							
P	Unlegierter Stahl, <0, 15%C, vergütet	125	G	380	350	330	350	330	315	330	315	295											
	" , <0, 45%C, vergütet	190	G	285	255	235	255	235	220	235	220	220											
	" , <0, 45%C, angelassen	250	G	235	210	190	210	190	170	190	170	150											
	" , <0, 75%C, vergütet	270	G	190	162	143	171	152	133	152	133	115											
	" , <0, 75%C, angelassen	300	G	145	115	95	115	95	75	95	75	55											
	Niedrig legierter Stahl, vergütet	180	G	265	235	220	235	220	200	220	200	180											
	" , angelassen	275	G	170	145	125	150	130	115	130	115	95											
	" , angelassen	300	G	150	125	105	135	115	95	115	95	75											
	" , angelassen	350	G	125	95	75	105	85	65	85	65	45											
	Hoch legierter und Werkzeugstahl, vergütet	200	G	235	210	190	210	190	170	190	170	150											
" , angelassen	325	G	125	95	75	95	75	55	75	55	35												
M	Rostfreier Stahl, ferritisch/martensitisch, vergütet	200	E														175	155	125	155	140	110	
	Martensitisch, angelassen	240	EH														160	140	110	145	125	100	
	Austenitisch	180	E														190	170	140	170	150	125	
K	Grauguss		G										285	255	235	255	235	220					
	Kugelgraphitguss		G										190	160	140	160	140	125					
S	Hitzebeständige Legierungen, Fe basierend, vergütet	300	E														50	40		45	35		
	" , angelassen	330	E														35	25		30	20		
N	Aluminiumlegierung, Si < 13%		S																		1000	750	500
	Aluminiumlegierung, Si > 13%		S																		250	200	170
	Kupferlegierung		S																		350	330	300

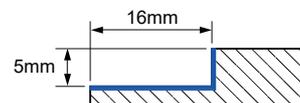
Die empfohlenen Schnittdaten dienen als Richtlinie. Die tatsächlichen Zerspanungsbedingungen variieren je nach Bearbeitungsmaschine, Anwendung und Einspannung. Die Werte sollten entsprechend der aktuellen Bedingungen angepasst werden.

Beim Nutfräsen ist der Vorschub auf 70% der in der Tabelle angegebenen Werte zu reduzieren.

### WEX3000 Serie

Fräser: WEX30325E      Platte: AXMT170508PEER - □

Schnittdaten:  $a_p = 5\text{mm}$ ,  $a_e = 16\text{mm}$ , trocken

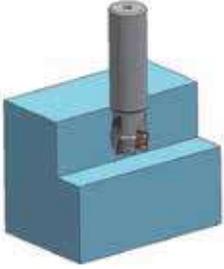
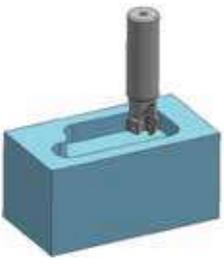
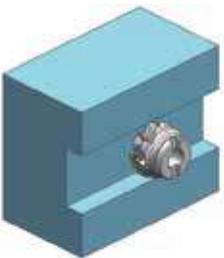
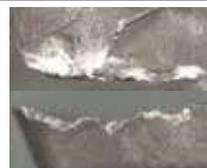
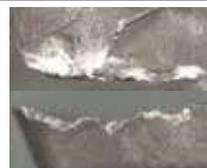
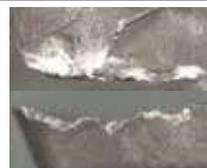


ISO	Material	HB	Spanbrecher	Hartmetall, beschichtet												DLC								
				ACP100	ACP200	ACP300	ACK200	ACK300	ACM200	ACM300	DL1000													
				Vorschub (mm/Zahn)																				
				0,12	0,25	0,35	0,12	0,25	0,35	0,12	0,25	0,35	0,12	0,25	0,35	0,12	0,25	0,35	0,12	0,25	0,35	0,12	0,25	0,35
Schnittgeschwindigkeit $v_c$ (m/min)																								
P	Unlegierter Stahl, <0, 15%C, vergütet	125	G	400	370	350	370	350	330	350	330	310												
	" , <0, 45%C, vergütet	190	G	300	270	250	270	250	230	250	230	210												
	" , <0, 45%C, angelassen	250	G	250	220	200	220	200	180	200	180	160												
	" , <0, 75%C, vergütet	270	G	200	170	150	180	160	140	160	140	120												
	" , <0, 75%C, angelassen	300	G	150	120	100	120	100	80	100	80	60												
	Niedrig legierter Stahl, vergütet	180	G	280	250	230	250	230	210	230	210	190												
	" , angelassen	275	G	180	150	130	160	140	120	140	120	100												
	" , angelassen	300	G	160	130	110	140	120	100	120	100	80												
	" , angelassen	350	G	130	100	80	110	90	70	90	70	50												
	Hoch legierter und Werkzeugstahl, vergütet	200	G	250	220	200	220	200	180	200	180	160												
" , angelassen	325	G	130	100	80	100	80	60	80	60	40													
M	Rostfreier Stahl, ferritisch/martensitisch, vergütet	200	E														185	165	135	165	150	120		
	Martensitisch, angelassen	240	EH														170	150	120	150	135	110		
	Austenitisch	180	E														200	180	150	180	160	135		
K	Grauguss		G										300	270	250	270	250	230						
	Kugelgraphitguss		G										200	170	150	170	150	130						
S	Hitzebeständige Legierungen, Fe basierend, vergütet	300	E														50	30		45	25			
	" , angelassen	330	E														50	30		45	25			
N	Aluminiumlegierung, Si < 13%		S																		1000	750	500	
	Aluminiumlegierung, Si > 13%		S																		250	200	170	
	Kupferlegierung		S																		350	330	300	

Die empfohlenen Schnittdaten dienen als Richtlinie. Die tatsächlichen Zerspanungsbedingungen variieren je nach Bearbeitungsmaschine, Anwendung und Einspannung. Die Werte sollten entsprechend der aktuellen Bedingungen angepasst werden.

Beim Nutfräsen ist der Vorschub auf 70% der in der Tabelle angegebenen Werte zu reduzieren.

■ **Empfohlene Schnittbedingungen**

<p><b>Quader, 42CrMo4</b></p> <p>Auch wenn sehr kleine Absatzmarken auf der Bearbeitungsfläche zu erkennen sind, ist der WEX für Schlichtoperationen zu empfehlen.</p>  <p><u>Schnittdaten:</u></p> <p>Fräser: WEX 3032 E (Ø32)     <math>v_c = 200\text{m/min}</math>          Platte: AXMT170508PEER-G     <math>f_z = 0,15\text{mm/Zahn}</math>          Sorte: ACP200     <math>a_e = 5\text{mm}</math>             <math>a_p = 10\text{mm} \times 3\text{Schnitte}</math>             trocken</p>	<p><b>Gesenk, (kohlenstoffarmer Stahl)</b></p> <p>Doppelte Standzeit mit Verschleiß gegenüber dem bisherigen Werkzeug mit Schneidenbruch.</p>  <p><u>Schnittdaten:</u></p> <p>Fräser: WEX 3032 E (Ø32)     <math>v_c = 151\text{m/min}</math>          Platte: AXMT170508PEER-G     <math>f_z = 0,2\text{mm/Zahn}</math>          Sorte: ACP100     <math>a_e = 25\text{mm}</math>             <math>a_p = 5\text{mm}</math>             nass</p>																	
<p><b>Schraubenrotor, X5CrNiMo17122</b></p> <p>Einsatz des WEX bis zu 5 Stunden mit einem relativ geringen Geräuschpegel wurd ohne Ausbröckelungen möglich. Kurze Standzeit des Vergleichswerkzeugs durch Vibrationen mit Ausbröckelungen.</p>  <p><u>Schnittdaten:</u></p> <p>Fräser: WEX 3050 F (Ø50)     <math>v_c = 80\text{m/min}</math>          Platte: AXMT170508PEER-G     <math>f_z = 0,15\text{mm/Zahn}</math>          Sorte: ACP300     <math>a_e = 15\text{--}50\text{mm}</math>             <math>a_p = 4\text{mm}</math>             trocken</p>	<p><b>Gesenk, 55NiCrMoV6</b></p> <p>Geringerer Geräuschpegel, besserer Spanfluss und glattere Oberflächen als beim Vergleichswerkzeug. Auch bei doppelter Vorschubgeschwindigkeit wird eine stabile Zerspannung erreicht.</p>  <p><u>Schnittdaten:</u></p> <p>Fräser: WEX 3080 E (Ø80)     <math>v_c = 75\text{m/min}</math>          Platte: AXMT170508PEER-G     <math>f_z = 0,19\text{mm/Zahn}</math>          Sorte: ACP200     <math>a_e = 80\text{mm}</math>             <math>a_p = 5\text{mm}</math>             trocken</p>																	
<p><b>Maschinenteile, X5CrNi810</b></p> <table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">Standweg: 4500mm</td> <td colspan="3" style="text-align: center;">Standweg: 1650mm</td> <td rowspan="4" style="vertical-align: top; padding-left: 20px;">                 Fräser: WEX 3032 E (Ø32)                  Platte: AXMT170508PEER-E                  Sorte: ACM300   <u>Schnittdaten:</u>  <math>v_c = 150\text{m/min}</math>  <math>f_z = 0,15\text{mm/Zahn}</math>  <math>a_e = 10\text{mm}</math>  <math>a_p = 3,0\text{mm}</math>                  trocken             </td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td style="text-align: center;">WEX</td> <td style="text-align: center;">WEX</td> <td style="text-align: center;">Wettbewerber A</td> <td style="text-align: center;">Wettbewerber B</td> </tr> <tr> <td colspan="4">Das Substrat mit exzellenter Bruchfestigkeit und einer sehr harten Beschichtung garantiert eine hohe Bearbeitungsstabilität.</td> </tr> </table>		Standweg: 4500mm	Standweg: 1650mm			Fräser: WEX 3032 E (Ø32) Platte: AXMT170508PEER-E Sorte: ACM300  <u>Schnittdaten:</u> $v_c = 150\text{m/min}$ $f_z = 0,15\text{mm/Zahn}$ $a_e = 10\text{mm}$ $a_p = 3,0\text{mm}$ trocken					WEX	WEX	Wettbewerber A	Wettbewerber B	Das Substrat mit exzellenter Bruchfestigkeit und einer sehr harten Beschichtung garantiert eine hohe Bearbeitungsstabilität.			
Standweg: 4500mm	Standweg: 1650mm			Fräser: WEX 3032 E (Ø32) Platte: AXMT170508PEER-E Sorte: ACM300  <u>Schnittdaten:</u> $v_c = 150\text{m/min}$ $f_z = 0,15\text{mm/Zahn}$ $a_e = 10\text{mm}$ $a_p = 3,0\text{mm}$ trocken														
																		
WEX	WEX	Wettbewerber A	Wettbewerber B															
Das Substrat mit exzellenter Bruchfestigkeit und einer sehr harten Beschichtung garantiert eine hohe Bearbeitungsstabilität.																		
<p><b>Machine Parts, X5CrNi1810</b></p> <table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">Standweg: 900mm</td> <td colspan="3" style="text-align: center;">Standweg: 600mm</td> <td rowspan="4" style="vertical-align: top; padding-left: 20px;">                 Fräser: WEX 3032 E (Ø32)                  Platte: AXMT170508PEER-E                  Sorte: ACM300   <u>Schnittdaten:</u>  <math>v_c = 150\text{m/min}</math>  <math>f_z = 0,15\text{mm/Zahn}</math>  <math>a_e = 10\text{mm}</math>  <math>a_p = 3,0\text{mm}</math>                  nass             </td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td style="text-align: center;">WEX</td> <td style="text-align: center;">WEX</td> <td style="text-align: center;">Wettbewerber A</td> <td style="text-align: center;">Wettbewerber B</td> </tr> <tr> <td colspan="4">ACM300 zeigt eine sehr hohe Bruchfestigkeit auch bei der Nassbearbeitung, die eine Kammissbildung verursacht.</td> </tr> </table>		Standweg: 900mm	Standweg: 600mm			Fräser: WEX 3032 E (Ø32) Platte: AXMT170508PEER-E Sorte: ACM300  <u>Schnittdaten:</u> $v_c = 150\text{m/min}$ $f_z = 0,15\text{mm/Zahn}$ $a_e = 10\text{mm}$ $a_p = 3,0\text{mm}$ nass					WEX	WEX	Wettbewerber A	Wettbewerber B	ACM300 zeigt eine sehr hohe Bruchfestigkeit auch bei der Nassbearbeitung, die eine Kammissbildung verursacht.			
Standweg: 900mm	Standweg: 600mm			Fräser: WEX 3032 E (Ø32) Platte: AXMT170508PEER-E Sorte: ACM300  <u>Schnittdaten:</u> $v_c = 150\text{m/min}$ $f_z = 0,15\text{mm/Zahn}$ $a_e = 10\text{mm}$ $a_p = 3,0\text{mm}$ nass														
																		
WEX	WEX	Wettbewerber A	Wettbewerber B															
ACM300 zeigt eine sehr hohe Bruchfestigkeit auch bei der Nassbearbeitung, die eine Kammissbildung verursacht.																		



SUMITOMO ELECTRIC Hartmetall GmbH  
Siemensring 84, D - 47877 Willich

Tel. +49(0)2154 4992-0, Fax +49(0)2154 41072, E-mail: [Info@SumitomoTool.com](mailto:Info@SumitomoTool.com) Internet: [www.SumitomoTool.com](http://www.SumitomoTool.com)



Vertretung: