Combined Finite Element and Deep Learning Techniques for Rapid Prediction of Workpiece Structural Dynamics During Turning

by

Meshkat Salehi

A thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

University of Manitoba

Winnipeg

Abstract

Cylindrical parts are widely used in the manufacturing industry, e.g., in making shafts. Such parts are typically machined on lathes by clamping the part on one end and removing material step by step using a cutting tool. The surface quality and dimensional accuracy of machined parts are critical as they directly affect the performance of end products. Structural flexibilities of a cylindrical part clamped on a lathe can lead to unstable vibrations known as chatter. Chatter causes poor surface finish on the part as well as damage to the tool and machine. Chatter can be avoided by proper selection of spindle speed and depth of cut, provided the structural flexibilities of the clamped part are known. However, as a part is machined, we observe changes in the structural dynamics of the part, which means the process parameters, such as spindle speed, must be updated at different stages during machining. This thesis presents a computationally efficient model to update the structural behavior of cylindrical parts during machining. These updated dynamics can then be used to calculate the stability lobe diagrams for each machining step, which leads to optimizing process parameters to maximize manufacturing efficiency.

The proposed model is developed using the finite element (FE) method combined with deep learning techniques. First, a finite element model of the initial part is developed numerically, and the associated system matrices are generated and assembled. The removed volumes during the machining stages are then planned and segmented as substructures of the initial part. These substructures are dynamically decoupled from the workpiece by adding

the opposite of their dynamics to the initial workpiece. This process updates the frequency response function (FRF) of the workpiece.

A complete (full) order finite element model is implemented in the preliminary algorithm. Then, this model is reduced through model order reduction to improve the efficiency of the process by lowering the computational time. To reduce the order of the model, first, the master degrees of freedom (DOF) with the most contribution to the dynamics of the part are selected in an iterative procedure. Different methods are used to reflect the effect of the eliminated slave DOFs on the master DOFs. The FRF prediction results of the reduced order model are investigated.

A deep learning-based framework is proposed to improve prediction efficiency. The developed algorithm in the first part of this study is used to generate a training dataset. A deep neural network is designed using hyperparameter tuning and then trained on the generated dataset. The performance of the trained neural network on an unseen set of workpieces is assessed. It is shown that the developed deep learning model can predict the structural dynamics of the part in less than 0.05s, making it suitable for online process monitoring and control applications.

Table of Contents

List of Figures

List of Tables

List of Symbols

$aDOF$	Internal DOFs of the removed material
A_i	Removed volume at step i of machining
$-A_i$	Fictitious substructure at step i of machining
$bDOF$	Internal DOFs
B_i	Workpiece at step i of machining
$cDOF$	Interface DOFs
$\boldsymbol{C}$	Matrix of coefficients in hook's law
C_{cond}	Condition number
$\boldsymbol{D}$	Dynamic stiffness matrix
$\widetilde{\boldsymbol{D}}$	Condensed dynamic stiffness
E	Elastic modulus
$\boldsymbol{E}_D$	Effective independence distribution vector
$\boldsymbol{EI}$	Effective independence distribution matrix
$\boldsymbol{EV}$	Expected value
$\boldsymbol{f}$	Applied force vector
$\boldsymbol{F}_E$	Product of shape function, eigenvector, and eigenvalue
$\boldsymbol{G}$	Product of shape function and eigenvector
$\boldsymbol{Gn}$	Gaussian white noise
$\boldsymbol{G}_{ns}$	A given non-square matrix

$\boldsymbol{H}$	FRF matrix
$\boldsymbol{H}_H$	Contribution of the high-frequency modes
$\boldsymbol{H}_{H_0}$	Static compensation term
$\boldsymbol{H}_L$	Contribution of the first L low-frequency modes
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{K}$	Stiffness matrix
$\boldsymbol{K}_e$	Elemental stiffness matrix
KE	Kinetic energy
L	Number of low-frequency modes
m	Master DODs
$\boldsymbol{M}$	Mass matrix
$\boldsymbol{M}_e$	Elemental mass matrix
M_{ij}	Elemental mass matrix elements
$\boldsymbol{N}$	Matrix of shape functions
$\boldsymbol{O}_o$	Square of eigenvectors
$\boldsymbol{P}$	Covariance matrix
p	Number of significant modes
$\boldsymbol{q}$	Modal coordinate vector
$\boldsymbol{Q}$	The unitary orthogonal matrix in QR decomposition
$\boldsymbol{r}$	Interface reaction forces

$\boldsymbol{R}$	The upper triangular matrix in QR decomposition
s	Slave DOFs
$\boldsymbol{S}$	Fisher information matrix
$\boldsymbol{S}_e$	Element velocity
t	Time
$\boldsymbol{T}$	Reduction transformation matrix
$\boldsymbol{T}_b$	Boolean matrix
$\boldsymbol{t}_G$	The lower part of the Guyan reduction transformation matrix
$\boldsymbol{T}_G$	Guyan reduction transformation matrix
$\boldsymbol{t}_{IRS}$	The lower part of the IRS reduction transformation matrix
$\boldsymbol{T}_{IRS}$	IRS reduction transformation matrix
$\boldsymbol{T}_L$	The left reduction transformation matrix
$\boldsymbol{T}_R$	The right reduction transformation matrix
$\boldsymbol{u}$	Displacement vector caused by the external force
$\boldsymbol{U}$	Matrix of right eigenvectors
$\boldsymbol{U}_G$	Global displacement vector
V	Element volume
$\boldsymbol{V}$	Matrix of left eigenvectors
x	Position in the local coordinate system in x direction
$\boldsymbol{x}$	Displacement vector
$\ddot{\boldsymbol{x}}$	Acceleration vector

X	Position of a given point in the global coordinate system in X direction
X_i	Position of node i in the global coordinate system in X direction
y	Position in the local coordinate system in y direction
y_i	Ground truth
$\hat{y}_i$	Predicted value
Y	Position of a given point in the global coordinate system in Y direction
Y_i	Position of node i in the global coordinate system in Y direction
z	Position in the local coordinate system in z direction
Z	Position of a given point in the global coordinate system in Z direction
Z_i	Position of node i in the global coordinate system in Z direction
$\boldsymbol{\alpha}$	Modal DOFs
$\boldsymbol{\beta}$	Diagonal matrix of eigenvalues
ε_{Tol}	Convergence tolerance
$\boldsymbol{\theta}^{str}$	Dynamics of a structure
λ	Eigenvalue
$\boldsymbol{\Lambda}$	Matrix of eigenvalues
ϑ	Poisson's ratio
ρ	Density

$\boldsymbol{\Phi}$	Mode shape matrix
$\boldsymbol{\psi}_o^2$	Gaussian white noise variance
ω	Circular frequency

List of Acronyms

ANN	Artificial Neuronal Networks
AIS	Absolute Identification Space
ASA	Angular Synchronous Averaging
CAD	Computer Aided Design
CNN	Convolutional Neural Network
CWT	Continuous Wavelet Transform
DOF	Degrees of Freedom
EEMD	Ensemble Empirical Mode Decomposition
EMA	Experimental Modal Analysis
EVP	Eigen Value Problem
FE	Finite Element
FEA	Finite element analysis
FEM	Finite Element Method
FIM	Fisher Information Matrix
FRF	Frequency Response Function
FSM	Finite Strip Model
GPU	Graphic Processing Units
GTB	Gradient Tree Boosting
IRS	Improved Reduction System

KI	Kernel Interpolation
KNN	K-Nearest Neighbors
LS	Laplacian Score
mae	Mean Absolute Error
mape	Mean Absolute Percentage Error
MDP	Markov Decision Process
ML	Machine Learning
MPE	Multiscale Permutation Entropy
mse	Mean Squared Error
msle	Mean Squared Logarithmic Error
NC	Numerical Control
nD	n-Dimensional
ReLU	Rectified Linear Unit
RF	Random Forest
RFE	Recursive Feature Elimination
SEREP	System Equivalent Reduction and Expansion Process
SLD	Stability Lobe Diagrams
SVM	Support Vector Machine
TL	Transfer Learning
WPT	Wavelet Packet Transform

Acknowledgment

First and foremost, I am deeply indebted to my research supervisor, Dr. Matt Khoshdarredgi, for his mentorship, for helping me overcome academic and personal challenges, and for supporting me throughout my research and coursework by sharing his knowledge and experience.

I would also like to thank my M.Sc. examining committee members, Dr. Ojo and Dr. Luo, for their valuable input and constructive comments on my thesis.

Thanks should also go to the Natural Sciences and Engineering Research Council of Canada and the Canadian Network for Research and Innovation in Machining Technology for financially supporting this research.

I want to acknowledge my friends and fellow students in the Intelligent Digital Manufacturing Lab for making my studies a great experience. Also, I want to thank my friend Maryam, who provided me with ample emotional support.

My most profound appreciation extends to my beloved family for their patience, unconditional love, and support from miles away. Words cannot express my gratitude to my parents, who dedicated their lives to my success. Special thanks to my dear sister, Dr. Reyhaneh Salehi, for her patronage and for always offering me a helping hand since I started this journey. Last but not least, I wish to express my heartfelt gratefulness to my colleague, best friend, and darling husband, Ali Maghami, who helped and encouraged me in every step of my studies. This endeavor would not have been possible without Ali's and my family's absolute confidence in me.

To Ali and my family …

1 Introduction

Machining is one of the most common operations in manufacturing, and its advancements can be counted as the roots of manufacturing revolution. The high quality of finished parts makes this type of process desirable for industrial mass production of metal parts. To improve production efficiency, there is always a need to increase the machining speed, especially for parts that require many material removal steps, as the cost of production is directly associated with the consumed time.

One phenomenon affecting the workpiece during the machining process is chatter or machining vibrations. In turning, chatter happens when the rotation speed of the workpiece is close to one of the natural frequencies of the workpiece. In the case of severe chatter, surface marks on the part can violate the required tolerance, which means the part must be scrapped. Hence, chatter prediction and avoidance are among the most critical aspects of process planning in machining.

Cylinders are one of the vastly used geometries in industrial parts, such as pistons or oil pipes. As an axisymmetric geometry, cylindrical parts are manufactured by lathe machines using operations such as axial turning, boring, facing, etc. Once chatter occurs during an operation, the generated wave marks on the workpiece surface induce more vibrations in the following steps, which leads to poor finish quality. This phenomenon is called “Regenerative Chatter.” Thereby, simulation and modeling of these processes to predict the dynamic characteristics of the workpiece and the possibility of chatter are crucial

for efficiency improvement in industries. However, the changes in structural dynamics of the workpiece caused by material removal make the chatter prediction challenging. When the geometry of the workpiece varies due to the machining process, the part's natural frequency and frequency response function also change, which leads to an alteration in the stability of the machining toolpath. As a result, we have to model the dynamics of the part and include it in the chatter prediction model.

The conventional method in industry for finding the machining parameters is based on empirical knowledge, usually obtained through trial and error. On the hand, finite element software packages can be used to predict the structural dynamics of a part. However, to consider the structural parameters' variations during machining, many sets of simulations must be performed to predict the part behavior at different stages during machining. This process can become very time-consuming using commercial full order finite element software packages. The main purpose of this thesis is to develop a reduced FE model combined with a deep learning model to allow for the rapid prediction of structural dynamics of cylindrical parts with various geometries. The implemented model is used on cylindrical geometries under the turning process. Still, it can also be used in other procedures, such as facing. Figure 1-1 shows a schematic of the complete scope of the conducted research in this dissertation.

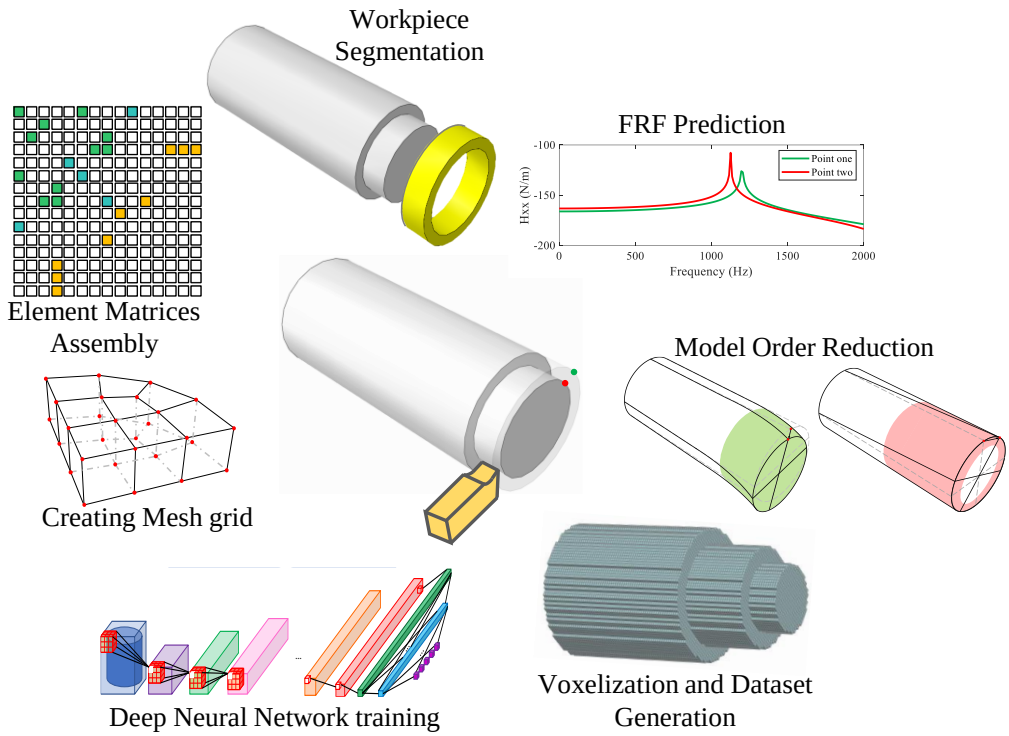


Figure 1-1 A schematic outline of the thesis project

The finite element method has been utilized as an offline approach to foresee the part's structural dynamics and plan the process. Unlike the techniques that use several FE models at different stages of the machining process, a semi-analytical model in the frequency domain is implemented that only uses the initial workpiece's FE model and divides it into various segments. In this method, the workpiece's geometry and mesh grid are consistent throughout the cutting process. Each of these segmentations represents one step of material removal. The physical system matrices of the initial part and each of these segments are obtained using the generated FE mesh grid and the material properties. The implemented model only updates the dynamics of the workpiece by adding the opposite of the removed

volume's dynamics to the initial part. The updated dynamics are utilized to estimate the vibrations of the workpiece and predict the chatter stability during machining operations.

Modeling parts with curved edges and surfaces always introduces inaccuracy due to the approximation with prismatic elements. This is known as the "Babuska Paradox" [1]–[3]. To achieve an acceptable result that provides sufficient accuracy, it is beneficial to use an element size that can minimize the difference between the FE model and the actual part and reflect different machining stages, such as roughing and finishing. Smaller element size gives more accuracy; concurrently, the increased number of nodes results in the multiplication of modeling data size. Using these data in the calculations requires extensive computational power. Thus, having a method to alleviate the computational load with minimum compromise on accuracy is desired. In this thesis, two methods are used to increase the computational speed. First, a model order reduction is conducted. It means that those degrees of freedom in the FE model that contribute most to the model's dynamics are selected by an iterative process, and the contribution of the rest of the workpiece to the dynamics of the model is reflected on the chosen set of DOFs. This method, which is applied on both the removed material and the in-process workpiece, combines several reduction processes due to the different restrictions on each set of nodes. The second approach for reducing the number of required calculations is using a modal truncation method for computing the FRF to avoid the inversion calculation of large matrices.

With the expansion of intelligent and automated manufacturing and the offered flexibility, making specific process planning and reparatory decisions in real-time with minimal to no human intervention is possible. For instance, the depth of cut or the feed rate

might need to be adjusted during machining. Even the toolpath can be changed based on feedback from a real-time monitoring system. In these cases, updating the structural model rapidly and recalculating the optimum machining parameters during the process is vital. With the revolution in the field of Graphic Processing Units (GPU) in the past few years, the use of deep learning based solutions as fully automated methods is on the rise. In this thesis, a novel deep learning-based framework is developed to predict the structural dynamics of the workpiece with minimum time consumption. In this method, the developed FE model is used to generate a training dataset, which is processed and converted to digital format to be suitable as the input of a deep learning algorithm. Also, the dynamic parameters of each geometry are stored as the output. A convolutional neural network learns the variations of structural dynamics of the workpiece in association with its geometry using the generated dataset.

In the rest of this thesis, the related literature is reviewed in Chapter 2, and the developed FE model and dynamics updating technique are provided in Chapter 3. In Chapter 4, all the approaches for reducing the computational load and increasing the speed of calculations are explained in detail. The automated algorithm based on the deep learning method is demonstrated in Chapter 5. Possible future research areas and approaches are suggested, and the thesis is concluded in Chapter 6.

2 Literature Review

2.1 Overview

This chapter reviews the literature relevant to the prediction of structural dynamics and machining chatter. Section 2.2 presents literature related to modeling workpiece characteristics under machining, machining dynamics, and chatter stability. Section 2.3 reviews the relevant literature for machine learning-based characterization and process planning. The chapter is concluded in Section 2.4.

2.2 Varying Workpiece Characteristics and Machining Processes Stability

Removing material from the in-process workpiece results in a change in the mass and stiffness matrices of the part. It means the compliance of the part increases as the material is removed during the machining process. The two conventional methods to obtain the structural properties of the workpiece at different removal steps are the full-order FE model and Experimental Modal Analysis (EMA).

Eksioglu [4] used the EMA method to predict chatter stability and surface error of a rectangular plate under the peripheral milling process. Ismail and Ziaei [5] used this method for vibration suppression in a five-axis machining process of turbine blades made of flexible material at different tool positions with relatively large distances. These studies did not repeat EMA for different steps of material removal; as a result, time was not considered a relevant

parameter in the process. Although there are studies [6]–[8] on the vibrations of thin webs, measuring the FRF of this type of structure during the material removal is impractical and cumbrous.

On the other hand, utilizing several FE models for various lengths of cut or thicknesses, researchers studied the variation in the dynamic compliance during straight cylindrical turning for slender parts [9], [10] and thin-wall tubular parts [11]–[14]. Stepan et al. [9] and Urbikain et al. [10] used model order reduction for the developed FE model and obtained the dynamic parameters along the toolpath. Although some researchers [13] assumed that the cutting process does not affect the vibration mode frequencies, despite the significant discrepancies in the chatter frequencies in experiments and simulations, it was experimentally proven by others [11] that variation in the dynamics of the part during material removal causes changes to chatter stability conditions. To overcome this challenge and reduce chatter vibrations, Lorong et al. [13] proposed regulating the spindle speed. However, generating multiple FE models is excessively time-consuming, making it infeasible to be used in models developed for digital machining processes.

In similar research by Wan et al. [15], the element stiffness is modified to reflect the effect of volume reduction on the part's dynamic and static stiffness, consistent with the change in their volume. In other research works [16]–[20] the workpieces' geometries are updated, and new mesh grids are generated for the FE model. The main purpose of these research works [15]–[20] is to predict the form errors and deflections induced by the cutting forces in the peripheral milling process on the tubular and rectangular thin-walled geometries. Koike et al. [21] took the effect of removing volume from a machined titanium

alloy cantilevered beam into account and added it to the initial part before the machining process. In their research, the authors propelled the cutting forces to the directions with higher stiffness. By this means, they minimized the static deflection of the workpiece under milling operation.

Both full-order models [22]–[28] and statically condensed models [29], [30] are used to investigate chatter vibrations, stability, and the quality of the machined surface. These studies are conducted on the rectangular thin-walled workpiece under the peripheral milling process [22]–[30]. Similar studies are carried out on machining thin-floor workpieces using bull-nose cutters [31], [32]. The interpolation of dynamic characteristics of the workpiece in intermediate stages along the tool path is used to consider the effect of the changes in natural frequencies. However, since the removed volume in the finishing operation is very small, some researchers [25], [27], [29], [30], [32] disregard the changes in the vibration characteristics of the part and consider these characteristics as uniform thorough out the process. For predicting the stability, only the effective modes in the cutting region were contemplated, and the rest of the identified global modes were ignored [23], [28], [29]. In order to place emphasis on the influence of workpiece dynamics alteration on predicting chatter, Bravo et al. [23] modified the typical stability lobe diagrams. They added the step number of material removal as the third axis, while others [28], [29], [31], [32] used the tool position as the extra parameter. As a method of avoiding chatter in the cutting region along the toolpath and reducing undesired vibration caused by alteration of the dynamic parameters of the part during the milling process, spindle speed controlling and variating was recommended [26], [28], [29], [31]. As an alternative, to avoid changing the cutting

condition while the machining process is ongoing, the predicted stability lobes were superimposed [32] to obtain the most stable cutting region applicable throughout the whole tool path, even though it is not practically possible in common cases.

Kersting and Biermann [33] used three numerical methods to update the dynamics of the changing thin-wall parts and predict the deflection of flexible aerospace parts during the NC milling process. Evaluating these methods shows that since the oscillator-based method also needs modal parameters at different machining steps, it does not have any advantages over the EMA method in terms of time prohibition. In addition, it is shown that the mesh grid in the particle-based and FE models must be modified repeatedly along the path. As a result, these methods can be used on a small section of the workpiece in processes with minimal material removal, such as finishing.

Meshreki et al. [34] reduced a three-dimensional thin-walled pocket to a two-dimensional multispan plate to take the influence of continuous thickness variation during milling operations on both sides of rectangular pockets on the dynamics of the workpiece into account and predict the part vibration. Ahmadi [35] also used the finite strip model (FSM) and aerospace parts re-meshing to model the thin-wall workpiece's structural dynamics under the pocket milling process. Their research predicted the modes and mode shapes of the part without using repetitive computationally heavy full-order FE analyses and the limiting EMA. In both studies [34], [35], the algorithms are developed for plate structures with rectangular geometries.

Song and colleagues [36] obtain the inverse of the modified physical system matrices by implementing the Sherman Morrison-Woodbury formulation. In their study, at each step

of the material removal of the machining process, damping was considered identical, and the in-process workpiece was modified consistently. To employ this method, the dynamic characteristics of the workpiece before and after machining must be provided. The predetermined material removal steps directly influence this method's accuracy. Budak et al. [37] combined FE analysis with structural dynamics modification to investigate the surface of an aerodynamic workpiece with free-form features. In their study, FE analysis is used to obtain the FRF of the final part after machining. Following a reverse approach, the obtained FRF is modified by adding the removed material in a backward manner at several points of the toolpath until the original unmachined part is reconstructed. In order to use this method, information on the removed material is required, and it must be acquired by geometric simulation of the cutting process. Yang et al. [38] investigated curved thin-walled parts with rectangular geometries. They updated the parts' natural frequencies and mode shapes in the modal domain and with modal truncation. In their study, the dynamics of the part are updated with respect to the original workpiece, and material removal is defined as one removed volume to reduce the error propagation. However, the mass and stiffness matrices must be assembled repeatedly along the toolpath in this method. The results of this process are relatively accurate for a limited number of cutting steps with small removed volumes, but the performance deteriorates afterward.

Khoshdarregi and Altintas [39], [40] investigated the dynamics of multipoint thread turning. They assessed the influence of vibrations and vibration marks on the surface of the workpiece and modeled the chip regeneration mechanism [39]. This model is used to obtain the process damping and dynamic cutting forces at the cutting edge by projecting the tool's

three-dimensional (3D) vibrations and workpiece in the direction of local chip thickness. They obtained the equation of motion in physical and modal spaces and used the Nyquist criterion to analyze the stability in the frequency domain. They extended the developed formulation for thin-walled cylindrical shell structures, focusing on application to thin-walled oil pipes [40]. They investigated shell deformations of a discretized surface caused by the cutting forces. The response of the workpiece under threading is obtained by solving the dynamic equations of motion. After obtaining the vibrations around the circumference based on the dominant circumferential shell modes, it is shown that even in a stable process, residual shell vibrations are still observed. Using damper and different spindle speeds are demonstrated for chatter suppression.

Budak and Ozturk [41] formulated the dynamics of the turning process on a surface using two different tools. To focus on the effects of the parallel turning, they presented a one-dimensional analysis of chatter in the frequency domain. The tools were set up on two different turrets and cut the same surface with different depths of cut; as a result, they modeled the tools as attached to the rigid body of the machine with two springs and dampers. Compared to the tools, they ignored the workpiece's dynamics and considered it a rigid part. A time-domain model was also developed to verify the stability limit predictions made by the frequency-domain model. They showed that the dynamic interaction between the tools improves the stability limits in comparison to the single-tool turning process, especially when the modes are very close. The simulation results of their study were validated through cutting tests. In a later study, Ozturk et al. [42] investigated two tool configurations in parallel turning operation. In the first case, where the tools were mounted on the turret, the tools'

dynamic displacements interacted through the turret. The stability limit was observed to decrease compared to the single tool case, but the removal rate is higher due to having an extra tool. In the second configuration, similar to [41], removing material from a shared surface was investigated, and they showed that cutting tools with the same natural frequencies have a lower stability limit. Reith et al. [43] conducted similar research and reported similar behavior. Their study employed a nonproportional damping model to consider the tool holder's effect. Investigating the effect of tools' radial orientation showed that it is possible to increase the stability limit by using a proper orientation for the tool [44]. For flexible parts, Azvar and Budak [45] demonstrated that the true geometry of the insert must be modeled since the cutting forces are dominant in radial directions. It was observed that the stability limits are higher when using the same insert geometry while machining a shared surface. On the other hand, in the case of turning operation on different surfaces, a sharper side-edge cutting angle and nose radius are more beneficial in terms of stability.

Tuysuz and Altintas [46] divided a thin-wall workpiece with rectangular geometry into substructures and used the model to update the part's frequency response function during the milling process. An imaginary mass with opposite characteristics is assigned as the removed material, and an order reduction method is applied. Then, the FRF is determined by solving the equation of motion in the frequency domain as the workpiece is machined. The obtained FRF through updating procedure is used to predict chatter. They also used perturbation methods to update dynamic parameters [47] and obtain the in-process workpiece's mode shapes. A vector sequence convergence accelerating algorithm is

integrated to improve the perturbation convergence. They verified their model using a thin-walled fan blade under five-axis ball-end milling.

In past research, both EMA and full-order FE models have been used extensively to consider the dynamics change of the part. These approaches require multiple modeling steps resulting in a higher computational time. Predicting the part's dynamics in the process planning stage is preferable. In this study, an efficient model is developed to update the workpiece dynamics using substructuring and order reduction methods. Using this algorithm, we can obtain the updated FRF of the workpiece in the machined surface areas.

2.3 Machine Learning Applications in Machining Processes

The revolution in computer hardware in recent years has broadened the industrial applications of machine learning algorithms. The new generation of intelligent manufacturing machines is expected to make fast corrective decisions and update process parameters during operation. Using machine learning and deep learning algorithms, we can eliminate the need to run FE models during a machining process for updating parameters. The use of deep learning by neural networks is expanding in different machining applications, such as condition monitoring [48], [49], inspection [50], and feature recognition [51].

A conventional approach in chatter prediction is using Stability Lobe Diagrams (SLD). These diagrams evaluate the process parameters to predict undesirable vibrations. In this method, the data obtained from analytical models are labeled by SLD, so machine learning can provide new solutions for online chatter recognition.

Denkena et al. [52] trained different models such as Support Vector Machine (SVM), Artificial Neuronal Networks (ANN), and a newly introduced Kernel Interpolation (KI) algorithm by the process parameters to predict SLD. The research in [52] was conducted using simulated data and real process measurements from a five-axis milling machine with a four-teeth tool with a 10 mm diameter. The training data were labeled using a microphone and acceleration sensors. They showed that Machine Learning (ML)-based generated SLD are in good agreement with the analytically calculated results. Since all the models achieved an accuracy greater than 88%, they can be used as an alternative with minimum compromise. Particularly, the KI, as a flexible and comprehensive model with an accuracy of 94%, was highly recommended. However, this method was restricted by data availability when the condition changed. Postel and colleagues [53] suggest a novel Transfer Learning (TL) and deep neural network-based method to refine the stability limits in milling operations to overcome this challenge. The neural networks in their research were pre-trained on simulated data generated by analytical stability models. The main purpose of their study was to alleviate the disparities between the experimental measurements under different cutting conditions and the model output caused by uncertainties in the input parameters and models' imperfections. The researchers used spindle speed, entry and exit angle, tool clamping length, and depth of cut as inputs. By employing ensemble learning and combining the output of multiple networks, they reached an improved accuracy of 83.6% for the test set.

TL with Random Forest (RF) for chatter detection in turning was investigated by Yesilli et al. [54]. They used two feature extraction methods, Ensemble Empirical Mode Decomposition (EEMD) and Wavelet Packet Transform (WPT). These two methods were

tried on time series obtained from an acceleration sensor mounted on the lathe tool holder. Three categories for data labeling were considered: chatter, mild chatter, and no chatter. Recursive Feature Elimination (RFE) was combined with SVM, logistic regression, RF classification, and gradient boosting to classify different machining states into one of the aforementioned categories. The results were used to evaluate the performance of the applied feature extraction methods. It was shown that EEMD has a better performance compared to WPT. In another study, Yesilli et al. [55] investigated Carlsson coordinates and template functions, which are two feature extraction methods. They used these methods on simulated vibration data of a single-DOF milling tool. They simulated different cutting configurations, such as up-milling and down-milling, and added high noise to the obtained data. The feature extraction methods were combined with supervised ML algorithms such as SVM, RF, gradient boosting, and logistic regression. Based on the reported results, the template function method outperforms Carlsson coordinates in most cases. Vibration signals were used as input in combination with SVM [56] and ANN [57] for two-class chatter classification.

Although accelerometers are the most common tool for chatter detection, Kvinevskiy et al. [58] used sound sensors for chatter recognition. In [58], an SVM, an autoencoder to reduce the dimension, along with a Short-Time Fourier Transform (STFT), are used to generate a binary classifier and develop an algorithm for chatter identification without manual input. Shi and colleagues [59] also combined vibration and sound signals obtained from several experiments on a CNC machine during high-speed milling operations and used a t-distributed stochastic neighbor embedding method for feature extraction and

visualization. They employed a reinforced K-Nearest Neighbors (KNN) method for chatter identification.

Using force signals for chatter recognition is covered in different studies [60], [61]. Tran et al. [60] employed Continuous Wavelet Transform (CWT) to convert 1D cutting force signals measured in different cutting conditions during the milling process into 2D scalogram images. These images are fed into a Convolutional Neural Network (CNN) as the training data to classify cutting states into stable, transitive, or unstable. Their results showed that CNN performance is much higher than the common ML techniques. Li and colleagues [61] classified chatter in milling operations using cutting force signals. Their research used Angular Synchronous Averaging (ASA) to extract the features. They characterized the chatter-related signal's properties using Multiscale Permutation Entropy (MPE) and Multiscale power spectral entropy (MPSE). To finalize the selection of optimal sensitive scale features, Laplacian Score (LS) is exploited, and Gradient Tree Boosting (GTB) is used to detect and classify the chatter severity.

ML and deep learning methods can be used to model and predict the behavior of the machining tool or structural parts. An ANN scheme for modeling the cutting forces was developed by Vaishnav et al. [62]. They investigated the rotation angle and end milling process parameters and used a mechanistic force model to generate a dataset to train a supervised neural network. Zhang et al. [63] implemented a similar structure in their research on robotic constant-force grinding control. They developed a force control algorithm based on model-based reinforcement learning. The developed model-based reinforcement learning was used to find the optimal process parameters. Reinforcement learning is also used to

optimize clamping position in the milling process [64] such that the acceleration and traveled distance were minimum. The problem was formalized as a Markov Decision Process (MDP), and it was shown that the developed algorithm is able to find the optimum position for clamping in unseen geometries.

In past studies, ML methods were mainly used to develop a model based on experimental data. Although the experimentally gathered datasets contain valuable information, generating a large dataset that covers various machining scenarios through experiments is neither cost-effective nor possible. In addition, predicting the possibility of chatter in real-time or even ahead of time is advantageous for process planning. It also reduces the number of required restorative changes during manufacturing operations. This thesis proposes a 3D-CNN framework based on the developed FE model. In this framework, the developed FE model is used to generate an extensive training dataset. The trained CNN is used to predict the natural frequencies of the in-process workpiece at different machining stages.

2.4 Conclusion

Past studies with related topics to the field of this thesis can be improved in terms of efficiency and computational load. Although there are proposed solutions for other geometries under different machining processes, there is no study for efficient and optimized simulation of cylindrical workpieces under turning processes. This thesis uses the FEM to develop a semi-analytical model for rapid prediction of the workpiece's dynamics. In

addition, the novel machin learning techniques are combined with the FE method to further improve the prediction speed.

3 FRF Updating with Finite Element Method

3.1 Overview

In order to solve a problem with the Finite Element Method (FEM), the system must be divided into smaller parts with simpler geometries and characteristics. These subdivisions are called finite elements. This chapter provides a brief overview of the procedures to create an FE model for a cylindrical workpiece under machining and updating its varying dynamics. Section 3.2 explains the steps for setting up the model, discretization, assembly, and retrieving the required data. The FE model is then used in Section 3.3 to update the dynamics of the workpiece during the machining process. The obtained results are provided in Section 3.4, and the chapter is concluded in Section 3.5.

3.2 Setting Up the Finite Element Model

In the proposed solution for predicting the FRF for the in-process workpiece, the first step is to create a finite element model of the part. The geometry of a solid part can be specified using a finite element mesh. This mesh grid can also show the internal displacements of the part. Usually, the whole part must be grided, and then the forces must be applied. However, when both the part's geometry and the applied forces are symmetric, we can use axisymmetric mesh. The symmetry of the loads and the geometry must be around the same axis. Since the loads, and as a result, the vibrations and displacements, are not

symmetric in the turning process of cylinders, we do not use an axisymmetric model in this study.

3.2.1 Nodes and Elements

Sets of nodes forming elements determine a finite element mesh grid. The organization of these nodes and their specific places in the part are defined based on the selected element type and the part's geometrical features. As a result, to develop a finite element model for the workpiece, first, we need to select the element type and the conditions for the nodes. A possible mesh pattern of a cylindrical workpiece is shown in Figure 3-1. Since the shape of the part and the mesh grid are symmetric, this figure only represents a quarter of the mesh cross-section.

Each node can have multiple degrees of freedom. In general, the total number of unknown quantities at each node in which we are interested are considered as the degrees of freedom. In this study, we are interested in the vibration of the part; as a result, three translational degrees of freedom are considered for each node.

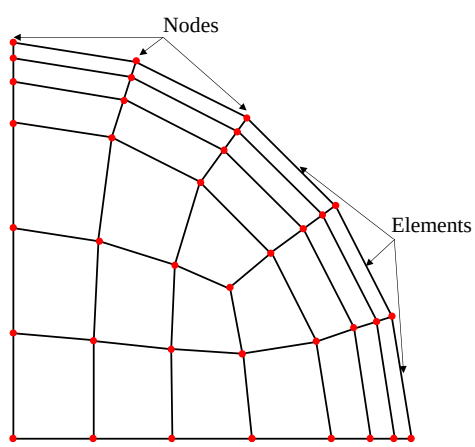


Figure 3-1 A quarter of the cylindrical mesh grid cross-section

For modeling a solid part, there are many ways to discretize and represent the part. In this study, hexahedron elements are used as the element type. In the developed model for this study, elements are formed by eight nodes and have six faces, as shown in Figure 3-2.

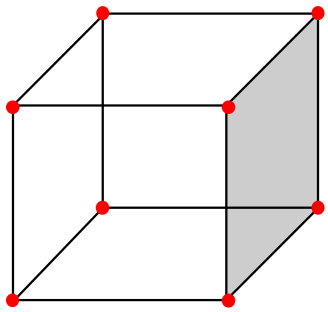


Figure 3-2 Hexahedron element

In order to keep the information about the connection between nodes, the nodes which create an element are stored in a matrix called the "Element Connectivity Matrix." The order of listing the nodes of each element in the element connectivity matrix is counter-clockwise from the left corner of the bottom face, followed by the same order for the top face, as shown in Figure 3-3.

In an FE model, there are two types of coordinate systems. First is the local (natural) coordinate system, which is defined for each element, and its center is aligned with the element's center. The second one is the global coordinate system which can be defined anywhere on the model [65]. These coordinate systems are shown in Figure 3-4.

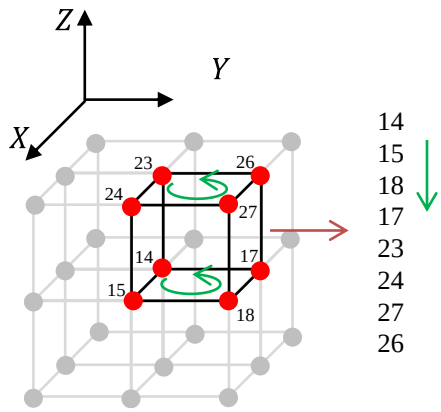


Figure 3-3 Order of nodes in an element connectivity matrix

In each element, the nodes' coordinates are integer numbers in the range of $[-1,1]$ for each direction. These coordinate systems are important in deriving the required equations for calculating mass and stiffness matrices.

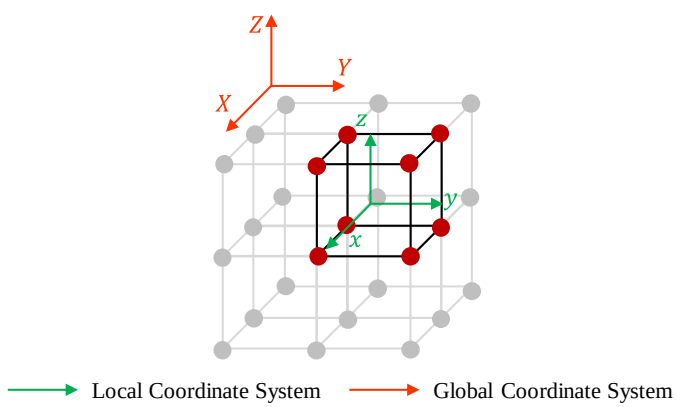


Figure 3-4 Local and global coordinate systems

3.2.2 Generating Element Mass and Stiffness Matrices

To study the dynamic behavior of our workpiece, we need to calculate its mass and stiffness, also known as system matrices, according to its developed finite element model.

The stiffness matrix for an 8-node brick element in global coordinates can be calculated from [66]

$$\boldsymbol{K}_e = \int_V \boldsymbol{B}^T \boldsymbol{C} \boldsymbol{B} dV. \tag{3.1}$$

This matrix can also be calculated with respect to the local coordinates as

$$\boldsymbol{K}_e = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \boldsymbol{B}^T \boldsymbol{C} \boldsymbol{B} |\boldsymbol{J}| \, dxdydz. \tag{3.2}$$

Here $\boldsymbol{C}$ is the matrix of coefficients [67], $\boldsymbol{B}$ is the strain matrix [66], and $\boldsymbol{J}$ is the Jacobian matrix [68].

For this element type, the element mass matrix is [65]

$$\boldsymbol{M}_e = \int_V \rho \boldsymbol{N}^T \boldsymbol{N} dV, \tag{3.3}$$

which can be written using natural coordinates as

$$\boldsymbol{M}_e = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \rho \boldsymbol{N}^T \boldsymbol{N} |\boldsymbol{J}| \, dxdydz \tag{3.4}$$

Here, ρ is the material density, and $\boldsymbol{N}$ is the matrix of shape functions [66].

The resulting matrix from this formulation is called a consistent mass matrix. However, In many finite element literatures, using a lumped mass matrix instead of a consistent mass matrix is more customary.

There are three different methods for lumping the mass matrix: row sum method, diagonal scaling, and evaluation of the mass matrix using a quadrature [66]. In this study, the row sum method is used in the developed algorithm for lumping the mass matrix. Lumping the mass matrix results in a diagonal matrix in which the diagonal elements are

$$M_{ii,lump} = \sum_{j} M_{ij}. \tag{3.5}$$

Using a lumped mass matrix has computational advantages, especially in problems where inversions of the mass matrix are involved.

3.2.3 Element Assembly

After calculating mass and stiffness matrices for each element, we need to assemble them into global mass and stiffness matrices representing the mass and stiffness of the whole part. The reason for this is that neighboring elements always share some nodes. For example, in an FE model consisting of brick elements, the nodes located on the edges of a part are always mutual between at least two elements, and the internal nodes of a part can contribute to forming four elements. As a result, the contribution of element matrices must be assembled in a global matrix.

Since we used eight-node elements with three DOF for each node, element mass and stiffness matrices have a size of 24×24. The global mass and stiffness matrices follow a similar rule on a bigger scale. It means that in a FE model for a part with n nodes, the system matrices have a size of $3n \times 3n$.

In order to assemble global matrices for a part, the assigned numbers to the nodes forming each given element are extracted from the connectivity matrix. These numbers also give the corresponding DOF numbers, hence their position in the global matrices. Then, the corresponding values to each DOF from element mass and stiffness are added to the equivalent position in the global matrices. For the nodes contributing to more than one element, the final value in the global matrices is a summation of shares of that specific node in each element. A schematic of this process is shown in Figure 3-5.

Unlike the element mass and stiffness matrices with completely non-zero elements, the global mass and stiffness matrices have many zero entries. However, at least 24 non-zero elements are in each row and column.

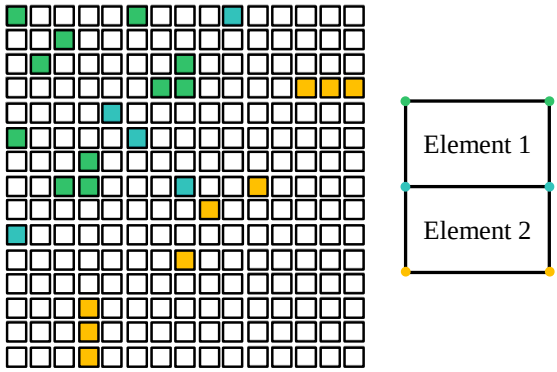


Figure 3-5 Schematics of the global matrix assembly process

3.2.4 Boundary Condition

To generate an accurate FEA model, we must impose boundary conditions, or prescribed displacements, to specific nodes in the generated mesh grid. We can only specify the applied load to a solid part after defining boundary conditions.

Here, we are studying a cylinder under machining. It means one end of the cylinder is fully clamped in the lathe's chuck. The developed model uses the displacement boundary condition to simulate this situation.

As we know, it is possible to specify the displacements of any node located on the part's surface or inside it. These displacements can be prescribed in either one or all directions, depending on the type of constraints on those nodes. In a fully clamped boundary condition, all node displacements are prescribed and set equal to zero. In order to reflect these changes in the developed model, the corresponding rows and columns to the prescribed DOF in the global mass and stiffness matrices are eliminated.

3.3 Modeling of Varying Dynamics of the Workpiece

In a machining operation, as the material is removed from the part, the structural dynamics of the part change, as shown in Figure 3-6. The developed model in this study uses the FE discretization on the original part, consisting of nodes and elements, to predict these changes in advance.

The physical system matrices for the part calculated in the previous section based on the discretized model are used in this algorithm. In order to find the dominant range of frequencies, the Eigen Value Problem (EVP) for the initial part is solved, and the required range is selected.

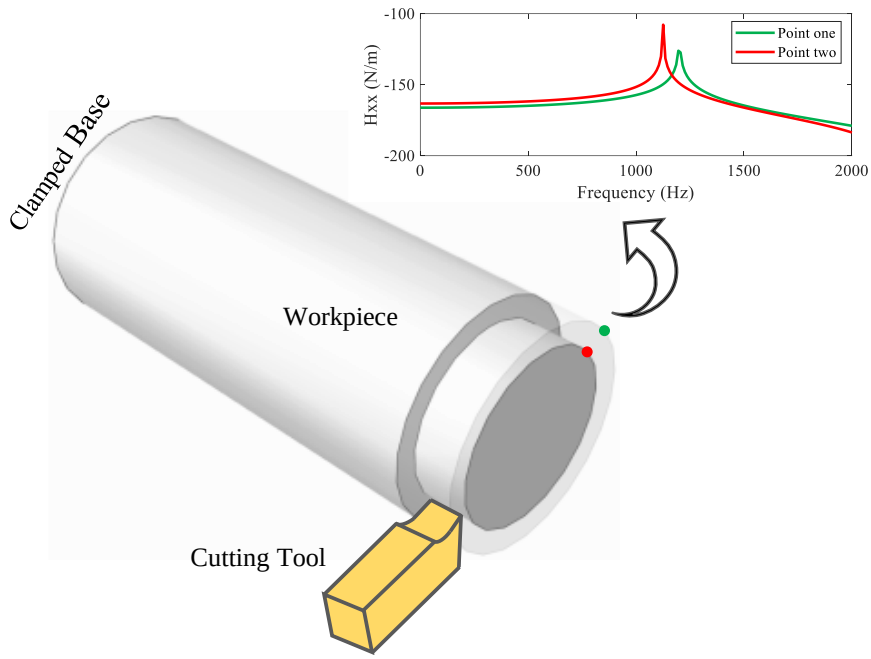


Figure 3-6 Variation of the in-process workpiece dynamics

As can be seen in Figure 3-7, the material removal between two consequent steps, $(i-1)^{th}$ and $(i)^{th}$, can be defined as a mass reduction of the workpiece and an updated geometry. This method is also standard for machining operations validation.

In the classical method for updating the dynamics in machining, the associated dynamics of the in-process workpiece are obtained by calculating the eigenvalues of the full-order FE model. For this process, the stiffness and mass matrices of the workpiece at several cutting steps are obtained and used.

Contrary to the classical method, the proposed solution in this research cancels the removed material's contribution to the workpiece's dynamics. However, the geometry of the workpiece remains the same. This process uses negative structural modification [69], [70].

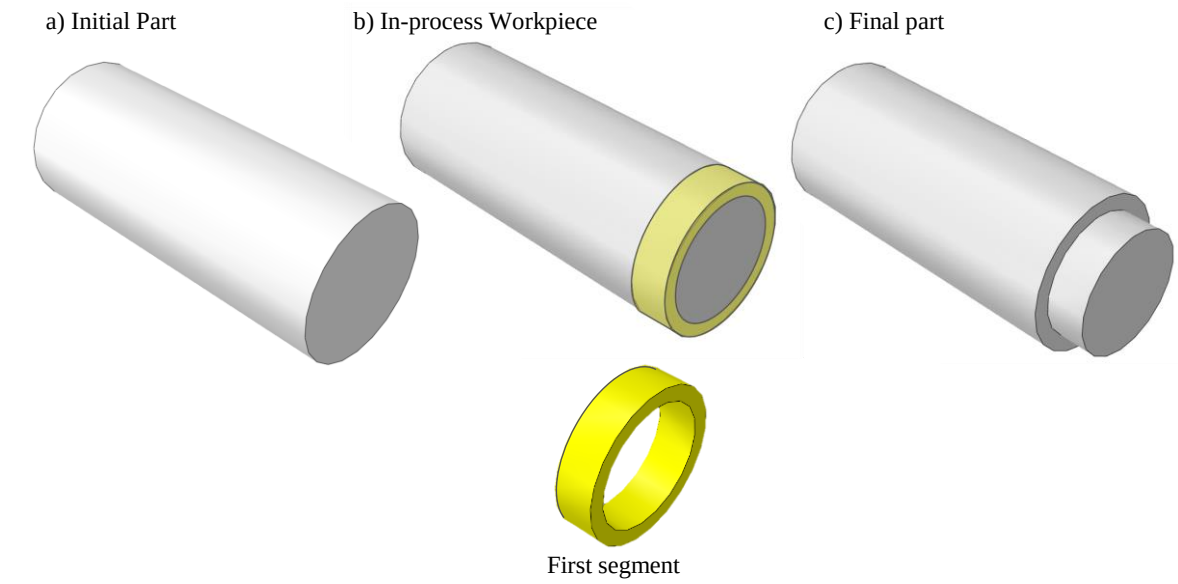


Figure 3-7 Workpiece segmentation between two consequent machining steps

In order to cancel the dynamics of the removed material, an imaginary substructure with the exact geometry is considered. The dynamics of this substructure are the opposite of the removed volume. By adding the dynamics of this fictitious substructure to the initial part, the material removal is converted to a coupling inquiry, shown in Figure 3-8, and the removed volume's contribution to the dynamics of the workpiece is canceled.

This process returns the same results as solving the updated dynamics of the machined workpiece. The mathematical expression of this process is

$$\boldsymbol{\theta}^{B_i}(t) = \boldsymbol{\theta}^{B_{i-1}}(t) + [-\boldsymbol{\theta}^{A_i}(t)], \tag{3.6}$$

in which t represents time, $\boldsymbol{\theta}^{B_i}(t)$ and $\boldsymbol{\theta}^{B_{i-1}}(t)$ show time-dependent dynamics at two consecutive cutting steps, and $-\boldsymbol{\theta}^{A_i}(t)$ is the dynamics of the removed volume in the

opposite direction. The removed material between two steps of machining (A_i) is considered a substructure of the workpiece.

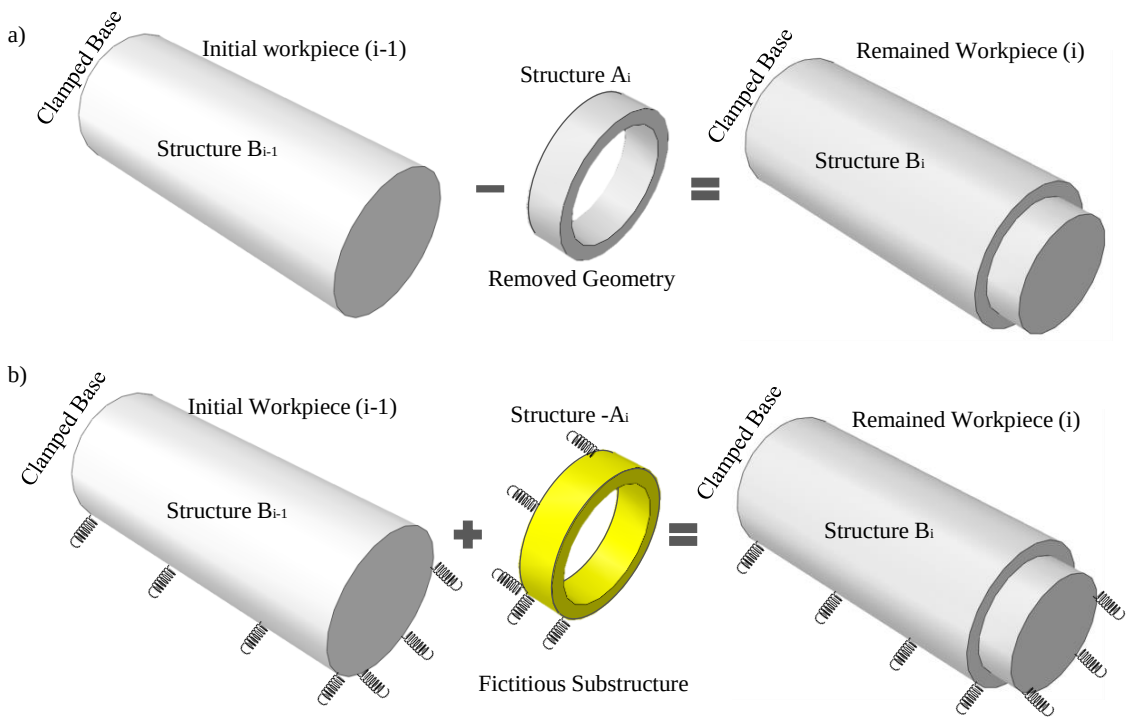


Figure 3-8 (a) Geometrically and (b) dynamically removed segments

As shown in Figure 3-9 (a), the nodes of the workpiece are divided into three categories: internal nodes of the removed volume, interface nodes, and internal nodes of the workpiece. Here, we use a three-dimensional model, which means three translational DOFs are assigned to each node. The DOF correlated with the node sets are divided into three categories as well: $aDOF$ for the internal DOFs of the removed material, $cDOF$ for the coupling DOFs which are on the interface, and $bDOF$ for the internal DOFs of the workpiece.

We can show the interactions between the interface DOFs on the workpiece and the substructure by rigid links, as shown in Figure 3-9 (b), which satisfies the displacement compatibility for interface DOFs and the force equilibrium condition. These links are replaced by interface reaction forces represented by $\boldsymbol{r}_{str1}^{str2}$ [71]. This force vectors show the force that substructure two feels because it is connected to substructure one, as shown in Figure 3-9 (c).

As discussed before, the added fictitious substructure for the dynamic decoupling process is shown in Figure 3-9 (d). Assuming this substructure with negative dynamics is added to the initial workpiece, it gives enough conditions for simulating material removal, which causes dynamics removal. It means the dynamic contribution of the removed volume (A_i) to the initial workpiece (B_{i-1}) is eliminated by the opposite dynamic stiffness of the fictitious substructure. Also, the interaction forces between the workpiece and the removed material are canceled by the added substructure's opposing coupling forces.

A standard way to express the structure dynamics, which we also use in this study, is receptance FRF, which can be obtained by solving the EVP as

$$\boldsymbol{K}^{B_0}\boldsymbol{\phi}_{j_m}^{B_0} = \lambda_{j_m}^{B_0}\boldsymbol{M}^{B_0}\boldsymbol{\phi}_{j_m}^{B_0}. \tag{3.7}$$

In this equation, $\boldsymbol{K}^{B_0}$ and $\boldsymbol{M}^{B_0}$ are the mass and stiffness matrices of the initial workpiece, $\boldsymbol{\phi}_{j_m}^{B_0}$ is the eigenvector, and $\lambda_{j_m}^{B_0}$ is the eigenvalue of the initial workpiece. The implemented mass and stiffness in this equation are computed from the FE model, which was developed previously. This step is performed only once at the beginning of the machining process. For

further steps after the material removal, Eq. (3.6) is used to update the receptance FRF of the machined workpiece.

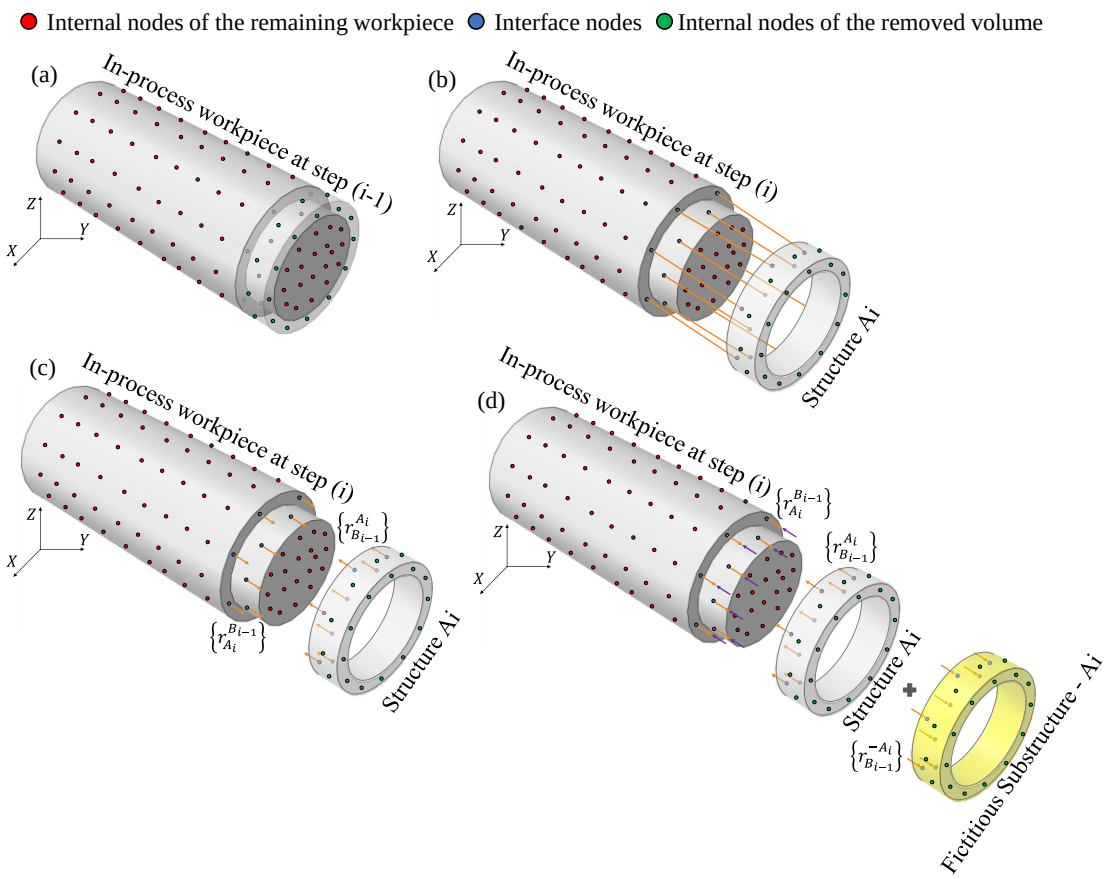


Figure 3-9 (a) In-process workpiece at step (i-1), (b) interaction between the remaining workpiece and substructures considered as rigid links on $cDOF$, (c) interaction between the remaining workpiece and substructures shown by reaction forces, (d) opposite interaction forces at $cDOF$ generated by the imaginary substructure

The equation of motion in the time domain for the workpiece is [71]

$$\boldsymbol{M}^{B_{i-1}}(t)\ddot{\boldsymbol{x}}^{B_{i-1}}(t) + \boldsymbol{K}^{B_{i-1}}(t)\boldsymbol{x}^{B_{i-1}}(t) = \boldsymbol{f}^{B_{i-1}}(t) + \boldsymbol{r}^{B_{i-1}}(t), \tag{3.8}$$

where $\boldsymbol{x}^{B_{i-1}}$ is the displacement vector for all the DOF, $\boldsymbol{f}^{B_{i-1}}$ represents the applied force, and $\boldsymbol{r}^{B_{i-1}}$ is the coupling or reaction force. The reaction force only affects the interface nodes and their corresponding DOF. The expression of this equation of motion in the frequency domain [46] is

$$\boldsymbol{x}^{B_{i-1}}(\omega) = \boldsymbol{H}^{B_{i-1}}(\omega)\{\boldsymbol{f}^{B_{i-1}}(\omega) + \boldsymbol{r}^{B_{i-1}}(\omega)\}, \tag{3.9}$$

where ω is the frequency and $\boldsymbol{H}^{B_{i-1}}(\omega)$ represents the FRF matrix. Also, the vector of displacements caused by the externally applied force [46] is

$$\boldsymbol{u}^{B_{i-1}}(\omega) = \boldsymbol{H}^{B_{i-1}}(\omega)\boldsymbol{f}^{B_{i-1}}(\omega). \tag{3.10}$$

Substituting Eq. (3.10) into Eq. (3.9) gives

$$\boldsymbol{x}^{B_{i-1}}(\omega) - \boldsymbol{u}^{B_{i-1}}(\omega) = \boldsymbol{H}^{B_{i-1}}(\omega)\boldsymbol{r}^{B_{i-1}}(\omega). \tag{3.11}$$

Similarly, the equation of motion for the added fictitious substructure in the time domain [46] can be written as

$$-\boldsymbol{M}^{A_i}(t)\ddot{\boldsymbol{x}}^{-A_i}(t) + [-\boldsymbol{K}^{A_i}(t)]\boldsymbol{x}^{-A_i}(t) = \boldsymbol{f}^{-A_i}(t) + \boldsymbol{r}^{-A_i}(t). \tag{3.12}$$

Since the material removal at different machining stages can be non-uniform, the mass and stiffness matrices are written as a function of time. Similar to the equation of motion for the in-process workpiece, $\boldsymbol{f}^{-A_i}$ and $\boldsymbol{r}^{-A_i}$ are the external force and the coupling force acting on the fictitious substructure. This equation in the frequency domain is

$$-\boldsymbol{D}^{A_i}(\omega)\boldsymbol{x}^{-A_i}(\omega) = \boldsymbol{f}^{-A_i}(\omega) + \boldsymbol{r}^{-A_i}(\omega). \tag{3.13}$$

In this equation, $\boldsymbol{D}^{A_i}(\omega)$ is the dynamic stiffness matrix which is defined as

$$\boldsymbol{D}^{A_i}(\omega) = -\omega^2\boldsymbol{M}^{A_i} + \boldsymbol{K}^{A_i}. \tag{3.14}$$

Combining Eq. (3.11) and Eq. (3.13), we have [46]

$$\begin{bmatrix} \boldsymbol{H}^{B_{i-1}}(\omega) & 0 \\ 0 & -\boldsymbol{D}^{A_i}(\omega) \end{bmatrix} \begin{Bmatrix} \boldsymbol{r}^{B_{i-1}}(\omega) \\ \boldsymbol{x}^{-A_i}(t) \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{u}^{B_{i-1}}(\omega) \\ \boldsymbol{f}^{-A_i}(t) \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{x}^{B_{i-1}}(\omega) \\ \boldsymbol{r}^{-A_i}(t) \end{Bmatrix}. \tag{3.15}$$

In this set of equations, the interaction forces and the displacements of both structures are used on both sides. In this situation, the compatibility and equilibrium conditions cannot be satisfied directly [72]. As a result, Eq. (3.42) must be expanded to enable assembly, and both substructures' forces and interface displacements must be in the global DOF set with explicit partitioning. It means all the components involved in this equation set must be divided into subsets respective to each structure's DOF sets. Once we have the re-arranged DOF sets, Eq. (3.15) can be rewritten as [72]

$$\begin{bmatrix} \boldsymbol{H}_{aa}^{B_{i-1}} & \boldsymbol{H}_{ac}^{B_{i-1}} & \boldsymbol{H}_{ab}^{B_{i-1}} & -\boldsymbol{I}_{aa} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_{ca}^{B_{i-1}} & \boldsymbol{H}_{cc}^{B_{i-1}} & \boldsymbol{H}_{cb}^{B_{i-1}} & \boldsymbol{0} & -\boldsymbol{I}_{cc} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_{ba}^{B_{i-1}} & \boldsymbol{H}_{bc}^{B_{i-1}} & \boldsymbol{H}_{bb}^{B_{i-1}} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}_{bb} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}_{aa} & \boldsymbol{0} & -\boldsymbol{D}_{aa}^{A_i} & -\boldsymbol{D}_{ac}^{A_i} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}_{cc} & -\boldsymbol{D}_{ca}^{A_i} & -\boldsymbol{D}_{cc}^{A_i} \end{bmatrix} \begin{Bmatrix} \boldsymbol{r}_a^{B_{i-1}} \\ \boldsymbol{r}_c^{B_{i-1}} \\ \boldsymbol{r}_b^{B_{i-1}} \\ \boldsymbol{x}_a^{B_{i-1}} \\ \boldsymbol{x}_b^{B_{i-1}} \\ \boldsymbol{x}_c^{B_{i-1}} \\ \boldsymbol{r}_a^{-A_i} \\ \boldsymbol{r}_c^{-A_i} \\ \boldsymbol{x}_a^{-A_i} \\ \boldsymbol{x}_c^{-A_i} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{u}_a^{B_{i-1}} \\ -\boldsymbol{u}_c^{B_{i-1}} \\ -\boldsymbol{u}_b^{B_{i-1}} \\ \boldsymbol{0}_a^{B_{i-1}} \\ \boldsymbol{0}_c^{B_{i-1}} \\ \boldsymbol{0}_b^{B_{i-1}} \\ \boldsymbol{0}_a^{-A_i} \\ \boldsymbol{0}_c^{-A_i} \\ \boldsymbol{f}_a^{-A_i} \\ \boldsymbol{f}_c^{-A_i} \end{Bmatrix}, \tag{3.16}$$

where $\boldsymbol{I}$ is the identity matrix. The subindices for each vector or matrix identify the corresponding DOF set. Since the original equation set is augmented into this expanded form, the interface coupling forces and displacements ($\boldsymbol{r}_c$, $\boldsymbol{x}_c$) are mentioned in the formulation twice, once as part of the workpiece and the other time as part of the removed volume. This duplication can be eliminated by using a Boolean matrix as [72], [73]

$$\begin{Bmatrix} \boldsymbol{r}_a^{B_{i-1}} \\ \boldsymbol{r}_c^{B_{i-1}} \\ \boldsymbol{r}_b^{B_{i-1}} \\ \boldsymbol{x}_a^{B_{i-1}} \\ \boldsymbol{x}_b^{B_{i-1}} \\ \boldsymbol{x}_c^{B_{i-1}} \\ \boldsymbol{r}_a^{-A_i} \\ \boldsymbol{r}_c^{-A_i} \\ \boldsymbol{x}_a^{-A_i} \\ \boldsymbol{x}_c^{-A_i} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{I}_{aa} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{cc} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{bb} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{aa} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{cc} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{bb} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{aa} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I}_{cc} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{aa} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{cc} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{r}_a^{B_{i-1}} \\ \boldsymbol{r}_c \\ \boldsymbol{r}_b^{B_{i-1}} \\ \boldsymbol{x}_a^{B_{i-1}} \\ \boldsymbol{x}_c \\ \boldsymbol{x}_b^{B_{i-1}} \\ \boldsymbol{r}_a^{-A_i} \\ \boldsymbol{x}_a^{-A_i} \end{Bmatrix} = \boldsymbol{T}_b \begin{Bmatrix} \boldsymbol{r}_a^{B_{i-1}} \\ \boldsymbol{r}_c \\ \boldsymbol{r}_b^{B_{i-1}} \\ \boldsymbol{x}_a^{B_{i-1}} \\ \boldsymbol{x}_c \\ \boldsymbol{x}_b^{B_{i-1}} \\ \boldsymbol{r}_a^{-A_i} \\ \boldsymbol{x}_a^{-A_i} \end{Bmatrix}. \tag{3.17}$$

In this equation, $\boldsymbol{T}_b$ is a Boolean or logical matrix consisting of identity and zero matrices. This transformation matrix also satisfies the required equilibrium and compatibility of interface DOFs. The distribution of coupling forces in interface DOFs, shown by Lagrange multipliers which are $\boldsymbol{r}_c = -\boldsymbol{r}_c^{-A_i} = \boldsymbol{r}_c^{B_{i-1}}$, fulfills equilibrium on interface DOF [70], [74], and the displacement vector for interface DOF, which is represented by $\boldsymbol{x}_c = \boldsymbol{x}_c^{-A_i} = \boldsymbol{x}_c^{B_{i-1}}$, satisfies the interface compatibility.

This equation results in a unique set of DOF. By inserting this DOF set into Eq. (3.16) and pre-multiplying it by $\boldsymbol{T}_b^T$, we have the coupled set of equations as [46]

$$\begin{bmatrix} \boldsymbol{H}_{aa}^{B_{i-1}} & \boldsymbol{H}_{ac}^{B_{i-1}} & \boldsymbol{H}_{ab}^{B_{i-1}} & -\boldsymbol{I}_{aa} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_{ca}^{B_{i-1}} & \boldsymbol{H}_{cc}^{B_{i-1}} & \boldsymbol{H}_{cb}^{B_{i-1}} & \boldsymbol{0} & -\boldsymbol{I}_{cc} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_{ba}^{B_{i-1}} & \boldsymbol{H}_{bc}^{B_{i-1}} & \boldsymbol{H}_{bb}^{B_{i-1}} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}_{bb} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{cc} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{D}_{cc}^{A_i} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{D}_{ca}^{A_i} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{D}_{ac}^{A_i} & \boldsymbol{0} & -\boldsymbol{I}_{aa} & -\boldsymbol{D}_{aa}^{A_i} \end{bmatrix} \begin{Bmatrix} \boldsymbol{r}_a^{B_{i-1}} \\ \boldsymbol{r}_c \\ \boldsymbol{r}_b^{B_{i-1}} \\ \boldsymbol{x}_a^{B_{i-1}} \\ \boldsymbol{x}_c \\ \boldsymbol{x}_b^{B_{i-1}} \\ \boldsymbol{r}_a^{-A_i} \\ \boldsymbol{x}_a^{-A_i} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{u}_a^{B_{i-1}} \\ -\boldsymbol{u}_c^{B_{i-1}} \\ -\boldsymbol{u}_b^{B_{i-1}} \\ \boldsymbol{0}_a^{B_{i-1}} \\ \boldsymbol{f}_c^{-A_i} \\ \boldsymbol{0}_b^{B_{i-1}} \\ \boldsymbol{0}_a^{-A_i} \\ \boldsymbol{f}_a^{-A_i} \end{Bmatrix}. \quad (3.18)$$

The interface DOFs fulfill the coupling in the equation, so the coupling forces only operate on $cDOF$. Considering this, the reaction force vectors of $aDOF$ and $bDOF$ are zero, and we can replace them with zero vectors in Eq. (3.18) as [46]

$$\begin{bmatrix} \boldsymbol{H}_{aa}^{B_{i-1}} & \boldsymbol{H}_{ac}^{B_{i-1}} & \boldsymbol{H}_{ab}^{B_{i-1}} & -\boldsymbol{I}_{aa} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_{ca}^{B_{i-1}} & \boldsymbol{H}_{cc}^{B_{i-1}} & \boldsymbol{H}_{cb}^{B_{i-1}} & \boldsymbol{0} & -\boldsymbol{I}_{cc} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_{ba}^{B_{i-1}} & \boldsymbol{H}_{bc}^{B_{i-1}} & \boldsymbol{H}_{bb}^{B_{i-1}} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}_{bb} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{cc} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{D}_{cc}^{A_i} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{D}_{ca}^{A_i} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{D}_{ac}^{A_i} & \boldsymbol{0} & -\boldsymbol{I}_{aa} & -\boldsymbol{D}_{aa}^{A_i} \end{bmatrix} \begin{Bmatrix} \boldsymbol{0}_a^{B_{i-1}} \\ \boldsymbol{r}_c \\ \boldsymbol{0}_b^{B_{i-1}} \\ \boldsymbol{x}_a^{B_{i-1}} \\ \boldsymbol{x}_c \\ \boldsymbol{x}_b^{B_{i-1}} \\ \boldsymbol{0}_a^{-A_i} \\ \boldsymbol{x}_a^{-A_i} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{u}_a^{B_{i-1}} \\ -\boldsymbol{u}_c^{B_{i-1}} \\ -\boldsymbol{u}_b^{B_{i-1}} \\ \boldsymbol{0}_a^{B_{i-1}} \\ \boldsymbol{f}_c^{-A_i} \\ \boldsymbol{0}_b^{B_{i-1}} \\ \boldsymbol{0}_a^{-A_i} \\ \boldsymbol{f}_a^{-A_i} \end{Bmatrix}. \tag{3.19}$$

This way, the effects of the right-hand side matrix's first, third, and seventh columns are eliminated. As a result, this equation can be rewritten as [46]

$$\begin{bmatrix} \boldsymbol{H}_{ac}^{B_{i-1}} & -\boldsymbol{I}_{aa} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_{cc}^{B_{i-1}} & \boldsymbol{0} & -\boldsymbol{I}_{cc} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_{bc}^{B_{i-1}} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}_{bb} & \boldsymbol{0} \\ \boldsymbol{I}_{cc} & \boldsymbol{0} & -\boldsymbol{D}_{cc}^{A_i} & \boldsymbol{0} & -\boldsymbol{D}_{ca}^{A_i} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{D}_{ac}^{A_i} & \boldsymbol{0} & -\boldsymbol{D}_{aa}^{A_i} \end{bmatrix} \begin{Bmatrix} \boldsymbol{r}_c \\ \boldsymbol{x}_a^{B_{i-1}} \\ \boldsymbol{x}_c \\ \boldsymbol{x}_b^{B_{i-1}} \\ \boldsymbol{x}_a^{-A_i} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{u}_a^{B_{i-1}} \\ -\boldsymbol{u}_c^{B_{i-1}} \\ -\boldsymbol{u}_b^{B_{i-1}} \\ \boldsymbol{f}_c^{-A_i} \\ \boldsymbol{f}_a^{-A_i} \end{Bmatrix}. \tag{3.20}$$

In order to find the new FRF for the remaining workpiece after updating the workpiece, Eq. (3.20) must be solved for interface DOFs ($cDOF$), and internal DOFs ($bDOF$) of the workpiece. To improve the numerical stability of the computations, coupling forces are omitted. Gauss–Jordan elimination method, in combination with Singular Value Decomposition, is used to perform this process [75], [76]. This stability and improvement are significant for the removed volume at the anti-resonance frequencies since decoupling shows more sensitivity at these frequencies [69], [70].

Since the internal DOF of the removed material, as well as its dynamics, are removed from the internal DOFs of B_{i-1}, all the forces acting on these DOFs are equal to zero in Eq.

(3.20). Therefore, the direct FRF and cross FRF of the workpiece (B_i) are calculated and presented in the following sections.

3.3.1 Direct FRF at Interface DOFs of the Workpiece

As we know, the cutting forces of machining only act on the interface between the removed material and the workpiece. As a result, to predict the vibration and chatter stability, the updated FRF at $cDOF$ is needed.

Using the fifth equation from Eq. (3.20), the displacement vector for the internal DOFs of removed material can be evaluated as [46]

$$\begin{gathered} -\boldsymbol{x}_c \boldsymbol{D}_{ac}^{A_i}(\omega) - \boldsymbol{x}_a^{-A_i} \boldsymbol{D}_{aa}^{A_i}(\omega) = \boldsymbol{0}_a^{-A_i}, \\ \boldsymbol{x}_a^{-A_i} = -\boldsymbol{x}_c \boldsymbol{D}_{ac}^{A_i}(\omega) \left[\boldsymbol{D}_{aa}^{A_i}(\omega)\right]^{-1}. \end{gathered} \tag{3.21}$$

From the fourth equation of Eq. (3.20), we have

$$\begin{gathered} \boldsymbol{r}_c - \boldsymbol{x}_c \boldsymbol{D}_{cc}^{A_i}(\omega) - \boldsymbol{x}_a^{-A_i} \boldsymbol{D}_{ca}^{A_i}(\omega) = \boldsymbol{0}_c^{-A_i}, \\ \boldsymbol{x}_c \boldsymbol{D}_{cc}^{A_i}(\omega) + \boldsymbol{x}_a^{-A_i} \boldsymbol{D}_{ca}^{A_i}(\omega) = \boldsymbol{r}_c. \end{gathered} \tag{3.22}$$

By inserting Eq. (3.21) into Eq. (3.22), the force vector for the interface DOFs can be obtained as

$$\left[\boldsymbol{D}_{cc}^{A_i}(\omega) - \boldsymbol{D}_{ac}^{A_i}(\omega)\left[\boldsymbol{D}_{aa}^{A_i}(\omega)\right]^{-1} \boldsymbol{D}_{ca}^{A_i}(\omega)\right] \boldsymbol{x}_c = \widetilde{\boldsymbol{D}}^{A_i} \boldsymbol{x}_c = \boldsymbol{r}_c. \tag{3.23}$$

In this formulation, $\widetilde{\boldsymbol{D}}^{A_i}$ is a matrix called condensed dynamic stiffness defined for interface DOFs. From the second equation in Eq. (3.20) and Eq. (3.10), we have

$$\boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\boldsymbol{r}_c - \boldsymbol{I}_{cc}x_c = -\boldsymbol{u}_c^{B_{i-1}} = -\boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\boldsymbol{f}_c^{B_{i-1}}. \tag{3.24}$$

Inserting Eq. (3.23) into Eq. (3.24) results in

$$\begin{aligned} \boldsymbol{I}_{cc}x_c - \boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\widetilde{\boldsymbol{D}}^{A_i}\boldsymbol{x}_c &= \boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\boldsymbol{f}_c^{B_{i-1}}, \\ \left[\boldsymbol{I}_{cc} - \boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\widetilde{\boldsymbol{D}}^{A_i}\right]\boldsymbol{x}_c &= \boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\boldsymbol{f}_c^{B_{i-1}}, \\ \boldsymbol{x}_c &= \left[\boldsymbol{I}_{cc} - \boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1}\boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\boldsymbol{f}_c^{B_{i-1}}. \end{aligned} \tag{3.25}$$

From this equation, the direct FRF of the interface DOF can be evaluated as [46]

$$\boldsymbol{H}_{cc}^{B_i} = \left[\boldsymbol{I}_{cc} - \boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1}\boldsymbol{H}_{cc}^{B_{i-1}}(\omega). \tag{3.26}$$

3.3.2 Cross FRF Between Internal DOF and Interface DOF of Workpiece

When the FRF is obtained in a DOF other than the DOF to which the force is applied, it is called cross FRF. Since the interface DOFs and the internal DOFs of the workpiece change along the toolpath, the interface DOFs at a given machining step could have mutual DOFs with the further steps due to the constant machining process.

In order to obtain cross FRF, we assume $\boldsymbol{f}_c^{B_{i-1}} = \boldsymbol{0}_c$. Substituting Eq. (3.23) into the second equation of Eq. (3.20) gives [46]

$$\begin{aligned} \left[\boldsymbol{I}_{cc} - \boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1}\boldsymbol{H}_{cb}^{B_{i-1}}(\omega)\boldsymbol{f}_b^{B_{i-1}} &= \boldsymbol{x}_c, \\ \boldsymbol{H}_{cb}^{B_i} &= \left[\boldsymbol{I}_{cc} - \boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1}\boldsymbol{H}_{cb}^{B_{i-1}}(\omega). \end{aligned} \tag{3.27}$$

3.3.3 Direct FRF at Internal DOF of the Workpiece

During machining, some of the internal DOFs of the workpiece at a given step become interface DOFs in the further steps. As a result, we also need the updated FRF for these DOFs.

When calculating the FRF for the internal DOFs of workpiece, the interface forces are $\boldsymbol{f}_c^{B_{i-1}} = \boldsymbol{0}_c$. So, Eq. (3.23) can be rewritten as [46]

$$\boldsymbol{x}_c = \left[\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1} \boldsymbol{r}_c. \tag{3.28}$$

By inserting this equation into the second equation of Eq. (3.20), we have

$$\boldsymbol{r}_c = -\left[\boldsymbol{H}_{cc}^{B_{i-1}}(\omega) - \left[\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1}\right]^{-1} \boldsymbol{H}_{cb}^{B_{i-1}}(\omega)\boldsymbol{f}_b^{B_{i-1}}. \tag{3.29}$$

Also, from Eq. (3.10), we have

$$\boldsymbol{u}_b^{B_{i-1}}(\omega) = \boldsymbol{H}_{bb}^{B_{i-1}}(\omega)\boldsymbol{f}_b^{B_{i-1}}(\omega). \tag{3.30}$$

Substituting Eq. (3.29) and (3.30) into the third equation of Eq. (3.20) results in

$$\begin{gathered}
\boldsymbol{H}_{bc}^{B_{i-1}}(\omega)\left[\boldsymbol{H}_{cc}^{B_{i-1}}(\omega) - \left[\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1}\right]^{-1} \boldsymbol{H}_{cb}^{B_{i-1}}(\omega)\boldsymbol{f}_b^{B_{i-1}} + \boldsymbol{x}_b^{B_{i-1}} = \\
\boldsymbol{H}_{bb}^{B_{i-1}}(\omega)\boldsymbol{f}_b^{B_{i-1}}(\omega), \\
\boldsymbol{x}_b^{B_{i-1}} = \boldsymbol{H}_{bb}^{B_{i-1}}(\omega)\boldsymbol{f}_b^{B_{i-1}}(\omega) - \boldsymbol{H}_{bc}^{B_{i-1}}(\omega)\left[\boldsymbol{H}_{cc}^{B_{i-1}}(\omega) - \right. \\
\left.\left[\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1}\right]^{-1} \boldsymbol{H}_{cb}^{B_{i-1}}(\omega)\boldsymbol{f}_b^{B_{i-1}}(\omega),
\end{gathered} \tag{3.31}$$

$$\boldsymbol{x}_b^{B_{i-1}} = \left[\boldsymbol{H}_{bb}^{B_{i-1}}(\omega) - \boldsymbol{H}_{bc}^{B_{i-1}}(\omega)\left[\boldsymbol{H}_{cc}^{B_{i-1}}(\omega) - \left[\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1}\right]^{-1}\boldsymbol{H}_{cb}^{B_{i-1}}(\omega)\right]\boldsymbol{f}_b^{B_{i-1}}(\omega).$$

From this equation, the updated direct FRF at internal DOFs is calculated as

$$\boldsymbol{H}_{bb}^{B_i}(\omega) = \boldsymbol{H}_{bb}^{B_{i-1}}(\omega) - \boldsymbol{H}_{bc}^{B_{i-1}}(\omega)\left[\boldsymbol{H}_{cc}^{B_{i-1}}(\omega) - \left[\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1}\right]^{-1}\boldsymbol{H}_{\boldsymbol{cb}}^{\boldsymbol{B_{i-1}}}(\omega). \tag{3.32}$$

In order to write this equation in a similar format to the previous ones, with some matrix manipulations, we have [46]

$$\boldsymbol{H}_{bb}^{B_i}(\omega) = \boldsymbol{H}_{bb}^{B_{i-1}}(\omega) + \boldsymbol{H}_{bc}^{B_{i-1}}(\omega)\widetilde{\boldsymbol{D}}^{A_i}\left[\boldsymbol{I}_{cc} - \boldsymbol{H}_{cc}^{B_{i-1}}(\omega)\widetilde{\boldsymbol{D}}^{A_i}\right]^{-1}\boldsymbol{H}_{cb}^{B_{i-1}}(\omega). \tag{3.33}$$

As the machining process moves forward by performing several cutting steps, the updated FRF of the in-process workpiece at different steps of machining can be obtained by using Eq. (3.26), (3.27), and (3.33) in a recursive manner. This method has some advantages over traditional methods. The first advantage of this method is that, because of partitioning the workpiece from the beginning and removing its substructures by adding fictitious substructures for simulating the machining process, there is no need to create several FE models for different machining steps. This leads to nodal compatibility at interface nodes which enables rigid coupling. Also, the updated FRF from this method shows the dominant modes at cutting regions which can be used for vibration stability and chatter prediction.

3.4 Validation Results

A simulation is carried out to validate the proposed FRF updating model. To do so, cylindrical workpieces at different machining steps are modeled by the ABAQUS Learning

Edition 2022 [77] and the developed FE algorithm. The cylinder length is 240 mm and has a diameter of 100 mm. The selected material is AISI 1080 Steel with a density of 7800 $\mathrm{kg/m^3}$, an elastic modulus of 200 GPa, and a Poisson's ratio of 0.3 [78], [79]. The initial part's FE model has 2,835 DOF, and the machining steps are considered in modeling this geometry by proper node and element partitioning. Here, three machining steps are simulated. In the first step, a ring with a thickness of 10 mm and a length of 25 mm is removed from the tip of the cylinder. In the second step, a layer with a thickness of 2 mm and length of 125 mm is removed from the outer side of the body of the cylinder. Finally, the third step is removing a ring with a thickness of 3 mm and length of 10 mm near the cylinder's end. These machining scenarios are shown in Figure 3-10, along with the dimension of the initial part and in-process workpieces. Three models are also generated for in-process workpieces to compare the updating algorithm with the regular FE results. These models are shown in Figure 3-10.

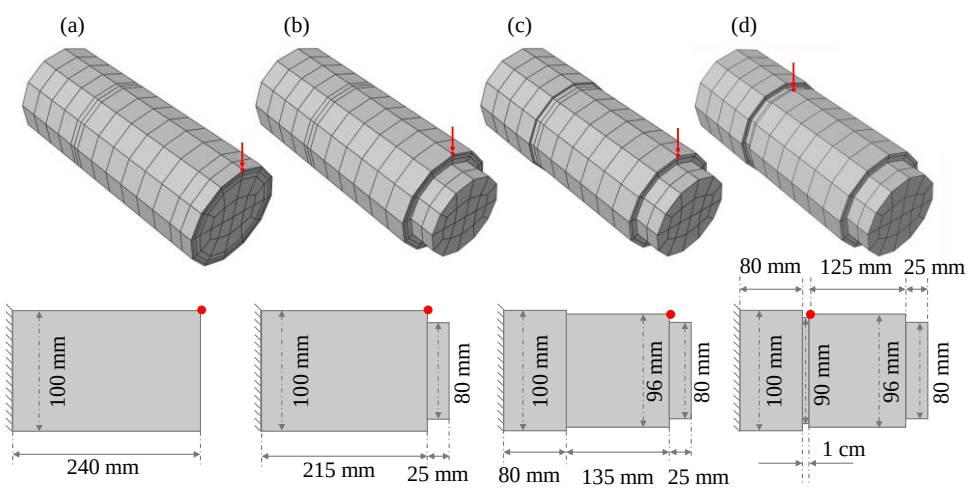


Figure 3-10 (a) Initial part geometry and FE model, (b) In-process workpiece geometry and FE model at the first step of machining, (c) In-process workpiece geometry and FE model at the second step of machining, (d) In-process workpiece geometry and FE model at the third step of machining

To validate the results, first, the FRF of the initial workpiece is calculated by the developed FE algorithm and ABAQUS, which are shown in Figure 3-11. As seen in the plot, the developed FE code can calculate the FRF of the geometry with acceptable accuracy.

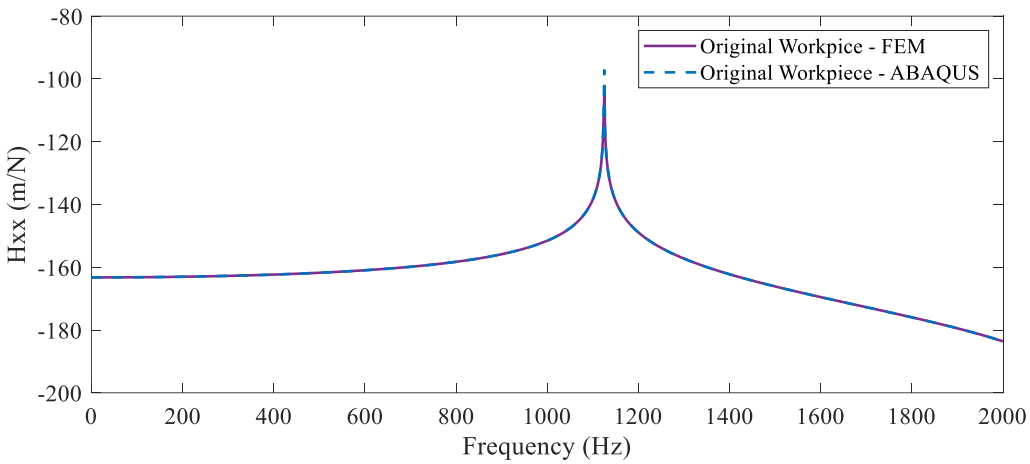


Figure 3-11 Initial part's FRF calculated by the developed FE algorithm and ABAQUS

To validate the FRF results obtained from the proposed solution, the FRF of the in-process workpiece after each machining step is calculated from an FE model based on the machined part and from the initial workpiece using the developed FRF updating method. These FRFs, along with the ABAQUS results, are shown in Figure 3-12 to Figure 3-14

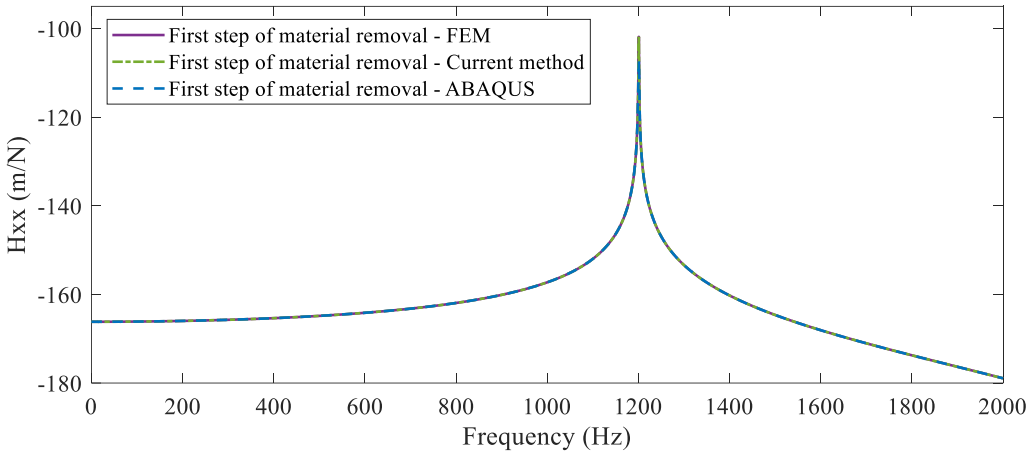


Figure 3-12 FRF of the in-process workpiece after the first machining step, calculated by the developed FE algorithm and ABAQUS

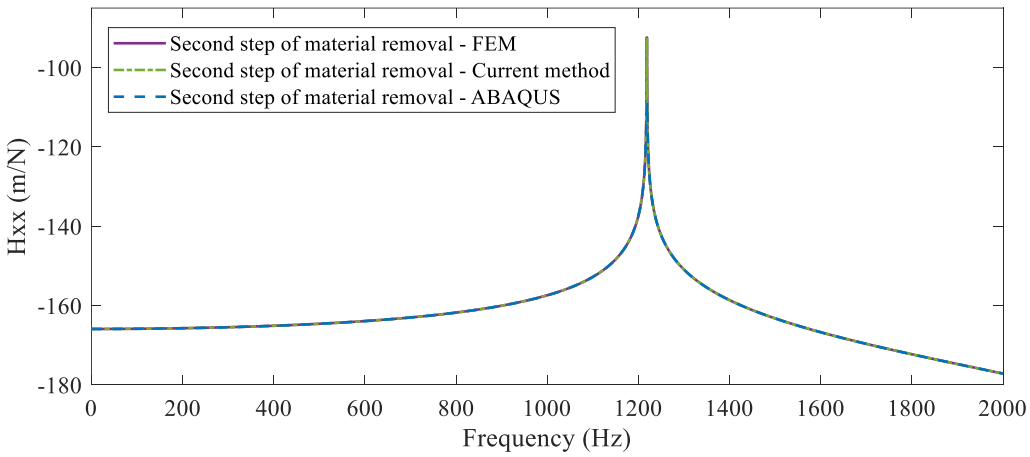


Figure 3-13 FRF of the in-process workpiece after the second machining step, calculated by the developed FE algorithm and ABAQUS

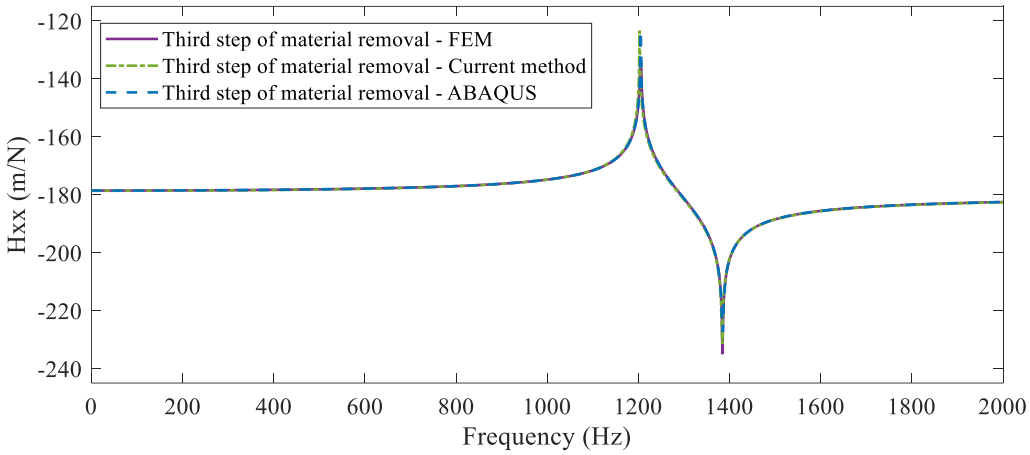


Figure 3-14 FRF of the in-process workpiece after the third machining step, calculated by the developed FE algorithm and ABAQUS

It is observed in these plots that the results of the proposed FRF updating method are in good agreement with ABAQUS results. However, there are some differences at the extreme points in some of the plots. These mismatches are the results of numerical errors.

Although we can calculate the FRF of the workpiece during different machining steps ahead of time without creating multiple FE models, this method is still computationally heavy and time-consuming. In the next chapter, different methods are put forward to further increase the computational speed.

3.5 Conclusion

In this chapter, a finite element model is developed using eight-node elements, and the boundary conditions are applied. Using this model and negative structural modification, an FRF updating algorithm is developed. This algorithm can predict the FRF of the in-process workpiece at different machining stages. A simulation with a cylindrical part as the

initial workpiece is carried out. Three material removal steps were considered, and the part was partitioned accordingly. The simulation results are compared with ABAQUS for validation. It is shown that the developed model is able to predict the FRF of the workpiece after each step of machining with reasonable accuracy.

4 Towards Faster Computation: Model Order Reduction

4.1 Overview

In the previous chapter, a method for calculating the FRF of the in-process workpiece without creating new FE models for each machining step was proposed. In this chapter, two different methods are used to increase the simulation speed. First is model order reduction, which limits the computations to the active DOFs. This method improves the efficiency of the developed algorithm by multiplying the speed of calculations, and it is explained extensively in Section 4.2. Another effective method for reducing computational load and time is using mode superposition for calculating the FRF. This method, which lifts heavy computational load by reducing the number of active modes in FRF calculation, is explained in detail in Section 4.3. The simulation results of the improved model are provided in Section 4.4, and the chapter is concluded in Section 4.5.

4.2 Model Order Reduction

Order reduction is an effective method to reduce the computational load in simulations. This method uses different approaches to increase the speed of calculations. This is most important when we have a model of considerable size, resulting in high computational time. For the FE model used for solving dynamic problems, we can effectively

reduce the model size without modifying the initial FE mesh grid using model order reduction. In this method, a selected group of DOFs are used in the solution process, and the effect of the rest of the model is reflected in these DOFs. This group of DOFs is called master DOFs, and we start this process by selecting them.

4.2.1 Selecting the Master DOF

In classical dynamics, when there are different parts, the DOFs with the most contribution to each part's dynamics are determined independently [80]. However, in this study, the removed materials, which are the fictitious substructures in the developed model, are generated by dividing the initial workpiece into proper partitions; this means the initial part and the removed volumes must have nodal compatibility, even after order reduction. To preserve nodal conformity, first, the master DOF of each removed volume (A_i) must be found. These master DOFs are retained in the master DOF set of the initial workpiece (B_0) even if these DOFs do not significantly contribute to the initial part's dynamics.

In order to find the master DOF of each structure, the iterative effective independence technique is used.

4.2.2 Iterative Effective Independence

This section presents a method for finding a limited number of DOF within a more extensive set. This method ranks the DOF based on their contribution to the spatial independence of the modes and modal kinetic energy [81]. The DOFs with the most contribution to the model's dynamics are opted as the master DOFs and kept in the calculations. On the other hand, the DOFs that have the most negligible contribution to the

model's dynamics are selected as the slave DOFs and are eliminated in an iterative manner. A schematic of the master and slave DOFs of a cylinder under torsion and bending modes is shown in Figure 4-1.

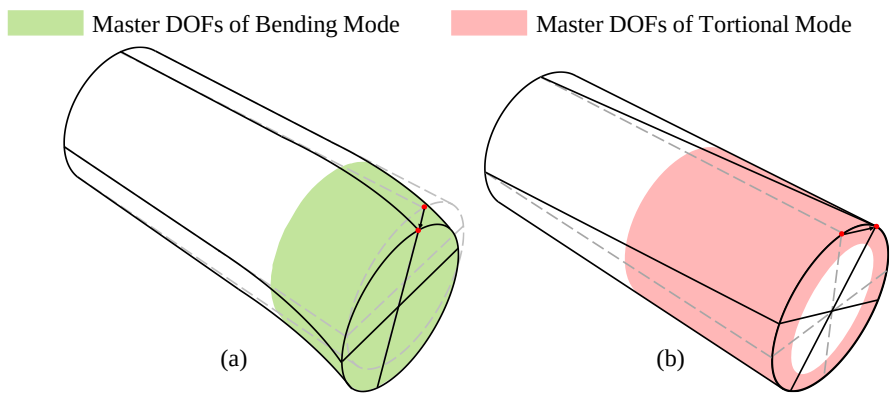


Figure 4-1 Schematics of the master DOFs of a cylinder in (a) bending mode, (b) torsional mode

This method begins with all the DOF as the master DOF. The next set of potential master DOFs with fewer members is selected based on the distribution of modal kinetic energy that shows how each DOF contributes to each mode shape. The distribution of modal kinetic energy is obtained by [81]

$$KE_{in} = \phi_{in} \sum_j M_{ij} \phi_{jn}, \tag{4.1}$$

where KE_{in} is the kinetic energy of ith DOF in the nth mode shape, ϕ_{in} is the ith element in that mode shape, and M_{ij} is the mass matrix element.

In addition to contributing to the modal kinetic energy, we are interested in the DOFs that keep the significant modes spatially independent. If the displacement of a given set of potential master DOFs can be calculated by [82]

$$\boldsymbol{u}_m = \boldsymbol{\phi}_p \boldsymbol{q}, \tag{4.2}$$

those DOFs are spatially independent. In this equation $\boldsymbol{u}_m$ is the displacement of the given DOFs, $\boldsymbol{\phi}_p$ is the matrix of significant targeted modes, and $\boldsymbol{q}$ is a vector containing modal coordinates. Then we can estimate the significant modal states ($\hat{q}$) by solving Eq. (4.2), which gives

$$\hat{\boldsymbol{q}} = \left[\boldsymbol{\phi}_p^T \boldsymbol{\phi}_p\right]^{-1} \boldsymbol{\phi}_p^T \boldsymbol{u}_m. \tag{4.3}$$

We assume that all DOFs keep the significant modes linearly independent. We need to select a limited number of DOFs so that the most possible independent information is maintained and we have the best estimation for modal states. It means the estimate errors' covariance matrix must be minimum. As a result, the displacement equation must be modified, and a term for error is added as

$$\boldsymbol{u}_m = \boldsymbol{\phi}_p \boldsymbol{q} + \boldsymbol{Gn}, \tag{4.4}$$

where $\boldsymbol{Gn}$ is a Gaussian white noise, and its variance is $\boldsymbol{\psi}_o^2$. If the estimation is unbiased, the covariance of the error is [82]

$$\boldsymbol{P} = \boldsymbol{EV}[(\boldsymbol{q} - \hat{\boldsymbol{q}})(\boldsymbol{q} - \hat{\boldsymbol{q}})^T] = \left[\frac{\partial(\boldsymbol{\phi}_p \boldsymbol{q})}{\partial \boldsymbol{q}}\right]^T [\boldsymbol{\psi}_o^2]^{-1} \left[\frac{\partial(\boldsymbol{\phi}_p \boldsymbol{q})}{\partial \boldsymbol{q}}\right]^{-1}, \tag{4.5}$$

where $\boldsymbol{EV}$ is the expected value. Therefore, the covariance matrix is

$$\boldsymbol{P} = \left[\boldsymbol{\phi}_p^T [\boldsymbol{\psi}_o^2]^{-1} \boldsymbol{\phi}_p\right] = \boldsymbol{S}^{-1}. \tag{4.6}$$

In this equation, $\boldsymbol{S}$ is called the Fisher Information Matrix (FIM). By maximizing $\boldsymbol{S}$, the covariance matrix is minimized, which means we have the best estimation for $\hat{\boldsymbol{q}}$. For simplification, assume that we have an uncorrelated noise with identical statistical properties for each DOF. In this way, the FIM can be simplified as

$$\boldsymbol{S} = \frac{1}{\boldsymbol{\psi}_o^2} \boldsymbol{\phi}_p^T \boldsymbol{\phi}_p = \frac{1}{\boldsymbol{\psi}_o^2} \boldsymbol{O}_o. \tag{4.7}$$

By maximizing a proper norm of $\boldsymbol{O}_o$, $\boldsymbol{P}$ is minimized. We know the trace norm is a very useful and physically meaningful matrix norm [83], and for the contribution of each DOF, we can represent it as

$$\boldsymbol{O}_o = \sum_{i=1}^{s} \boldsymbol{\phi}_p^{iT} \boldsymbol{\phi}_p^i = \sum_{i=1}^{s} \boldsymbol{O}^i. \tag{4.8}$$

In this equation, $\boldsymbol{\phi}_p^i$ is the i^{th}row of the significant mode shapes corresponding to the ith DOF.

Based on Eq. (4.8), adding a DOF to or subtracting it from the potential set of master DOFs has the same effect on the information content of the FIM. The primary purpose of this section is to reduce the order of the problem by eliminating the DOFs with a negligible contribution to the independent information of the significant modes. To reduce the number of DOF, we start with solving the eigenvalue problem. We assume that for the initial potential master DOF, the columns of $\boldsymbol{\phi}_p$ correspond to the significant modes and are linearly

independent. Assume we selected p first mode shapes as the significant modes. Therefore, the matrix $\boldsymbol{O}_o$ with size $p \times p$ is symmetric and positive definite [82]. As a result, the eigenvectors of $\boldsymbol{O}_o$ are orthonormal, and its eigenvalues are real and positive, which leads to

$$\boldsymbol{\psi}^T \boldsymbol{O}_o \boldsymbol{\psi} = \lambda \text{ and } \boldsymbol{\psi}^T \boldsymbol{\psi} = \boldsymbol{I}. \tag{4.9}$$

Here, $\boldsymbol{\psi}$ is the eigenvector, and λ is the eigenvalue of $\boldsymbol{O}_o$. The orthogonal eigenvectors represent p orthogonal directions in a space with p dimensions called Absolute Identification Space (AIS). If we form matrix $\boldsymbol{G}$ as

$$\boldsymbol{G} = [\boldsymbol{\phi}_p \boldsymbol{\psi}] \circ [\boldsymbol{\phi}_p \boldsymbol{\psi}], \tag{4.10}$$

where $\circ$ shows an element-wise matrix multiplication, we have a matrix in which rows consist of squares of the $\boldsymbol{\phi}_p$ rows' elements with respect to the columns of $\boldsymbol{\psi}$, which covers the AIS. The summation of the elements of each column of matrix $\boldsymbol{G}$ is equal to the related eigenvalue of $\boldsymbol{O}_o$. Therefore, each term in a column shows the contribution of the corresponding DOF to that eigenvalue. If we post-multiply matrix $\boldsymbol{G}$ by the inverse matrix of the eigenvalues as

$$\boldsymbol{F}_E = (\boldsymbol{\phi}_p \boldsymbol{\psi}) \circ (\boldsymbol{\phi}_p \boldsymbol{\psi})[\lambda \boldsymbol{I}]^{-1}, \tag{4.11}$$

then each direction in AIS is equally important. In this matrix, the ith element in j^{th} column shows the fractional contribution of the ith DOF in j^{th} eigenvalue. Summing up the elements in each row of the matrix $\boldsymbol{F}_E$, we have

$$\boldsymbol{E}_D = \left[\sum_{j=1}^{k} \boldsymbol{F}_{E1j} : \sum_{j=1}^{k} \boldsymbol{F}_{E2j} : \cdots : \sum_{j=1}^{k} \boldsymbol{F}_{Esj}\right], \tag{4.12}$$

where s is the number of rows and k is the number of columns in the matrix $\boldsymbol{F}_E$, and $\boldsymbol{F}_{Eij}$ shows the jth element in i^{th} row of $\boldsymbol{F}_E$. $\boldsymbol{E}_D$ is a column vector called Effective Independence Distribution vectore [82]. This vector can be written as the diagonal elements of

$$\boldsymbol{EI} = \boldsymbol{\phi}_p \boldsymbol{\psi} \lambda^{-1} \boldsymbol{\phi}_p^T \boldsymbol{\psi}^T. \tag{4.13}$$

Using the orthogonality conditions in Eq. (4.9), we have

$$\boldsymbol{EI} = \boldsymbol{\phi}_p \left[\boldsymbol{\phi}_p^T \boldsymbol{\phi}_p\right]^{-1} \boldsymbol{\phi}_p^T. \tag{4.14}$$

This matrix is an idempotent matrix meaning that $\boldsymbol{EI}^2 = \boldsymbol{EI}$. As we know, the diagonal idempotent matrices are equal to their rank [84]. Thus, the diagonal elements of $\boldsymbol{EI}$ demonstrate the contribution of the related DOF to the rank of the mode shape matrix or, as assumed before, the linear independence of its columns.

If we inspect the diagonal elements of matrix $\boldsymbol{EI}$, we will see that

$$0 \leq \boldsymbol{EI}_{ii} \leq 1. \tag{4.15}$$

If $\boldsymbol{EI}_{ii} = 0$, it means that DOF does not contribute to the dynamic of the part, and if $\boldsymbol{EI}_{ii} = 1$, it means that DOF is crucial for identifying the modes and their linear independence.

We can rank the potential master DOFs and identify the DOFs that do not contribute to the kinetic energy of the part significantly, and then eliminate these DOFs from the mode

shape matrix, $\boldsymbol{\phi}_p$, in an iterative procedure. This way, we can eliminate the DOFs in a sub-optimal manner and reach a limited number of DOFs as the master DOFs. Since we are using this formula repeatedly for removed material and the initial workpiece, we can rewrite it as

$$\boldsymbol{E}_{D_i}^{str} = diag\left(\boldsymbol{\phi}_{np\ i}^{str}\left[\left[\boldsymbol{\phi}_{np\ i}^{str}\right]^T \boldsymbol{\phi}_{np\ i}^{str}\right]^{-1}\left[\boldsymbol{\phi}_{np\ i}^{str}\right]^T\right). \tag{4.16}$$

where $\boldsymbol{E}_{D_i}^{str}$ is the vector of effective independence distribution for the structure of interest, and $\boldsymbol{\phi}_{np\ i}^{str}$ is the matrix of the mode shapes for the n total number of structure's DOFs containing the first p significant modes for the same structure at i^{th} iteration [81].

It must be mentioned that the values of effective independence for the remaining DOFs change at each iteration so that the summation of each column's elements remains equal to the number of significant modes. As each ineffective DOF (slave DOF) is removed, other DOFs become more or less important, and even the order of importance might change. The best approach for finding the master DOFs is eliminating one ineffective DOF at each iteration, so the change in each DOF's effective independence value is minimal. However, it is very time-consuming, and in most cases, more than one DOF can be removed.

To reduce the computation load and calculate the effective independence distribution vector faster, we can decompose the mode shape matrix into [81]

$$\boldsymbol{\phi}_{np\ i}^{str} = \boldsymbol{Q}_{np\ i}^{str}\boldsymbol{R}_{np\ i}^{str}. \tag{4.17}$$

This decomposition is called the QR decomposition method [85], where $\boldsymbol{Q}$ is a unitary orthogonal matrix and $\boldsymbol{R}$ is an upper triangular matrix. Substituting Eq. (4.17) into Eq. (4.14) gives

$$
\begin{aligned}
EI &= \boldsymbol{Q}_{np\,i}^{str}\boldsymbol{R}_{np\,i}^{str}\left[\left[\boldsymbol{Q}_{np\,i}^{str}\boldsymbol{R}_{np\,i}^{str}\right]^T\boldsymbol{Q}_{np\,i}^{str}\boldsymbol{R}_{np\,i}^{str}\right]^{-1}\left[\boldsymbol{Q}_{np\,i}^{str}\boldsymbol{R}_{np\,i}^{str}\right]^T \\
&= \boldsymbol{Q}_{np\,i}^{str}\boldsymbol{R}_{np\,i}^{str}\left[\left[\boldsymbol{R}_{np\,i}^{str}\right]^T\left[\boldsymbol{Q}_{np\,i}^{str}\right]^T\boldsymbol{Q}_{np\,i}^{str}\boldsymbol{R}_{np\,i}^{str}\right]^{-1}\left[\boldsymbol{R}_{np\,i}^{str}\right]^T\left[\boldsymbol{Q}_{np\,i}^{str}\right]^T \\
&= \boldsymbol{Q}_{np\,i}^{str}\boldsymbol{R}_{np\,i}^{str}\left[\left[\boldsymbol{R}_{np\,i}^{str}\right]^T\boldsymbol{I}\boldsymbol{R}_{np\,i}^{str}\right]^{-1}\left[\boldsymbol{R}_{np\,i}^{str}\right]^T\left[\boldsymbol{Q}_{np\,i}^{str}\right]^T \\
&= \boldsymbol{Q}_{np\,i}^{str}\boldsymbol{R}_{np\,i}^{str}\left[\boldsymbol{R}_{np\,i}^{str}\right]^{-1}\left[\left[\boldsymbol{R}_{np\,i}^{str}\right]^T\right]^{-1}\left[\boldsymbol{R}_{np\,i}^{str}\right]^T\left[\boldsymbol{Q}_{np\,i}^{str}\right]^T \\
&= \boldsymbol{Q}_{np\,i}^{str}\left[\boldsymbol{Q}_{np\,i}^{str}\right]^T.
\end{aligned}
\tag{4.18}
$$

And consequently, we have

$$
\boldsymbol{E}_{D_i}^{str} = diag\left(\boldsymbol{Q}_{np\,i}^{str}\left[\boldsymbol{Q}_{np\,i}^{str}\right]^T\right) \tag{4.19}
$$

All the DOFs are initially considered the potential master DOFs, and at each iteration, $\boldsymbol{E}_{D_i}^{str}$ is sorted in descending order. The DOFs, which constitute 90% of the $\boldsymbol{E}_{D_i}^{str}$, which means they constitute 90% of the kinetic energy of the part, are selected as the new set of master DOFs and carried to the next iteration. In this method, the number of master DOFs is unknown. However, the rank and the condition number of the mode shape matrix are monitored at each iteration as the stopping criteria. These criteria are

$$
rank\left(\boldsymbol{\phi}_{np\,i}^{str}\right) = p \;\text{ and }\; cond\left(\boldsymbol{\phi}_{np\,i}^{str}\right) \leq C_{cond}. \tag{4.20}
$$

Here, $rank(\,)$ and $cond(\,)$ evaluate the rank and condition number of the given matrix. These two measurements show the linear independency among the columns of $\boldsymbol{\phi}_{np\ i}^{str}$ [46]. They also describe the spatial independence level in the remained part of the mode shapes in the elimination process.

4.2.3 Reduction Process

The primary purpose of using the model order reduction method is to reduce the size of the displacement vector and, as a result, the FRF of the part. Order reduction can be applied to both the removed material and the workpiece itself.

The mathematical expression of this method is

$$\boldsymbol{x} = \begin{Bmatrix} \boldsymbol{x}_m \\ \boldsymbol{x}_s \end{Bmatrix} = \boldsymbol{T}_{red}\boldsymbol{x}_m, \tag{4.21}$$

where s demonstrates the slave DOFs, m represents the master DOFs, and $\boldsymbol{T}_{red}$ is the reduction transformation matrix with the size of $(s+m)\times m$. In addition to the displacement vector, the system matrices and the force vector must be partitioned too. Hence, we can rewrite the equation of motions as

$$\begin{bmatrix} \boldsymbol{M}_{mm} & \boldsymbol{M}_{ms} \\ \boldsymbol{M}_{sm} & \boldsymbol{M}_{ss} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{x}}_m \\ \ddot{\boldsymbol{x}}_s \end{Bmatrix} + \begin{bmatrix} \boldsymbol{K}_{mm} & \boldsymbol{K}_{ms} \\ \boldsymbol{K}_{sm} & \boldsymbol{K}_{ss} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_m \\ \boldsymbol{x}_s \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_m \\ \boldsymbol{f}_s \end{Bmatrix}. \tag{4.22}$$

By inserting Eq. (4.21) into this equation and pre-multiplying by $\boldsymbol{T}_{red}^T$, the reduced equation of motions can be obtained. Thus, the reduced mass and stiffness matrices are

$$\boldsymbol{M}_{red} = \boldsymbol{T}_{red}^{T} \begin{bmatrix} \boldsymbol{M}_{mm} & \boldsymbol{M}_{ms} \\ \boldsymbol{M}_{sm} & \boldsymbol{M}_{ss} \end{bmatrix} \boldsymbol{T}_{red} \quad , \quad \boldsymbol{K}_{red} = \boldsymbol{T}_{red}^{T} \begin{bmatrix} \boldsymbol{K}_{mm} & \boldsymbol{K}_{ms} \\ \boldsymbol{K}_{sm} & \boldsymbol{K}_{ss} \end{bmatrix} T_{red} \tag{4.23}$$

The size of these system matrices is $m \times m$.

In order to perform the model order reduction, we first need to obtain $\boldsymbol{T}_{red}$ for the different parts of the workpiece.

4.2.4 Original Part

In order to reduce the order of the original part before any material removal, first we need to find the reduction transformation matrix, $\boldsymbol{T}^{B_0}$. Calculating the transformation matrix for all the modes, which can be a very high number for a model with a fine mesh grid, is very time-consuming. To avoid high computation time, a limited number of modes are considered in the calculation. In this case, the System Equivalent Reduction and Expansion Process (SEREP) [86] is the most suitable method to obtain the transformation matrix.

4.2.4.1 SEREP

System Equivalent Reduction and Expansion Process or SEREP is a high accuracy reduction method that uses the model's eigenvalues. In this method, a selected number of modes and the master DOFs of the complete model are used to find a smaller space with lower dimensions and calculate the right and left transformation matrices ($\boldsymbol{T}_R^{B_0}$ and $\boldsymbol{T}_L^{B_0}$). When the mass and stiffness matrices are symmetric, each of these transformation matrices is equal to the transpose of the other one.

In order to derive the equations, first, the complete model is used, and the eigenvalue problem is solved. With symmetric mass and stiffness matrices, a diagonal matrix containing the eigenvalues ($\boldsymbol{\beta}$), and a matrix containing the eigenvectors ($\boldsymbol{\phi}$) are calculated as the eigenpair. For clarification, we assume that the mass and stiffness matrices are not symmetric. In this case, the problem has one matrix containing left eigenvectors ($\boldsymbol{V}$) and another one containing the right eigenvectors ($\boldsymbol{U}$) with the same size as the system matrices. These vectors are biorthonormal, which means $\boldsymbol{V}^T\boldsymbol{M}\boldsymbol{U} = \boldsymbol{I}$ and $\boldsymbol{V}^T\boldsymbol{K}\boldsymbol{U} = \boldsymbol{\beta}$. Next, the columns of these matrices, which associate with the significant modes, are extracted and shown by $\bar{U}$ and $\bar{V}$ with a size of $n \times p$, where n represents the total number of DOFs and p shows the number of selected significant modes.

Using $\bar{U}$, the generalized or modal DOFs ($\boldsymbol{\alpha}$) can be obtained from physical DOFs ($\boldsymbol{x}$) as

$$\boldsymbol{x} = \bar{\boldsymbol{U}}\boldsymbol{\alpha}. \tag{4.24}$$

In this equation, $\boldsymbol{x}$ is a vector with n elements, and $\boldsymbol{\alpha}$ is a vector with p elements. If we partition $\boldsymbol{x}$ and $\bar{\boldsymbol{U}}$ into master and slave DOFs, the vector of modal DOFs can be obtained based on the master DOFs as

$$\boldsymbol{x} = \begin{Bmatrix} \boldsymbol{x}_m \\ \boldsymbol{x}_s \end{Bmatrix} = \begin{bmatrix} \bar{\boldsymbol{U}}_m \\ \bar{\boldsymbol{U}}_s \end{bmatrix} \boldsymbol{\alpha} = \bar{\boldsymbol{U}}\boldsymbol{q}, \tag{4.25}$$

$$\boldsymbol{x}_m = \bar{\boldsymbol{U}}_m\boldsymbol{\alpha}, \tag{4.26}$$

$$\boldsymbol{\alpha} = \bar{\boldsymbol{U}}_m^+\boldsymbol{x}_m, \tag{4.27}$$

where $\overline{\boldsymbol{U}}_m^+$ is the generalized inverse of $\overline{\boldsymbol{U}}_m$. By substituting Eq. (4.27) into Eq. (4.24), the right transformation matrix can be calculated as

$$\boldsymbol{x} = \overline{\boldsymbol{U}}\overline{\boldsymbol{U}}_m^+\boldsymbol{x}_m = \boldsymbol{T}_R\boldsymbol{x}_m. \tag{4.28}$$

In order to find the left transformation matrix, Eq. (4.28) must be inserted into the equation of motion, which is

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{f}. \tag{4.29}$$

Then we have

$$\boldsymbol{M}\overline{\boldsymbol{U}}\overline{\boldsymbol{U}}_m^+\ddot{\boldsymbol{x}}_m + \boldsymbol{K}\overline{\boldsymbol{U}}\overline{\boldsymbol{U}}_m^+\boldsymbol{x}_m = \boldsymbol{f} \tag{4.30}$$

If this equation is pre-multiplied by $\overline{\boldsymbol{V}}^T$, then we have

$$\overline{\boldsymbol{V}}^T\boldsymbol{M}\overline{\boldsymbol{U}}\overline{\boldsymbol{U}}_m^+\ddot{\boldsymbol{x}}_m + \overline{\boldsymbol{V}}^T\boldsymbol{K}\overline{\boldsymbol{U}}\overline{\boldsymbol{U}}_m^+\boldsymbol{x}_m = \overline{\boldsymbol{V}}^T\boldsymbol{f}, \tag{4.31}$$

$$\overline{\boldsymbol{U}}_m^+\ddot{\boldsymbol{x}}_m + \overline{\boldsymbol{\beta}}\overline{\boldsymbol{U}}_m^+\boldsymbol{x}_m = \overline{\boldsymbol{V}}^T\boldsymbol{f}. \tag{4.32}$$

By pre-multiplying Eq. (4.32) by $(\overline{\boldsymbol{V}}_m^T)^+$, we have

$$[\overline{\boldsymbol{V}}_m^T]^+\overline{\boldsymbol{U}}_m^+\ddot{\boldsymbol{x}}_m + [\overline{\boldsymbol{V}}_m^T]^+\overline{\boldsymbol{\beta}}\overline{\boldsymbol{U}}_m^+\boldsymbol{x}_m = [\overline{\boldsymbol{V}}_m^T]^+\overline{\boldsymbol{V}}^T\boldsymbol{f}. \tag{4.33}$$

If we consider $[\overline{\boldsymbol{V}}_m^T]^+\overline{\boldsymbol{U}}_m^+ = \boldsymbol{M}_r$ and $[\overline{\boldsymbol{V}}_m^T]^+\overline{\boldsymbol{\beta}}\overline{\boldsymbol{U}}_m^+ = \boldsymbol{K}_r$, then we have

$$\boldsymbol{M}_r\ddot{\boldsymbol{x}}_m + \boldsymbol{K}_r\boldsymbol{x}_m = [\overline{\boldsymbol{V}}_m^T]^+\overline{\boldsymbol{V}}^T\boldsymbol{f} = \boldsymbol{T}_L\boldsymbol{f}, \tag{4.34}$$

where $\boldsymbol{T}_L$ is the left transformation matrix.

In these formulations, we need to calculate the generalized inverse of matrices several times. The generalized inverse is used when the matrix is not square [87]. It means the sizes of the matrix in two directions are different. For example, for matrix $\boldsymbol{G}_{ns}$ with size $a \times d$, two scenarios may occur, which can be solved with the Moore-Penrose pseudo-inverse algorithm [88], [89] as

$$\boldsymbol{G}_{ns}{}^{+} = \left[\boldsymbol{G}_{ns}{}^{T}\boldsymbol{G}_{ns}\right]^{-1}\boldsymbol{G}_{ns}{}^{T},\ a > d, \tag{4.35}$$

$$\boldsymbol{G}_{ns}{}^{+} = \boldsymbol{G}_{ns}{}^{T}\left[\boldsymbol{G}_{ns}\boldsymbol{G}_{ns}{}^{T}\right]^{-1},\ a < d. \tag{4.36}$$

After calculating the left and right transformation matrices, the reduced mass and stiffness matrices can be obtained as

$$\boldsymbol{M}_{red} = \boldsymbol{T}_{L}\boldsymbol{M}\boldsymbol{T}_{R} \tag{4.37}$$

$$\boldsymbol{K}_{red} = \boldsymbol{T}_{L}\boldsymbol{K}\boldsymbol{T}_{R}. \tag{4.38}$$

The left and right eigenvectors are equal if the mass and stiffness matrices are symmetric. It means $\overline{\boldsymbol{U}}_{m}^{+} = \overline{\boldsymbol{V}}_{m}^{+} = \overline{\boldsymbol{\phi}}_{m}^{+}$, so we have [90]

$$\boldsymbol{T}_{R} = \overline{\boldsymbol{U}}\overline{\boldsymbol{U}}_{m}^{+} = \overline{\boldsymbol{\phi}}\overline{\boldsymbol{\phi}}_{m}^{+}, \tag{4.39}$$

$$\boldsymbol{T}_{L} = [\overline{\boldsymbol{V}}_{m}^{T}]^{+}\overline{\boldsymbol{V}}^{T} = [\overline{\boldsymbol{V}}_{m}^{+}]^{T}\overline{\boldsymbol{V}}^{T} = [\overline{\boldsymbol{U}}_{m}^{+}]^{T}\overline{\boldsymbol{U}}^{T} = [\overline{\boldsymbol{U}}\overline{\boldsymbol{U}}_{m}^{+}]^{T} = \boldsymbol{T}_{R}{}^{T}. \tag{4.40}$$

As a result, in Eq. (4.37) and (4.38), we have

$$\boldsymbol{M}_{red} = \boldsymbol{T}_{R}{}^{T}\boldsymbol{M}\boldsymbol{T}_{R} = \boldsymbol{T}_{L}\boldsymbol{M}\boldsymbol{T}_{L}{}^{T}, \tag{4.41}$$

$$\boldsymbol{K}_{red} = \boldsymbol{T}_{R}{}^{T}\boldsymbol{K}\boldsymbol{T}_{R} = \boldsymbol{T}_{L}\boldsymbol{K}\boldsymbol{T}_{L}{}^{T}. \tag{4.42}$$

Also, if we partition $\boldsymbol{\phi}$ into $\boldsymbol{\phi}_m$ and $\boldsymbol{\phi}_s$, then we can calculate the transformation matrix as

$$\bar{\boldsymbol{\phi}}\bar{\boldsymbol{\phi}}_m^+ = \begin{bmatrix} \boldsymbol{\phi}_m \\ \boldsymbol{\phi}_s \end{bmatrix} \bar{\boldsymbol{\phi}}_m^+ = \begin{bmatrix} \boldsymbol{\phi}_m \bar{\boldsymbol{\phi}}_m^+ \\ \boldsymbol{\phi}_s \bar{\boldsymbol{\phi}}_m^+ \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{mm} \\ \boldsymbol{\phi}_s [\boldsymbol{\phi}_m]^+ \end{bmatrix}, \tag{4.43}$$

$$\boldsymbol{T}_{reduction,SEREP} = \begin{bmatrix} \boldsymbol{I}_{mm} \\ \boldsymbol{\phi}_s [\boldsymbol{\phi}_m]^+ \end{bmatrix}.$$

This equation needs fewer computations and is more cost-effective.

4.2.5 Removed Volumes

This section explains the methods used for reducing the order of removed material. Since the fictitious substructures are small parts, we need a method that does not include dynamic truncation. Hence, the SEREP method is not suitable for reducing the order of removed material. To overcome this challenge, Improved Reduction System (IRS) [91], [92] method, and Guyan reduction algorithm [93] are used for order reduction for the removed volumes.

4.2.5.1 Guyan

Guyan or static reduction is a straightforward method that only uses the stiffness matrix to calculate the transformation matrix. This method is usually used as the baseline for evaluating the results of other methods. The computational cost of this method is meager, and the results are relatively good. However, the level of accuracy is not high enough to be used as an independent reduction method. This method is used as the initial estimation for the IRS method in this study.

If we assume the effect of mass or inertia in the equation of motion is negligible, then we can partition this equation as

$$\begin{bmatrix} \boldsymbol{K}_{mm} & \boldsymbol{K}_{ms} \\ \boldsymbol{K}_{sm} & \boldsymbol{K}_{ss} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_m \\ \boldsymbol{x}_s \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_m \\ \boldsymbol{0} \end{Bmatrix}. \tag{4.44}$$

From the second row, $\boldsymbol{x}_s$ can be defined as a function of $\boldsymbol{x}_m$. So we have [93]

$$\boldsymbol{K}_{sm}\boldsymbol{x}_m + \boldsymbol{K}_{ss}\boldsymbol{x}_s = 0, \tag{4.45}$$

$$\boldsymbol{x} = \begin{Bmatrix} \boldsymbol{x}_m \\ \boldsymbol{x}_s \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{x}_m \\ -\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm}\boldsymbol{x}_m \end{Bmatrix} = \begin{bmatrix} \boldsymbol{I}_{mm} \\ -\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm} \end{bmatrix} \boldsymbol{x}_m = \boldsymbol{T}_G \boldsymbol{x}_m, \tag{4.46}$$

$$\boldsymbol{T}_G = \begin{bmatrix} \boldsymbol{I}_{mm} \\ \boldsymbol{t}_G \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{mm} \\ -\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm} \end{bmatrix}. \tag{4.47}$$

where $\boldsymbol{T}_G$ is Guyan reduction transformation matrix. Using this transformation matrix, the reduced mass and stiffness matrices can be calculated as

$$\boldsymbol{M}_{red,G} = \boldsymbol{T}_G^T \boldsymbol{M} \boldsymbol{T}_G \tag{4.48}$$

$$\boldsymbol{K}_{red,G} = \boldsymbol{T}_G^T \boldsymbol{K} \boldsymbol{T}_G \tag{4.49}$$

These reduced matrices are used as the initial estimation in the following method.

4.2.5.2 IRS

Now that we have Guyan transformation matrix and reduced mass and stiffness matrices, we can use them in IRS, which is an iterative approach, to improve the results. This method changes the static transformation by including the mass matrix as pseudo-static forces. In Eq. (4.50), a binomial series expansion is used to generate the transformation matrix for IRS based on the reduced stiffness and mass matrices by the Guyan method [92]. This transformation matrix is shown in Eq. (4.51).

$$\boldsymbol{t}_{IRS,1} = \boldsymbol{t}_G + \boldsymbol{K}_{ss}^{-1}(\boldsymbol{M}_{sm} + \boldsymbol{M}_{ss}\boldsymbol{t}_G)\boldsymbol{M}_{red,G}^{-1}\boldsymbol{K}_{Red,G}^{-1}, \tag{4.50}$$

$$\boldsymbol{T}_{IRS,1} = \begin{bmatrix} \boldsymbol{I}_{mm} \\ \boldsymbol{t}_{IRS,1} \end{bmatrix}. \tag{4.51}$$

Once the initial estimation for the transformation matrix is calculated, a more accurate approximation for the reduced mass and stiffness matrices is available using the $T_{IRS,j}$ as

$$\boldsymbol{M}_{red,IRS_j} = \boldsymbol{T}_{IRS,j}^T \boldsymbol{M}\boldsymbol{T}_{IRS,j} \tag{4.52}$$

$$\boldsymbol{K}_{red,IRS_j} = \boldsymbol{T}_{IRS,j}^T \boldsymbol{K}\boldsymbol{T}_{IRS,j} \tag{4.53}$$

By replacing $\boldsymbol{M}_{Red,G}$ and $\boldsymbol{K}_{Red,G}$ in Eq. (4.50) with these reduced mass and stiffness matrices, the reduction transformation can be iteratively improved by using

$$\boldsymbol{t}_{IRS,j+1} = \boldsymbol{t}_G + \boldsymbol{K}_{ss}^{-1}\left(\boldsymbol{M}_{sm} + \boldsymbol{M}_{ss}\boldsymbol{t}_{IRS,j}\right)\boldsymbol{M}_{red,IRS_j}^{-1}\boldsymbol{K}_{Red,IRS_j}^{-1}, \tag{4.54}$$

where j denotes the iteration number. After each iteration, the resulting transformation matrix has better accuracy than the previous iteration. However, this procedure is done repetitively until the convergence criterion in Eq. (4.55) is met.

$$\frac{\left|\left(\omega_{n,a}\right)_{j+1} - \left(\omega_{n,a}\right)_j\right|}{\left(\omega_{n,a}\right)_j} \leq \varepsilon_{Tol} \tag{4.55}$$

In this criterion, $\left(\omega_{n,a}\right)_j$ is the mode a of the system after j iterations, and convergence tolerance is represented by ε_{Tol}.

The resulting stiffness and mass matrices belong to the master DOFs, which contribute the most to the part's dynamics in the selected significant modes where the tool has contact with the workpiece.

4.3 FRF Calculation with Mode Superposition

The last step of structural analysis is calculating the FRF. FRF can be determined either by solving a set of linear equations or by the mode superposition method. The structures with a limited number of nodes and DOFs result in a small set of equations; however, this set of equations is enormous for the structures formed by a large group of nodes. In the first case, using the classic equation-solving method to calculate the FRF has a less computational load. For large structures, on the other hand, it is more efficient to use the mode superposition method to reduce the computational load and time. In this study, we are developing an algorithm to obtain the FRF of any large structure in a timely manner; as a result, instead of using classic equation-solving methods, we are using mode superposition.

Using dynamic stiffness for the fictitious substructures (removed material) prevents dynamics truncation. However, modal truncation still happens in the workpiece if we only consider the low-frequency modes.

In this method, a limited number of modes are used to calculate the FRF of the workpiece. Although this method has computational privileges, the accuracy of the coupling and decoupling is unfavorably influenced by modal incompleteness [94]. In order to overcome this challenge, the FRF of the initial structure must be recompensed for the contribution of the higher-order modes' frequency-dependent dynamics [95]. This process

can be performed without any explicit solution by utilizing the matrix of mode shapes of the initial workpiece and a few known low-frequency modes.

We can write the FRF as a superposition of low-frequency and high-frequency components. The formulation of this superposition for the initial workpiece is

$$\boldsymbol{H}^{B_0} = \boldsymbol{H}_L^{B_0} + \boldsymbol{H}_H^{B_0} = [\boldsymbol{K}^{B_0} - \omega^2 \boldsymbol{M}^{B_0}]^{-1}. \tag{4.56}$$

In this equation, the contribution of the first L low-frequency modes is shown by $\boldsymbol{H}_L^{B_0}$, and $\boldsymbol{H}_H^{B_0}$ represent the truncated modes with high frequencies. The contribution of low-frequency modes is [96]

$$\boldsymbol{H}_L^{B_0} = \boldsymbol{\phi}_L^{B_0} \left[\boldsymbol{\Lambda}_L^{B_0} - \omega^2 \boldsymbol{I}\right]^{-1} \left[\boldsymbol{\phi}_L^{B_0}\right]^T. \tag{4.57}$$

In this equation $\boldsymbol{\phi}_L^{B_0}$ represents the matrix of mode shapes and $\boldsymbol{\Lambda}_L^{B_0}$ shows the corresponding eigenvalues of the first L low-frequency modes organized as a diagonal matrix. By substituting Eq. (4.57) into Eq. (4.56), the contribution of the truncated high-frequency mode can be evaluated as

$$\boldsymbol{H}_H^{B_0} = [\boldsymbol{K}^{B_0} - \omega^2 \boldsymbol{M}^{B_0}]^{-1} - \boldsymbol{\phi}_L^{B_0} \left[\boldsymbol{\Lambda}_L^{B_0} - \omega^2 \boldsymbol{I}\right]^{-1} \left[\boldsymbol{\phi}_L^{B_0}\right]^T. \tag{4.58}$$

We can write the matrix inversions of this equation in the form of their power series expansion [95]–[97] as

$$
\begin{aligned}
[\boldsymbol{K}^{B_0} - \omega^2 \boldsymbol{M}^{B_0}]^{-1} &= [\boldsymbol{K}^{B_0}]^{-1}[I - \omega^2 \boldsymbol{M}^{B_0}[\boldsymbol{K}^{B_0}]^{-1}]^{-1} \\
&= [\boldsymbol{K}^{B_0}]^{-1} \sum_{n=0}^{\infty} (\omega^2)^n (\boldsymbol{M}^{B_0})^n ([\boldsymbol{K}^{B_0}]^{-1})^n,
\end{aligned}
\tag{4.59}
$$

and

$$
\begin{aligned}
\boldsymbol{\phi}_L^{B_0}\left[\boldsymbol{\Lambda}_L^{B_0} - \omega^2 \boldsymbol{I}\right]^{-1}\left[\boldsymbol{\phi}_L^{B_0}\right]^T &= \boldsymbol{\phi}_L^{B_0}\left[\boldsymbol{\Lambda}_L^{B_0}\right]^{-1}\left[\boldsymbol{I} - \omega^2\left[\boldsymbol{\Lambda}_L^{B_0}\right]^{-1}\right]^{-1}\left[\boldsymbol{\phi}_L^{B_0}\right]^T \\
&= \boldsymbol{\phi}_L^{B_0}\left[\boldsymbol{\Lambda}_L^{B_0}\right]^{-1} \sum_{n=0}^{\infty} (\omega^2)^n \left(\left[\boldsymbol{\Lambda}_L^{B_0}\right]^{-1}\right)^n \left[\boldsymbol{\phi}_L^{B_0}\right]^T.
\end{aligned}
\tag{4.60}
$$

Substituting Eq. (4.59) and (4.60) in Eq. (4.58), the truncated contribution of high-frequency modes is evaluated as

$$
\begin{aligned}
\boldsymbol{H}_H^{B_0} &= [\boldsymbol{K}^{B_0}]^{-1} \sum_{n=0}^{\infty} (\omega^2)^n (\boldsymbol{M}^{B_0})^n ([\boldsymbol{K}^{B_0}]^{-1})^n \\
&- \boldsymbol{\phi}_L^{B_0}\left[\boldsymbol{\Lambda}_L^{B_0}\right]^{-1} \sum_{n=0}^{\infty} (\omega^2)^n \left(\left[\boldsymbol{\Lambda}_L^{B_0}\right]^{-1}\right)^n \left[\boldsymbol{\phi}_L^{B_0}\right]^T.
\end{aligned}
\tag{4.61}
$$

By simplifying this equation, we have [95]–[97]

$$
\boldsymbol{H}_H^{B_0} = \boldsymbol{H}_{H0}^{B_0} \sum_{n=0}^{\infty} \omega^{2n} \left[\boldsymbol{M}^{B_0} \boldsymbol{H}_{H0}^{B_0}\right]^n \tag{4.62}
$$

where $\boldsymbol{H}_{H0}^{B_0}$ represents the static compensation term at $\omega = 0$, which is

$$
\boldsymbol{H}_{H_0}^{B_0} = [\boldsymbol{K}^{B_0}]^{-1} - \boldsymbol{\phi}_L^{B_0}\left[\boldsymbol{\Lambda}_L^{B_0}\right]^{-1}\left[\boldsymbol{\phi}_L^{B_0}\right]^T. \tag{4.63}
$$

This static compensation term cannot compensate sufficiently for the complete frequency range. As a result, the compensation term is augmented by considering higher-order modes of Eq. (4.62). If the upper boundary of the targeted frequency range satisfies [85]

$$\omega_{upper} < \omega_{n,L}, \tag{4.64}$$

then, only the first two to five terms of the series in Eq. (4.62) are adequate [97]. Also, this equation can be used to determine the number of required low-frequency modes.

Using the obtained formulations and Eq. (4.56), the FRF of the workpiece can be calculated as

$$\boldsymbol{H}^{B_0} = \boldsymbol{\phi}_L^{B_0}\left[\boldsymbol{\Lambda}_L^{B_0} - \omega^2\boldsymbol{I}\right]^{-1}\left[\boldsymbol{\phi}_L^{B_0}\right]^T + \boldsymbol{H}_{H0}^{B_0}\sum_{n=0}^{\infty}\omega^{2n}\left[\boldsymbol{M}^{B_0}\boldsymbol{H}_{H0}^{B_0}\right]^n. \tag{4.65}$$

The benefit of using this method is that we only need to calculate two full-size matrix inversion in Eq. (4.63) for $[\boldsymbol{K}^{B_0}]^{-1}$ and $\left[\boldsymbol{\Lambda}_L^{B_0}\right]^{-1}$. This equation is not frequency dependent, so we do not need to calculate these matrix inversions for each frequency. The other matrix inversions in this method are the inversion of diagonal matrices with a limited number of modes. Even though one of these inversions is frequency-dependent ($\left[\boldsymbol{\Lambda}_L^{B_0} - \omega^2\boldsymbol{I}\right]^{-1}$) and must be calculated for each frequency separately, the computational cost of calculating these inversions is significantly lower than the inversion of non-diagonal full-size matrices.

4.4 Results and Discussion

After implementing all these methods for time efficiency, a simulation is carried out to see if the proposed method is able to update the FRF of the in-process workpiece with

good accuracy in a timely manner. Here, the same geometry illustrated in Chapter 3 with a fine mesh grid with 11448 DOF is used. The removed volumes at each machining step are kept the same as the used model in Chapter 3. The master DOFs of the initial workpiece are selected using the first 300 and 50 modes as the significant modes for B_0 and A_i respectively. In Eq. (4.20), the condition number (C_{cond}) is set equal to 10^2, and in Eq. (4.55), the convergence tolerance for IRS reduction is set to 1%. These considerations reduce the initial model with 11448 DOFs to 1488 master DOFs.

The FRF of the initial workpiece is calculated considering 100 modes as the low-frequency modes which satisfy the convergence criterion in Eq. (4.64). The effect of truncated high-frequency modes is taken into account by using the two terms ($n = 2$) of the power series expansion in Eq. (4.62). The FRF of the initial workpiece is evaluated using the mode superposition method.

The FRF of the workpiece is evaluated at different stages of machining using both the full order model and the reduced order model.

Figure 4-2 shows the FRF of the initial part before machining. As can be seen, the obtained FRF by the reduced model shows good agreement with the results of the full order model.

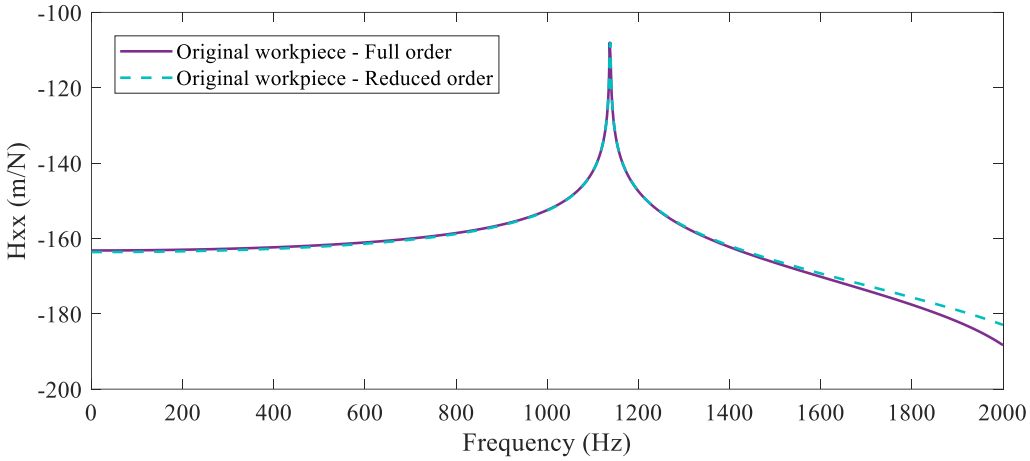


Figure 4-2 FRF of the initial part before machining, calculated by the full order model and the reduced order model

The FRF of the in-process workpiece is updated after each machining step using the proposed solution with the reduced and full order models. These FRFs are presented in Figure 4-3 to Figure 4-5.

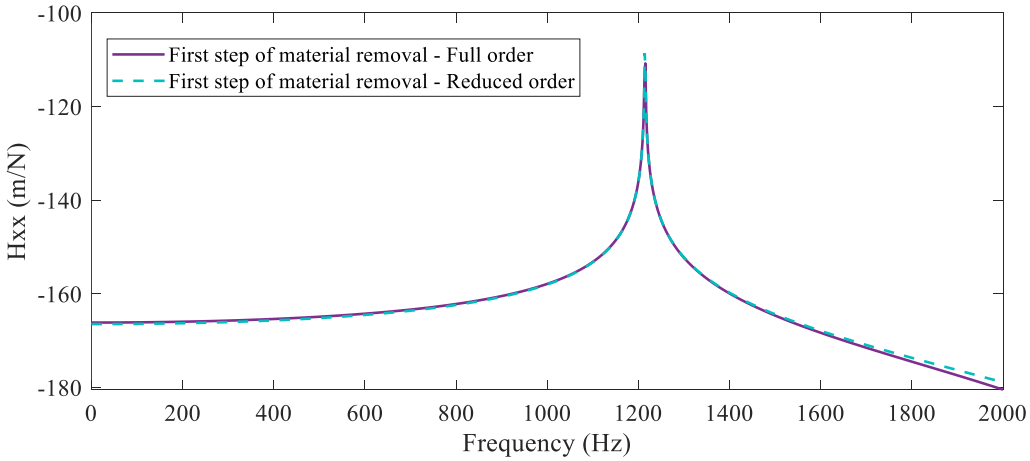


Figure 4-3 FRF of the in-process workpiece after the first machining step, calculated by the full order model and the reduced order model

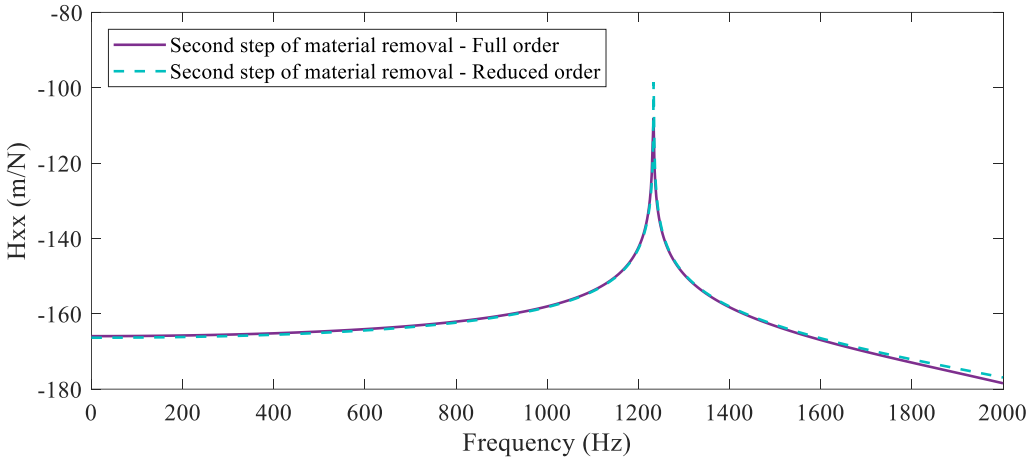


Figure 4-4 FRF of the in-process workpiece after the second machining step, calculated by the full order model and the reduced order model

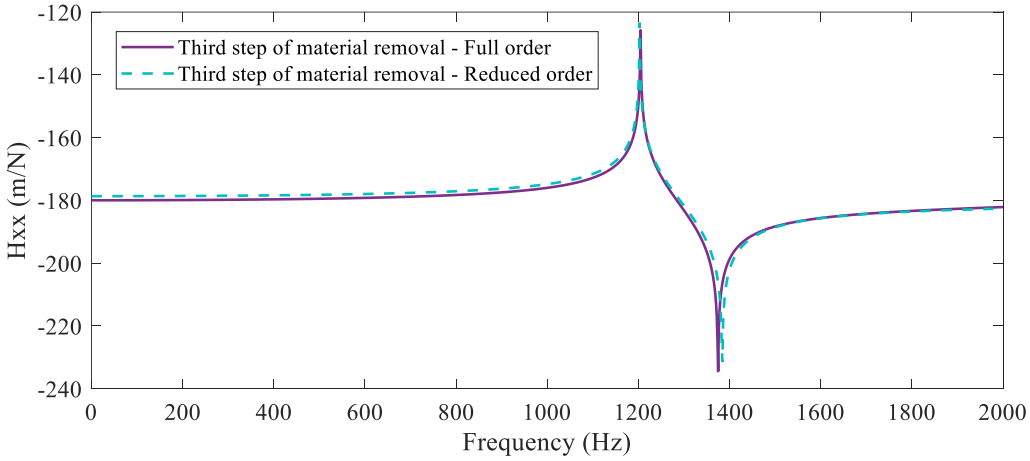


Figure 4-5 FRF of the in-process workpiece after the third machining step, calculated by the full order model and the reduced order model

As shown in the plots, the FRF of the reduced model has great accuracy. The deviation between the FRF of the full order model and the reduced model is mainly a result of missing dynamics in the process of model order reduction. In addition, it must be mentioned that an

increase in the error is observed in the last step of machining. This increase is caused by accumulating smaller errors from previous steps.

It is observed that the increase in the calculation and FRF updating speed is almost proportional to the reduction in the selected DOFs population compared to the full order model. The solution times for the full order model and the improved reduced order model are listed in Table 4-1. As listed in this table, the computation time for the full order model is 103 minutes. Using the improved model with reduced order and mode superposition, the computation time drops to 16 minutes. It must be mentioned that a change in the reduction ratio in the order reduction process leads to a change in computational time.

Table 4-1 Comparison between the solution time for the full order model and improved reduced order model

	Number of Computational DOFs	**Solution Time (min)**
Full Order Model	11448	103
Improved Reduced Order Model	1488	16

Although the developed algorithm is able to calculate the FRF of the workpiece and predict the FRF of the part after the upcoming machining steps, we can use the novel machine learning algorithm for process planning at the fastest rate to be used in online monitoring. This method is covered in the next chapter.

4.5 Conclusion

In this chapter, different methods are used to improve the computation time for the developed algorithm. The first method is model order reduction which reduces the number of computational DOFs. In this method, the DOFs with the most contribution to the model's kinetic energy, called master DOFs, are selected by iterative effective independence. Then, SEREP, Guyan, or IRS methods are used to reflect the effect of the eliminated slave DOFs on the master DOF and reduce the size of the system matrices. These matrices can be used in the solution process with minimum compromisation. The mode superposition is also used as another improvement method. In this method, the FRF of the part is calculated only at low-frequency modes. Then the effect of the truncated modes at high frequencies is added to the calculated FRF. The performance of the improved algorithm is investigated through simulations. It is shown that while the accuracy of results is acceptable, the enhanced model is able to update the FRF of the in-process workpiece much faster than the full order model.

5 Prediction of Varying Dynamics Using Deep Learning

5.1 Overview

Machine learning solutions have been utilized in many applications, such as condition monitoring and stress analysis, to develop a fast and fully automated solution. With the flexibility offered by intelligent manufacturing, machines must be able to plan the process and correct it when necessary, with minimum dependency on human interventions. For instance, a machine must be able to adjust the feed rate or the machining speed based on the feedback from the monitoring system. To do so, the machine must be able to promptly update the structural dynamics information of the workpiece to make a corrective decision and optimize the machining parameters as needed.

Although the proposed FE based algorithm in the previous chapters is able to predict the dynamics of the in-process workpiece ahead of time very accurately, we still need a much faster prediction method to be used in online process planning. To overcome this challenge, in this chapter, a novel deep learning based framework is developed that can instantly predict the required dynamics of the workpiece for chatter prevention. Section 5.2 explains the procedure for data generation. The design of a CNN for deep learning algorithm and the training process of this CNN are explained in Sections 5.3 and 5.4, respectively. The results

of testing this framework on an unseen dataset are presented in Section 5.5, and the chapter is concluded in Section 5.6. A derivation of this chapter is published in [98].

5.2 Data Generation

To be able to predict the dynamics of a workpiece with a deep learning model, first, we need a dataset to train a CNN. This project aims to predict a workpiece's natural frequencies at different machining stages. As a result, the required dataset must combine different geometries as inputs and their corresponding natural frequencies as the outputs.

5.2.1 Modeling the Geometries

In order to generate the geometries, the PythonOCC library [99] is used to develop an automatic algorithm. This algorithm generates an extensive dataset consisting of CAD files covering the workpiece's possible geometries throughout machining processes such as grooving and turning.

For the geometry modeling, a cylinder with a diameter of 100 mm and a length of 500 mm is considered as the initial part. In order to illustrate the boundary condition at the actual spindle chuck, the boundary condition for this cylinder is considered clamped at the base of the cylinder at $z = 0$. Then, this cylinder is divided into eight axial and four radial segments. Removing these segments in different orders allows us to simulate different geometries of an in-process workpiece. However, in order to use only the practical geometries, the following criteria must be met in all the segment removal scenarios:

- All the radial segments of the first three axial segments are retained

- For each axial segment, at least the first radial segment is retained.

The segmentation of the geometry is shown in Figure 5-1. Considering these two rules, a total number of 1024 possible geometries are modeled.

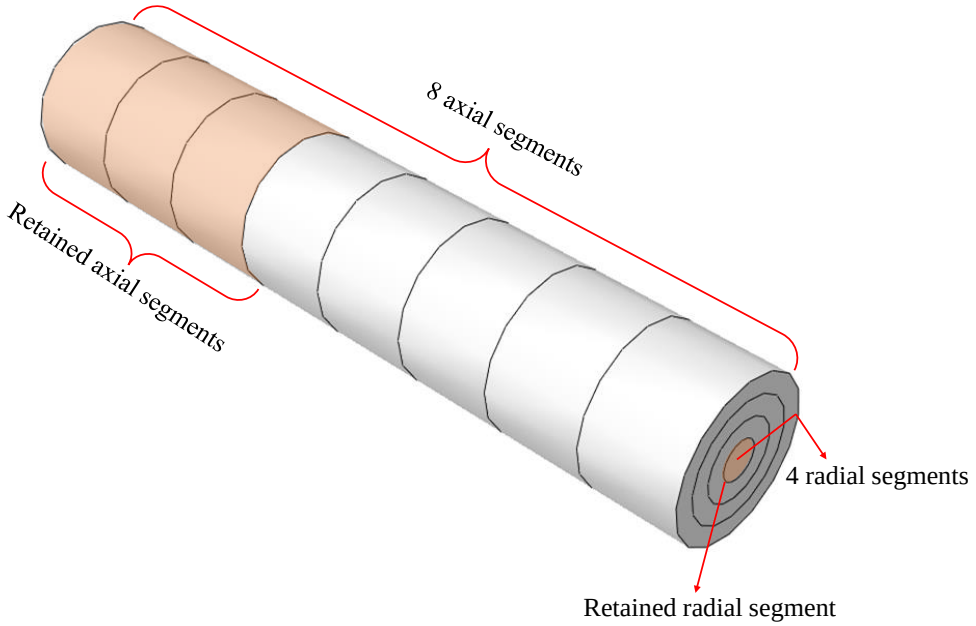


Figure 5-1 Radial and axial segmentation of the geometries

5.2.2 Eigenproblem Solution

The next step of data generation is to generate the outputs for the dataset. To do so, we need to calculate the natural frequencies of the generated geometries.

For this task, an FE algorithm is developed in MATLAB, which automatically generates the mesh grids for all geometries. To obtain the outputs for the dataset, these mesh grids must be used to calculate the natural frequencies of the corresponding geometries.

The selected material for this part of the research is the same AISI 1080 Steel as the previous chapters. Once the mass and stiffness matrices are assembled for each geometry,

the developed MATLAB code in Chapter 3 is used to solve the eigenproblem for these geometries. In this research, only the first five non-repeated natural frequencies are saved in NPY files and used as the outputs.

5.2.2.1 Normalization

Scaling the dataset can sometimes improve the performance of a machine learning model [100]. The two most common methods for scaling data are normalizing and standardizing the data. Since standardization may generate negative values, it is unsuitable for our application. This study chooses normalization as the scaling method.

In normalization, the numbers are transferred to the range of $[0\ 1]$. There are several methods for normalizing datasets. One of the most common methods is min-max normalization. In this method, first, the minimum and the maximum values of each feature in the dataset are obtained, then the dataset is normalized by

$$MX_{ji} = \frac{(x_{ji} - min_j)}{(max_j - min_j)}, \tag{5.1}$$

where x_{ji} is the jth feature of ith data point, min_j and max_j are the minimum and maximum values of the jth feature, and X_{ji} is the normalized data point.

In order to normalize the dataset, the `MinMaxScaler` module from sci-kitlearn [101] is used. This module receives the dataset and generates a scaler and an inverse scaler accordingly. Then, the scaler is used to normalize the dataset, and the process can be reversed using the inverse scaler. If a neural network is trained with the normalized dataset, the

prediction results of the test dataset are also normalized. So, we use the inverse scaler to unnormalize the predictions as well.

In this study, only the natural frequencies are normalized.

5.2.3 Voxelization

The deep learning models only accept the input data in a matrix form. As a result, if we want to use the generated geometries as inputs, they must be expressed as an array. A standard method for digitally representing 3D objects is voxelizations.

To transfer the generated geometries to a digital form, a 3D grid is designed in such a size that the largest geometry, the initial part before machining, can fit in it. Then, the geometries are enclosed in the grid, as shown in Figure 5-2 (a).

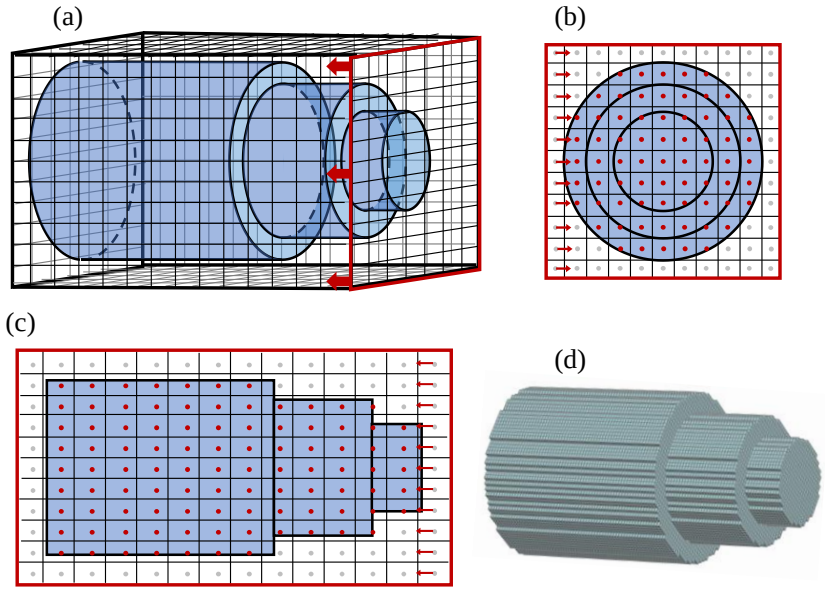


Figure 5-2 (a) Enclosed geometry in a 3D voxel grid, (b) cross-section of the 3D voxel grid and the enclosed geometry, (c) side view of the 3D voxel grid and the enclosed geometry, (d) voxelized geometry

The size of the grid is identical for all the geometries. Each voxel or small cube in the grid contains a binary digit demonstrating whether the center of that voxel is located inside the workpiece. As is shown in Figure 5-2 (b) and Figure 5-2 (c), if the center of the voxel is located inside the geometry, that voxel contains 1, and if not, it contains 0. This method gives us binary tensors to use as input instead of CAD models.

The selected voxel size must be able to capture and reflect the changes in the segments in the workpiece. A sample voxelized geometry is shown in Figure 5-2 (d). The grid size for this sample is $40 \times 40 \times 100$.

In order to load the training data efficiently during the training process, the voxelized geometries are saved as NPY files.

Now that we have the training data, we need to design and implement a deep learning model.

5.3 Convolutional Neural Network Design

The next step of the process is to design a deep learning model. In this project, a 3D CNN is used as the deep learning algorithm.

In general, a neural network has many features that must be tuned before using it on a test dataset. In the following section, all the possible parameters are selected by hyperparameter tuning.

5.3.1 K-Fold Cross-Validation

In order to select different parameters of the neural network to achieve the fastest and most accurate results, several hyperparameter tuning processes are carried out. Although a few random searches and selections are made, most hyperparameters are selected through the cross-validation method.

The K-fold cross-validation method [102] is based on statistics. It is used to estimate the performance of a deep learning model in different scenarios with a lower bias. This procedure has one parameter, K, which shows the number of subsets of the main dataset and the minimum number of repetitions of the process.

For this process, the dataset must be randomly shuffled and then split into K subsets. One of these subsets is selected as the test data, which will be changed at each iteration, and the rest are used as the training data. Then the hyperparameter is selected from the L available options, and the neural network is set to that hyperparameter before being trained using the training data. Once the training is finished, the trained model is used on the test data to predict the natural frequencies of the given geometries, and a score is assigned to its performance. This process is repeated K times, and a new subset is selected as the test dataset at each iteration.

After K iterations, the hyperparameter is changed to the next available option, and the training process is repeated K times with K random subsets. A score is assigned to the performance of the trained network after each iteration. This procedure must be done for all the L options for the hyperparameter. In the end, each option's performance scores are compared, and the option with the minimum standard deviation and the best performance

score is selected for that hyperparameter. The flowchart of this process is shown in Figure 5-3.

This study uses the average error for all the test set samples' natural frequencies as the performance score. A function is developed to calculate this score automatically. This function receives the test data, the trained model, and the inverse scaler as inputs. Then it predicts the natural frequencies of the unseen geometries and unnormalizes them using the inverse scaler. These unnormalized results are used to calculate the error of prediction for each natural frequency of each geometry. The output of this function is the mean value of the errors.

This process is performed for several hyperparameter tunings, such as selecting the depth of the network, number of filters, loss function, and optimizer.

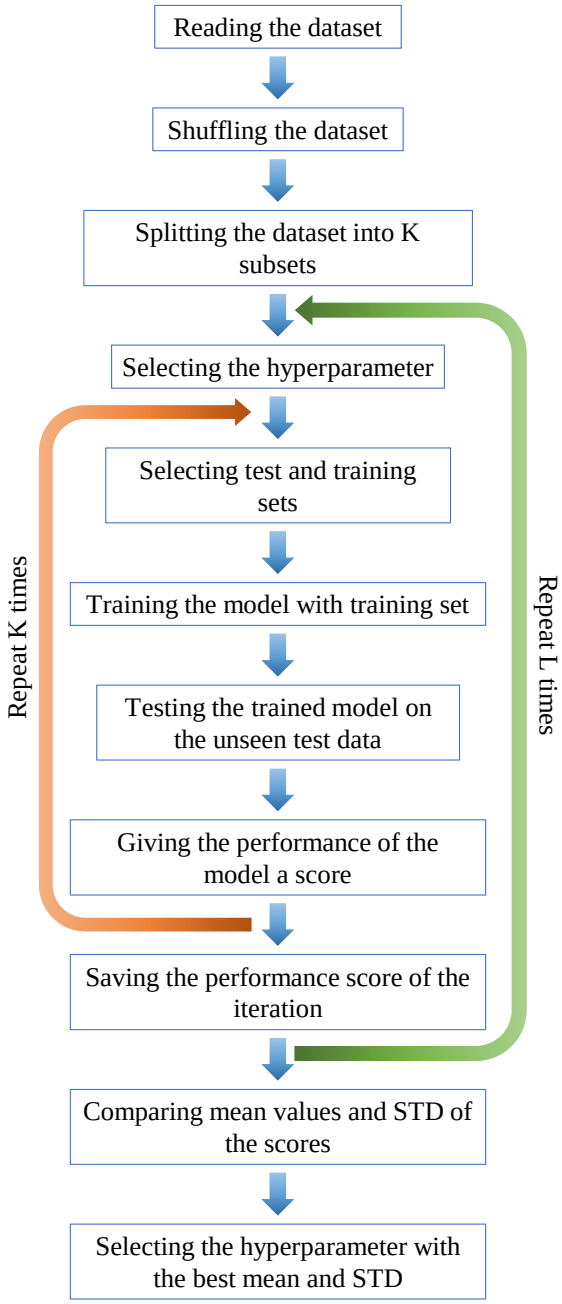


Figure 5-3 K-fold cross-validation flow chart

5.3.1.1 Depth of Network

The depth of the neural network is one of the hyperparameters that can affect the model's performance. In this project, five different convolutional neural networks with different depths are developed. The first network has three blocks of convolutional layers, and the last has seven blocks. These networks are then evaluated and scored by a 4-fold cross-validation process. The mean values of overall prediction error for each of these neural networks are shown in Figure 5-4.

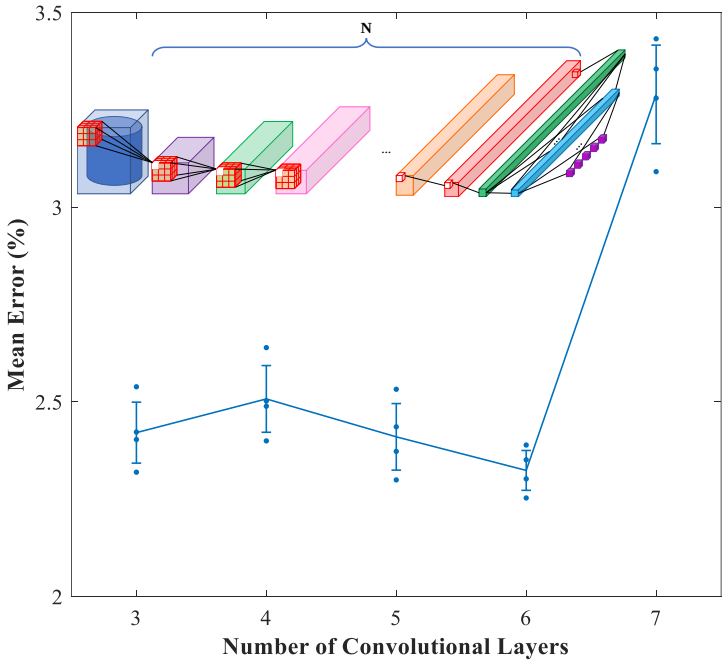


Figure 5-4 Mean error of 4-fold cross-validation for depths of the network

As can be seen in this figure, the model with six blocks of convolutional layers has the lowest error, and the performance stability is also better compared to other architectures.

5.3.1.2 Number of Filters

Similar to the depth of the network, the number of filters in each layer can also affect the model's performance. In order to select the best combination of the number of filters in the network, four different networks with six blocks of convolutional layer and different numbers of filters are developed. The number of filters for each developed model is shown in Table 5-1.

Table 5-1- Number of filters in four different neural networks

	Filter Combinations
Option 1	(4, 8, 12, 24, 36, 48)
Option 2	(8, 16, 32, 64, 128, 256)
Option 3	(16, 32, 64, 128, 256, 512)
Option 4	(32, 64, 128, 256, 512, 1024)

These networks are then tested in a 4-fold cross-validation process, and their mean error value is obtained. These results are shown in Figure 5-5.

As seen in this figure, the mean value for error increases with the number of filters. However, the stability shows minimum changes with the number of filters. As a result, the model with the lowest number of filters is selected as the final architecture of the neural network. This architecture is shown in Figure 5-6. The first stage includes six convolutional blocks, each consisting of a 3D convolutional layer followed by a max-pooling layer. The size of the filters is $3 \times 3 \times 3$ for all the layers, and the moving stride during convolution operations for each filter is fixed. When these filters move along the input array's depth,

width, and height, they generate scaler numbers as the output of each operation. In order to add nonlinearity to the network, ReLU (Rectified Linear Unit) activation is used on filter outputs.

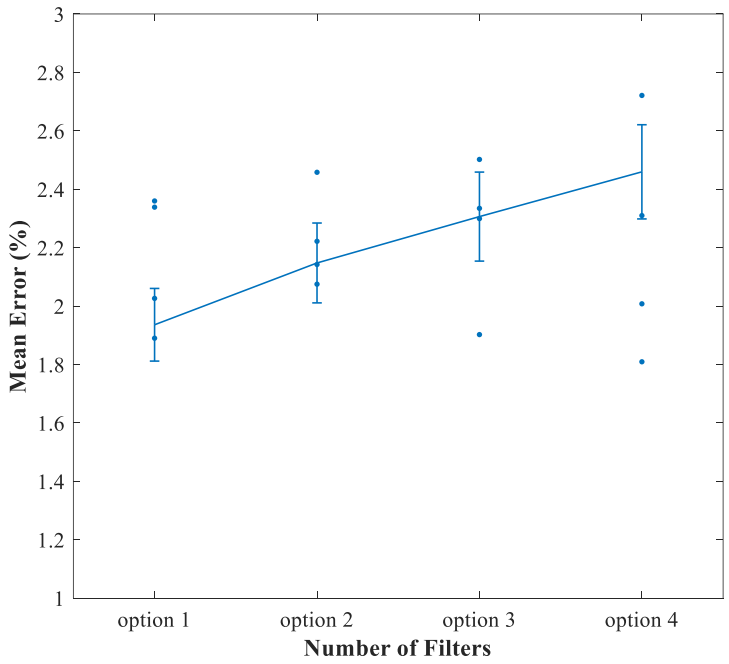


Figure 5-5 Mean error of 4-fold cross-validation for different numbers of filters

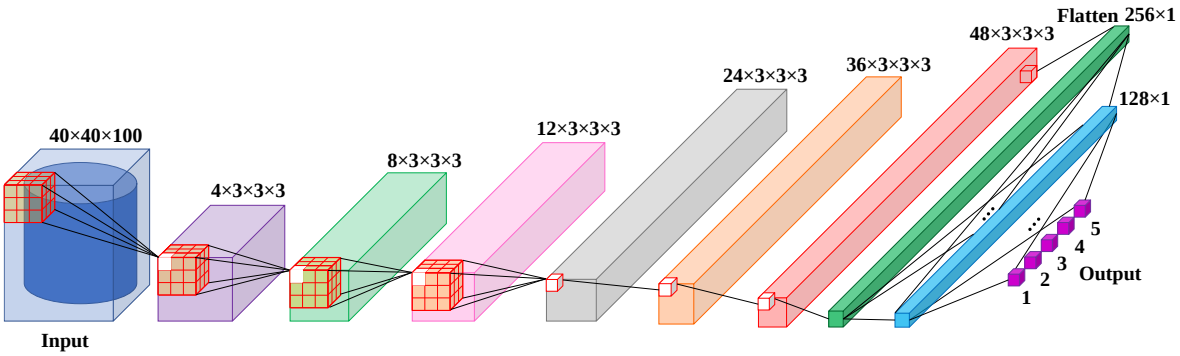


Figure 5-6 Architecture of the 3D convolutional neural network

The input array for the neural network is a 5D tensor with the size of (batch size, input width, input height, input depth, 1). The filters of the convolutional layers are being convolved with the input tensor with the stride of $(1 \times 1 \times 1)$. The number of filters for the six convolutional layers is 4, 8, 12, 24, 36, and 48, respectively. The max-pooling layer, which comes after each convolutional layer, has a kernel size of $2 \times 2 \times 2$ and is used to reduce the spatial size of the output by down sampling. Using this layer, each $2 \times 2 \times 2$ subregion is substituted by its maximum value.

In the flattening stage, the output of the last convolutional layer is flattened, and three fully connected layers are used. These layers alleviate the dimensions of the 3D feature map and extract the non-linear relation between high-level features and the outputs. The activation function for these layers is the same as convolutional layers. The final fully connected layer has five neurons, giving the input geometries' first five natural frequencies.

Now that the architecture of the network is selected, we need to find other parameters of the deep learning model. Considering each of them as a hyperparameter, the best option must be selected through cross-validation.

5.3.1.3 Loss Function

Mathematically, the primary purpose is to find a 3D convolutional neural network regression model that minimizes the loss function. The possible loss functions for a regression problem that are used in 4-fold cross-validation are Mean Squared Logarithmic Error (msle), Mean Squared Error (mse), Mean Absolute Error (mae), Huber Loss, and log-cosh. The mean error and standard deviation of the 4-fold cross-validation of these loss functions are shown in Figure 5-7. It can be seen in this plot that the Mean Squared

Logarithmic Error method returns a better result with acceptable stability. The formulation of this loss function is

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2, \tag{5.2}$$

where n is the number of outputs, and y_i and $\hat{y}_i$ are the ground truth and the predicted values for the outputs, respectively.

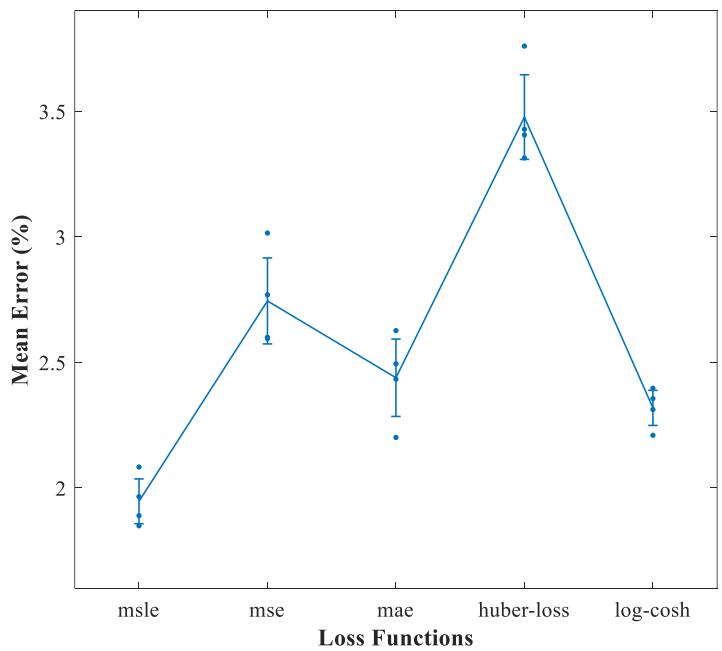


Figure 5-7 Mean error of 4-fold cross-validation for different loss functions

5.3.1.4 Optimizer

Optimizer is one of the arguments needed to compile a neural network model. The optimizers evaluated in a 4-fold cross-validation process are RMSprop, Adam, Adamax, and Nadam. Based on the test results of the cross-validation, shown in Figure 5-8, Nadam gives more accurate results with better stability.

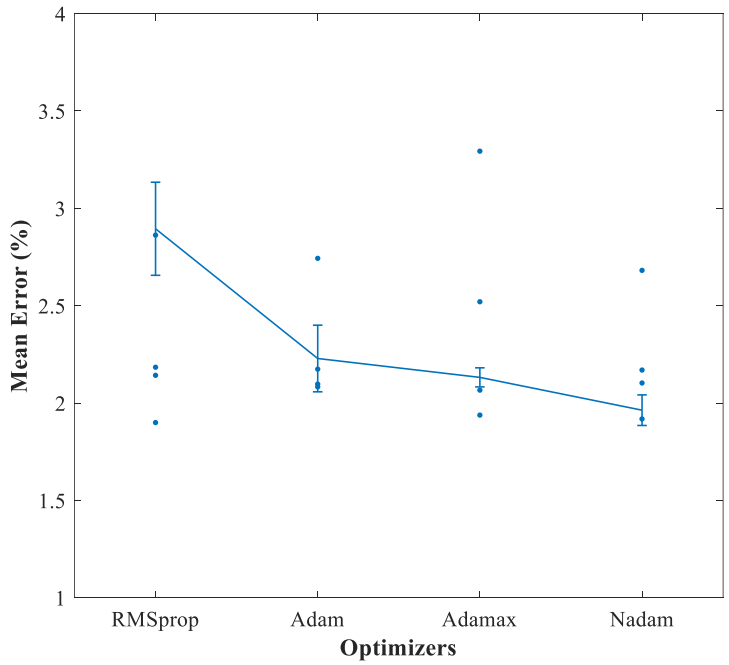


Figure 5-8 Mean error of 4-fold cross-validation for different optimizers

5.4 Training Process

In the learning process, the weights and biases of the model are trained such that the network predicts the outputs as close to the ground truth, which are natural frequencies obtained by the FE method, as possible. The designed neural network is implemented using the Keras library in Python. Out of 1024 generated geometries, 100 samples are used for validation, 100 samples are used for the test phase, and the rest are used for training the neural network with a batch size of 10. Figure 5-9 shows the loss as a training iteration function for training and validation data. As can be seen in this figure, after 200 epochs, there is no considerable improvement in validation accuracy. Hence, the model at epoch number 200 is used as the trained model for test data evaluation on unseen data.

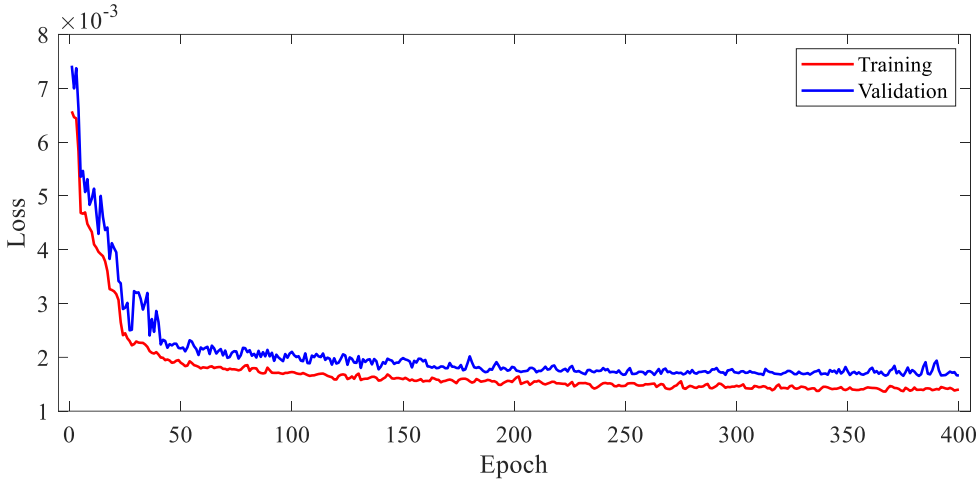


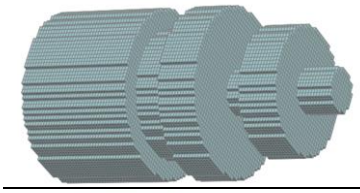
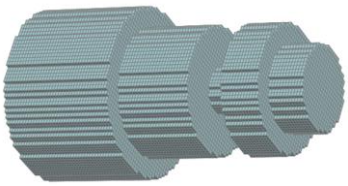
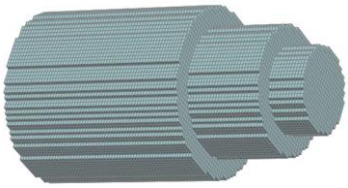
Figure 5-9 Training and validation loss for each training epoch

5.5 Unseen Data Test Results

After training the neural network, the selected trained model is used to predict the natural frequencies of unseen geometries. These geometries were excluded from the training dataset and are new to the trained model. The test dataset is fed to the trained model as the input. The model's output is compared to the actual natural frequencies of the geometries to evaluate the model's performance.

Table 5-2 shows examples of the test geometries and their natural frequencies acquired by the trained 3D-CNN model and FE method. The FE method uses the exact geometries, while the 3D-CNN obtains the natural frequencies based on the voxelized geometries.

Table 5-2 Examples of test data natural frequencies obtained from FE and 3D-CNN algorithm

Input Geometries	Method	$\omega_1(Hz)$	$\omega_2(Hz)$	$\omega_3(Hz)$	$\omega_4(Hz)$	$\omega_5(Hz)$
	FEM	299	775	817	2469	2529
	3D-CNN	296	784	816	2492	2458
	Error	0.8%	1.2%	0.1%	1%	2.8%
	FEM	232	748	837	2302	2639
	3D-CNN	238	745	839	2281	2644
	Error	1.3%	0.4%	0.3%	0.9%	0.2%
	FEM	393	1691	2114	3130	3671
	3D-CNN	392	1679	2139	3145	3700
	Error	0.2%	0.7%	1.2%	0.5%	0.8%

For the geometries shown in Table 5-2, the prediction of natural frequencies by the FE model takes over 15 s. The 3D-CNN model, on the other hand, generates the prediction results in about 0.05 s, making it a viable method for online monitoring and control applications.

5.6 Conclusion

In this chapter, a deep learning based framework, which is able to predict the natural frequencies of an in-process workpiece, is developed. In the first step, a training dataset is generated using the developed FE model and geometry voxelization method. Then, a neural network is designed using K-fold cross-validation. This neural network is trained on the training dataset. The trained model is then used on the test dataset. It is shown that the

developed framework can predict the natural frequencies of unseen geometries with reasonable accuracy in milliseconds.

6 Conclusion and Future Research Directions

6.1 Summary and Conclusion

The stability of a turning process depends on the structural dynamics of the part that is clamped on a lathe. The varying geometry of the part as the material is removed leads to changes in the structural dynamics of the workpiece. To predict and avoid chatter vibration, it is crucial to predict the dynamic characteristics of the part during operation. In this thesis, a solution is implemented to rapidly predict the structural dynamics of cylindrical geometries during the turning process. The developed algorithms can be summarized as:

- In order to reduce the number of simulations and computational time in the process planning stage, an efficient physics-based model for process simulation is developed in this study. Using this model, the characteristics of the workpiece can be identified efficiently before the actual process takes place. This model is based on the finite element method and uses dynamics substructuring approach to foresee the dynamics of the in-process workpiece without generating several FE models for different machining steps or remeshing the part after each material removal. The mesh grid for the initial part is generated only once before machining with consideration of the depth of cut at each machining step. The FRF of the initial part is constructed in the frequency domain based on the generated mesh grid.

- In the substructuring approach, a fictitious substructure with opposite dynamics is virtually added to the workpiece, and the part's dynamics are analytically updated accordingly without changing the geometry. This process is carried out by decoupling the removed material's dynamics from the initial part's FRF at discrete machining locations. By comparing and validating the results of the developed algorithm with ABAQUS results, it was observed that the developed model could update the FRF of the in-process workpiece with good accuracy.
- To decrease the number and the size of the matrix inversions in FRF calculations, the FRF of the initial part is calculated using mode superposition. In this method, only a limited number of modes are used to calculate the FRF of the workpiece. To overcome the modal incompleteness, the FRF of the initial part is recompensed for the contribution of the truncated higher-order modes by adding their effect to the low-frequency modes.
- An order reduction procedure is carried out to reduce the computation time and make the calculation process more efficient. First, the DOFs with the most contribution to the model's dynamics are selected using the iterative effective independence method. Then, the effect of the rest of DOFs is reflected on master DOFs using a combination of three different reduction methods. It is shown that using the FRF updating algorithm and reduction method, the implemented solution can update the dynamics of the in-process workpiece much faster than the standard methods. It is observed that the

reduction in computational time is almost proportional to the reduction of computational DOFs.

- To reduce the computational time for real-time applications, a deep learning based method for predicting the dynamics of the part is proposed. This method generates the training input and output data for the CNN using the developed FE model. The created mesh grids of a set of geometries are converted to digital format to be used as training inputs, and each geometry's natural frequencies are used as outputs. The parameters of the CNN are selected using hyperparameter tuning with cross-validation, and the neural network is trained by the synthetically generated dataset. It is shown that the developed framework can predict the natural frequencies of an unseen part with a similar geometry within milliseconds, making it a suitable method for online process planning.

6.2 Future Research Directions

The study presented in this thesis can be improved and continued in multiple ways.

- Since tubes and pipes are widely used in the industry, the developed algorithm could be extended to cylindrical thin-walled structures.
- Including modal damping in the model can increase the model's accuracy and, therefore, should be investigated.
- After adding modal damping to the model, a training dataset can be generated using the predicted FRFs as the outputs.

- A more sophisticated deep learning based framework can be developed and trained on the new dataset to predict the FRF of the in-process workpieces.

Bibliography

[1] G. Birkhoff, "Piecewise bicubic interpolation and approximation in polygons," in *Approximation with Special Emphasis on Spline Functions V*, I. Schoenberg, Ed. New York: Academic Press, 1969, pp. 185–221.

[2] H. C. Martin and G. F. Carey, *Introduction to finite element analysis: theory and application*. New York: McGraw-Hill College, 1973.

[3] G. Strang and G. J. Fix, *An analysis of the finite element method*. New Jersey: Englewood Cliffs, N. J., Prentice-Hall, Inc., 1973.

[4] C. Eksioglu, "Mechanics and Dynamics of Thin-Wall Machining," MASc. Thesis, The University of British Columbia, Vancouver, 2011.

[5] F. Ismail and R. Ziaei, "Chatter suppression in five-axis machining of flexible parts," *Int J Mach Tools Manuf*, vol. 42, no. 1, pp. 115–122, Jan. 2002, doi: 10.1016/S0890-6955(01)00088-8.

[6] D. Feng, J. Wang, G. T.-C. Chiu, and A. Raman, "Vibrations of Air-Coupled Web Systems," *J Vib Acoust*, vol. 143, no. 1, Feb. 2021, doi: 10.1115/1.4047702.

[7] A. Raman, K.-D. Wolf, and P. Hagedorn, "Observations on the vibrations of paper webs," in *Proceedings of the International Conference on Web Handling*, Jun. 2001, pp. 415–439.

[8] H. Koivurova and A. Pramila, "Nonlinear vibration of axially moving membrane by finite element method," *Comput Mech*, vol. 20, no. 6, pp. 573–581, Nov. 1997, doi: 10.1007/s004660050277.

[9] G. Stepan, A. K. Kiss, B. Ghalamchi, J. Sopanen, and D. Bachrathy, "Chatter avoidance in cutting highly flexible workpieces," *CIRP Ann Manuf Technol*, vol. 66, no. 1, pp. 377–380, 2017, doi: 10.1016/j.cirp.2017.04.054.

[10] G. Urbikain, L. N. López De Lacalle, F. J. Campa, A. Fernández, and A. Elías, "Stability prediction in straight turning of a flexible workpiece by collocation method," *Int J Mach Tools Manuf*, vol. 54–55, pp. 73–81, Mar. 2012, doi: 10.1016/j.ijmachtools.2011.11.008.

[11] A. Fischer, P. Eberhard, and J. Ambrósio, "Parametric flexible multibody model for material removal during turning," *J Comput Nonlinear Dyn*, vol. 9, no. 1, 2014, doi: 10.1115/1.4025283.

[12] A. Gerasimenko, M. Guskov, J. Duchemin, P. Lorong, and A. Gouskov, "Variable compliance-related aspects of chatter in turning thin-walled tubular parts," in *Procedia CIRP*, 2015, vol. 31, pp. 58–63. doi: 10.1016/j.procir.2015.03.088.

[13] P. Lorong, A. Larue, and A. Perez Duarte, "Dynamic study of thin wall part turning," in *Advanced Materials Research*, 2011, vol. 223, pp. 591–599. doi: 10.4028/www.scientific.net/AMR.223.591.

[14] K. Mehdi, J. F. Rigal, and D. Play, "Dynamic behavior of a thin-walled cylindrical workpiece during the turning process, Part 1: Cutting process simulation," *J Manuf Sci Eng*, vol. 124, no. 3, pp. 562–568, 2002, doi: 10.1115/1.1431260.

[15] M. Wan, W. Zhang, K. Qiu, T. Gao, and Y. Yang, "Numerical prediction of static form errors in peripheral milling of thin-walled workpieces with irregular meshes," *J Manuf Sci Eng*, vol. 127, no. 1, pp. 13–22, 2005, doi: 10.1115/1.1828055.

[16] E. Budak and Y. Altintas, "Modeling and avoidance of static form errors in peripheral milling of plates," *Int J Mach Tools Manuf*, vol. 35, no. 3, pp. 459–476, Mar. 1995, doi: 10.1016/0890-6955(94)P2628-S.

[17] M. A. Elbestawi and R. Sagherian, "Dynamic modeling for the prediction of surface errors in the milling of thin-walled sections," *J Mater Process Technol*, vol. 25, no. 2, pp. 215–228, Mar. 1991, doi: 10.1016/0924-0136(91)90090-2.

[18] J.-S. Tsai and C.-L. Liao, "Finite-element modeling of static surface errors in the peripheral milling of thin-walled workpieces," *J Mater Process Technol*, vol. 94, no. 2–3, pp. 235–246, Sep. 1999, doi: 10.1016/S0924-0136(99)00109-0.

[19] W. A. Kline, R. E. DeVor, and I. A. Shareef, "The Prediction of Surface Accuracy in End Milling," *Journal of Engineering for Industry*, vol. 104, no. 3, pp. 272–278, Aug. 1982, doi: 10.1115/1.3185830.

[20] M.-P. Tsai, N.-C. Tsai, and C.-W. Yeh, "On milling of thin-wall conical and tubular workpieces," *Mech Syst Signal Process*, vol. 72–73, pp. 395–408, May 2016, doi: 10.1016/j.ymssp.2015.10.009.

[21] Y. Koike, A. Matsubara, and I. Yamaji, "Design method of material removal process for minimizing workpiece displacement at cutting point," *CIRP Ann Manuf Technol*, vol. 62, no. 1, pp. 419–422, 2013, doi: 10.1016/j.cirp.2013.03.144.

[22] L. Arnaud, O. Gonzalo, S. Seguy, H. Jauregi, and G. Peigné, "Simulation of low rigidity part machining applied to thin-walled structures," *International Journal of Advanced Manufacturing Technology*, vol. 54, no. 5–8, pp. 479–488, May 2011, doi: 10.1007/s00170-010-2976-9.

[23] U. Bravo, O. Altuzarra, L. N. López De Lacalle, J. A. Sánchez, and F. J. Campa, "Stability limits of milling considering the flexibility of the workpiece and the machine," *Int J Mach Tools Manuf*, vol. 45, no. 15, pp. 1669–1680, Dec. 2005, doi: 10.1016/j.ijmachtools.2005.03.004.

[24] Y. Ding and L. Zhu, "Investigation on chatter stability of thin-walled parts considering its flexibility based on finite element analysis," *International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9–12, pp. 3173–3187, Feb. 2018, doi: 10.1007/s00170-016-9471-x.

[25] Y. Liu, B. Wu, J. Ma, and D. Zhang, "Chatter identification of the milling process considering dynamics of the thin-walled workpiece," *International Journal of Advanced Manufacturing Technology*, vol. 89, no. 5–8, pp. 1765–1773, Mar. 2017, doi: 10.1007/s00170-016-9190-3.

[26] R. B. Mundim and A. V. Borille, "An approach for reducing undesired vibrations in milling of low rigidity structures," *International Journal of Advanced Manufacturing Technology*, vol. 88, no. 1–4, pp. 971–983, Jan. 2017, doi: 10.1007/s00170-016-8804-0.

[27] S. Seguy, G. Dessein, and L. Arnaud, "Surface roughness variation of thin wall milling, related to modal interactions," *International Journal of Machine Tools and*

Manufacture, vol. 48, no. 3–4. pp. 261–274, Mar. 2008. doi: 10.1016/j.ijmachtools.2007.09.005.

[28] V. Thevenot, L. Arnaud, G. Dessein, and G. Cazenave-Larroche, "Influence of material removal on the dynamic behavior of thin-walled structures in peripheral milling," *Machining Science and Technology*, vol. 10, no. 3, pp. 275–287, Sep. 2006, doi: 10.1080/10910340600902082.

[29] I. Mañé, V. Gagnol, B. C. Bouzgarrou, and P. Ray, "Stability-based spindle speed control during flexible workpiece high-speed milling," *Int J Mach Tools Manuf*, vol. 48, no. 2, pp. 184–194, Feb. 2008, doi: 10.1016/j.ijmachtools.2007.08.018.

[30] Q. Song, X. Ai, and W. Tang, "Prediction of simultaneous dynamic stability limit of time-variable parameters system in thin-walled workpiece high-speed milling processes," *International Journal of Advanced Manufacturing Technology*, vol. 55, no. 9–12, pp. 883–889, Aug. 2011, doi: 10.1007/s00170-010-3139-8.

[31] F. J. Campa, L. N. Lopez De Lacalle, and A. Celaya, "Chatter avoidance in the milling of thin floors with bull-nose end mills: Model and stability diagrams," *Int J Mach Tools Manuf*, vol. 51, no. 1, pp. 43–53, Jan. 2011, doi: 10.1016/j.ijmachtools.2010.09.008.

[32] O. B. Adetoro, W. M. Sim, and P. H. Wen, "An improved prediction of stability lobes using nonlinear thin wall dynamics," *J Mater Process Technol*, vol. 210, no. 6–7, pp. 969–979, Apr. 2010, doi: 10.1016/j.jmatprotec.2010.02.009.

[33] P. Kersting and D. Biermann, "Modeling techniques for simulating workpiece deflections in NC milling," *CIRP J Manuf Sci Technol*, vol. 7, no. 1, pp. 48–54, 2014, doi: 10.1016/j.cirpj.2013.08.002.

[34] M. Meshreki, H. Attia, and J. Kövecses, "Development of a new model for the varying dynamics of flexible pocket-structures during machining," *J Manuf Sci Eng*, vol. 133, no. 4, 2011, doi: 10.1115/1.4004322.

[35] K. Ahmadi, "Finite strip modeling of the varying dynamics of thin-walled pocket structures during machining," *International Journal of Advanced Manufacturing Technology*, vol. 89, no. 9–12, pp. 2691–2699, Apr. 2017, doi: 10.1007/s00170-016-8931-7.

[36] Q. Song, Z. Liu, Y. Wan, G. Ju, and J. Shi, "Application of Sherman-Morrison-Woodbury formulas in instantaneous dynamic of peripheral milling for thin-walled component," *Int J Mech Sci*, vol. 96–97, pp. 79–90, Jun. 2015, doi: 10.1016/j.ijmecsci.2015.03.021.

[37] E. Budak, L. T. Tunç, S. Alan, and H. N. Özgüven, "Prediction of workpiece dynamics and its effects on chatter stability in milling," *CIRP Ann Manuf Technol*, vol. 61, no. 1, pp. 339–342, 2012, doi: 10.1016/j.cirp.2012.03.144.

[38] Y. Yang, W. H. Zhang, Y. C. Ma, and M. Wan, "Chatter prediction for the peripheral milling of thin-walled workpieces with curved surfaces," *Int J Mach Tools Manuf*, vol. 109, pp. 36–48, Oct. 2016, doi: 10.1016/j.ijmachtools.2016.07.002.

[39] M. R. Khoshdarregi and Y. Altintas, "Dynamics of Multipoint Thread Turning—Part I: General Formulation," *J Manuf Sci Eng*, vol. 140, no. 6, Jun. 2018, doi: 10.1115/1.4038570.

[40] M. R. Khoshdarregi and Y. Altintas, "Dynamics of Multipoint Thread Turning—Part II: Application to Thin-Walled Oil Pipes," *J Manuf Sci Eng*, vol. 140, no. 4, Apr. 2018, doi: 10.1115/1.4038573.

[41] E. Budak and E. Ozturk, "Dynamics and stability of parallel turning operations," *CIRP Annals*, vol. 60, no. 1, pp. 383–386, 2011, doi: 10.1016/j.cirp.2011.03.028.

[42] E. Ozturk, A. Comak, and E. Budak, "Tuning of tool dynamics for increased stability of parallel (simultaneous) turning processes," *J Sound Vib*, vol. 360, pp. 17–30, Jan. 2016, doi: 10.1016/j.jsv.2015.09.009.

[43] M. J. Reith, D. Bachrathy, and G. Stepan, "Optimal Detuning of a Parallel Turning System—Theory and Experiments," *J Dyn Syst Meas Control*, vol. 139, no. 1, Jan. 2017, doi: 10.1115/1.4034497.

[44] C. Brecher, A. Epple, S. Neus, and M. Fey, "Optimal process parameters for parallel turning operations on shared cutting surfaces," *Int J Mach Tools Manuf*, vol. 95, pp. 13–19, Aug. 2015, doi: 10.1016/j.ijmachtools.2015.05.003.

[45] M. Azvar and E. Budak, "Multi-dimensional chatter stability for enhanced productivity in different parallel turning strategies," *Int J Mach Tools Manuf*, vol. 123, pp. 116–128, Dec. 2017, doi: 10.1016/j.ijmachtools.2017.08.005.

[46] O. Tuysuz and Y. Altintas, "Frequency Domain Updating of Thin-Walled Workpiece Dynamics Using Reduced Order Substructuring Method in Machining," *J Manuf Sci Eng*, vol. 139, no. 7, Jul. 2017, doi: 10.1115/1.4036124.

[47] O. Tuysuz and Y. Altintas, "Time-Domain Modeling of Varying Dynamic Characteristics in Thin-Wall Machining Using Perturbation and Reduced-Order Substructuring Methods," *J Manuf Sci Eng*, vol. 140, no. 1, Jan. 2018, doi: 10.1115/1.4038000.

[48] P. Franciosa, M. Sokolov, S. Sinha, T. Sun, and D. Ceglarek, "Deep learning enhanced digital twin for Closed-Loop In-Process quality improvement," *CIRP Annals*, vol. 69, no. 1, pp. 369–372, 2020, doi: 10.1016/j.cirp.2020.04.110.

[49] M. Hassan, A. Sadek, and M. H. Attia, "A Generalized Multisensor Real-Time Tool Condition–Monitoring Approach Using Deep Recurrent Neural Network," *Smart Sustain Manuf Syst*, vol. 3, no. 2, p. 20190020, Feb. 2019, doi: 10.1520/SSMS20190020.

[50] X. Lin, X. Wang, and L. Li, "Intelligent detection of edge inconsistency for mechanical workpiece by machine vision with deep learning and variable geometry model," *Applied Intelligence*, vol. 50, no. 7, pp. 2105–2119, Jul. 2020, doi: 10.1007/s10489-020-01641-3.

[51] Z. Zhang, P. Jaiswal, and R. Rai, "FeatureNet: Machining feature recognition based on 3D Convolution Neural Network," *Computer-Aided Design*, vol. 101, pp. 12–22, Aug. 2018, doi: 10.1016/j.cad.2018.03.006.

[52] B. Denkena, B. Bergmann, and S. Reimer, "Analysis of different machine learning algorithms to learn stability lobe diagrams," *Procedia CIRP*, vol. 88, pp. 282–287, 2020, doi: 10.1016/j.procir.2020.05.049.

[53] M. Postel, B. Bugdayci, and K. Wegener, "Ensemble transfer learning for refining stability predictions in milling using experimental stability states," *The International Journal of Advanced Manufacturing Technology*, vol. 107, no. 9–10, pp. 4123–4139, Apr. 2020, doi: 10.1007/s00170-020-05322-w.

[54] M. C. Yesilli, F. A. Khasawneh, and A. Otto, "On transfer learning for chatter detection in turning using wavelet packet transform and ensemble empirical mode decomposition," *CIRP J Manuf Sci Technol*, vol. 28, pp. 118–135, Jan. 2020, doi: 10.1016/j.cirpj.2019.11.003.

[55] M. C. Yesilli, S. Tymochko, F. A. Khasawneh, and E. Munch, "Chatter Diagnosis in Milling Using Supervised Learning and Topological Features Vector," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Dec. 2019, pp. 1211–1218. doi: 10.1109/ICMLA.2019.00200.

[56] Y. Chen, H. Li, X. Jing, L. Hou, and X. Bu, "Intelligent chatter detection using image features and support vector machine," *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 5–8, pp. 1433–1442, Jun. 2019, doi: 10.1007/s00170-018-3190-4.

[57] K. Zacharia and P. Krishnakumar, "Chatter Prediction in High Speed Machining of Titanium Alloy (Ti-6Al-4V) using Machine Learning Techniques," *Mater Today Proc*, vol. 24, pp. 350–358, 2020, doi: 10.1016/j.matpr.2020.04.286.

[58] I. Kvinevskiy, S. Bedi, and S. Mann, "Detecting machine chatter using audio data and machine learning," *The International Journal of Advanced Manufacturing Technology*, vol. 108, no. 11–12, pp. 3707–3716, Jun. 2020, doi: 10.1007/s00170-020-05571-9.

[59] F. Shi, H. Cao, X. Zhang, and X. Chen, "A Reinforced k-Nearest Neighbors Method With Application to Chatter Identification in High-Speed Milling," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10844–10855, Dec. 2020, doi: 10.1109/TIE.2019.2962465.

[60] M.-Q. Tran, M.-K. Liu, and Q.-V. Tran, "Milling chatter detection using scalogram and deep convolutional neural network," *The International Journal of Advanced Manufacturing Technology*, vol. 107, no. 3–4, pp. 1505–1516, Mar. 2020, doi: 10.1007/s00170-019-04807-7.

[61] K. Li, S. He, B. Li, H. Liu, X. Mao, and C. Shi, "A novel online chatter detection method in milling process based on multiscale entropy and gradient tree boosting," *Mech Syst Signal Process*, vol. 135, p. 106385, Jan. 2020, doi: 10.1016/j.ymssp.2019.106385.

[62] S. Vaishnav, A. Agarwal, and K. A. Desai, "Machine learning-based instantaneous cutting force model for end milling operation," *J Intell Manuf*, vol. 31, no. 6, pp. 1353–1366, Aug. 2020, doi: 10.1007/s10845-019-01514-8.

[63] T. Zhang, M. Xiao, Y. Zou, and J. Xiao, "Robotic Constant-Force Grinding Control with A Press-And-Release Model And Model-Based Reinforcement Learning," *The*

International Journal of Advanced Manufacturing Technology, vol. 106, no. 1–2, pp. 589–602, Jan. 2020, doi: 10.1007/s00170-019-04614-0.

[64] V. Samsonov, C. Enslin, H.-G. Köpken, S. Baer, and D. Lütticke, "Using Reinforcement Learning for Optimization of a Workpiece Clamping Position in a Machine Tool," in *Proceedings of the 22nd International Conference on Enterprise Information Systems*, 2020, pp. 506–514. doi: 10.5220/0009354105060514.

[65] L. J. Segerlind, *Applied Finite Element Analysis*, Second Edition. John Wiley & Sons Inc., 1984.

[66] A. F. Bower, *Applied Mechanics of Solids*, First Edition. CRC Press, 2009. doi: 10.1201/9781439802489.

[67] M. H. Sadd, "Elasticity: Theory, Applications, and Numerics," Third edition., Waltham: Academic Press, 2014, pp. 81–95.

[68] A. J. M. Ferreira, *MATLAB Codes for Finite Element Analysis*. Dordrecht: Springer Netherlands, 2009. doi: 10.1007/978-1-4020-9200-8.

[69] W. D'Ambrogio and A. Fregolent, "The role of interface DoFs in decoupling of substructures based on the dual domain decomposition," *Mech Syst Signal Process*, vol. 24, no. 7, pp. 2035–2048, Oct. 2010, doi: 10.1016/j.ymssp.2010.05.007.

[70] S. N. Voormeeren and D. J. Rixen, "A family of substructure decoupling techniques based on a dual assembly approach," *Mech Syst Signal Process*, vol. 27, no. 1, pp. 379–396, Feb. 2012, doi: 10.1016/j.ymssp.2011.07.028.

[71] R. R. Craig and A. J. Kurdila, *Fundamentals of Structural Dynamics*, Second Edition. Hoboken: John Wiley & Sons Inc., 2006.

[72] P. L. C. V. D. Valk, "Model Reduction and Interface Modeling in Dynamic Substructuring: Application to a Multi-Megawatt Wind Turbine," M.Sc. Thesis, Delft University of Technology, Delft, The Netherlands, 2010.

[73] S. N. Voormeeren, "Coupling Procedure Improvement and Uncertainty Quantiflcation in Experimental Dynamic Substructuring," M.Sc., Delft University of Technology, 2007.

[74] W. D'Ambrogio and A. Fregolent, "The role of interface DoFs in decoupling of substructures based on the dual domain decomposition," *Mech Syst Signal Process*, vol. 24, no. 7, pp. 2035–2048, Oct. 2010, doi: 10.1016/j.ymssp.2010.05.007.

[75] J. Ji, "Gauss-Jordan elimination methods for the Moore-Penrose inverse of a matrix," *Linear Algebra Appl*, vol. 437, no. 7, pp. 1835–1844, Oct. 2012, doi: 10.1016/j.laa.2012.05.017.

[76] P. S. Stanimirović and M. D. Petković, "Gauss-Jordan elimination method for computing outer inverses," *Appl Math Comput*, vol. 219, no. 9, pp. 4667–4679, Jan. 2013, doi: 10.1016/j.amc.2012.10.081.

[77] "ABAQUS." Dassault Systemes, 2022.

[78] J. R. Davis, Ed., *ASM Specialty Handbook - Carbon and Alloy Steels*. Metals Park, OH, United States: ASM International, 1996.

[79] M. Bauccio, Ed., *ASM Metals Reference Book*, Third Edition. Materials Park, OH, United States: ASM International, 1993.

[80] M. Law, A. S. Phani, and Y. Altintas, "Position-dependent multibody dynamic modeling of machine tools based on improved reduced order models," *J Manuf Sci Eng*, vol. 135, no. 2, 2013, doi: 10.1115/1.4023453.

[81] D. S. Li, H. N. Li, and C. P. Fritzen, "The connection between effective independence and modal kinetic energy methods for sensor placement," *J Sound Vib*, vol. 305, no. 4–5, pp. 945–955, Sep. 2007, doi: 10.1016/j.jsv.2007.05.004.

[82] D. C. Kammer, "Sensor placement for on-orbit modal identification and correlation of large space structures," in *Proceedings of the American Control Conference*, 1990, pp. 2984–2990. doi: 10.23919/acc.1990.4791265.

[83] T. K. Yu and J. H. Seinfeld, "Observability and optimal measurement location in linear distributed parameter systems†," *Int J Control*, vol. 18, no. 4, pp. 785–799, 1973, doi: 10.1080/00207177308932556.

[84] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Application*, Second Edition. New York: Springer-Verlag, 2002.

[85] J. H. Wilkinson, *The algebraic eigenvalue problem*. London, UK: Oxford University Press, 1965.

[86] J. O'Callahan and P. Avitabile, "System Equivalent Reduction Expansion Process (SEREP)," in *7th International Modal Analysis Conference (IMAC), Society of Experimental Mechanics*, 1989, pp. 29–37.

[87] C. V. S. Sastry, D. Roy Mahapatra, S. Gopalakrishnan, and T. S. Ramamurthy, "An iterative system equivalent reduction expansion process for extraction of high frequency response from reduced order finite element model," *Comput Methods Appl*

Mech Eng, vol. 192, no. 15, pp. 1821–1840, Apr. 2003, doi: 10.1016/S0045-7825(03)00204-4.

[88] G. H. Golub and C. F. van Loan, *Matrix Computations*, Fourth Edition. Baltimore, Maryland, United States of America: The John Hopkins University Press, 2013.

[89] V. C. Fragkoulis, I. A. Kougioumtzoglou, and A. A. Pantelous, "Linear Random Vibration of Structural Systems with Singular Matrices," *J Eng Mech*, vol. 142, no. 2, Feb. 2016, doi: 10.1061/(ASCE)EM.1943-7889.0001000.

[90] L. B. saint Martin, R. U. Mendes, and K. L. Cavalca, "Model reduction and dynamic matrices extraction from state-space representation applied to rotating machines," *Mech Mach Theory*, vol. 149, p. 103804, Jul. 2020, doi: 10.1016/j.mechmachtheory.2020.103804.

[91] John. O'Callahan, "A Procedure for an Improved Reduced System (IRS) Model," in *Proceedings of 7th IMAC*, Jan. 1989, pp. 17–21.

[92] M. I. Friswell, S. D. Garvey, and J. E. T. Penny, "Model reduction using dynamic and iterated IRS techniques," *J Sound Vib*, vol. 186, no. 2, pp. 311–323, Sep. 1995, doi: 10.1006/jsvi.1995.0451.

[93] R. J. Guyan, "Reduction of stiffness and mass matrices," *AIAA J*, vol. 3, no. 2, pp. 380–380, 1965, doi: 10.2514/3.2874ï.

[94] S. G. Braun and Y. M. Ram, "Modal modification of vibrating systems: some problems and their solutions," *Mech Syst Signal Process*, vol. 15, no. 1, pp. 101–119, 2001, doi: 10.1006/mssp.2000.1354.

[95] M. L. M. Duarte, "Experimentally-derived structural models for use in further dynamic analysis," Ph.D., Imperial College of Science, Technology and Medicine, London, United Kingdom, 1996.

[96] Liu Z. S., S. H. Chen, Liu W. T., and Y. Q. Zhao, "An accurate modal method for computing responses to harmful excitation," *Modal Analysis: the International Journal of Analytical and Experimental Modal Analysis*, vol. 9, no. 1, pp. 1–13, Jan. 1994.

[97] Y. Zhao, S. Chen, S. Chai, and Q. Qu, "An improved modal truncation method for responses to harmonic excitation," *Comput Struct*, vol. 80, no. 1, pp. 99–103, Jan. 2002, doi: 10.1016/S0045-7949(01)00148-1.

[98] A. Maghami, M. Salehi, and M. Khoshdarregi, "A 3D Deep Learning Model for Rapid Prediction of Structural Dynamics of Workpieces During Machining," *Procedia CIRP*, vol. 104, pp. 1753–1758, 2021, doi: 10.1016/j.procir.2021.11.295.

[99] "PythonOCC," *opencascade.org. https://dev.opencascade.org/project/pythonocc (accessed May 24, 2022)*.

[100] L. Huang, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Iterative Normalization: Beyond Standardization Towards Efficient Whitening," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4874–4883.

[101] "sklearn.preprocessing.MinMaxScaler," *https://scikit-learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html, (accessed Apr. 12, 2022)*.

[102] P. Refaeilzadeh, L. Tang, and H. Liu, “Cross-validation,” *Encyclopedia of database systems*, vol. 5, pp. 532–538, Jan. 2009.