Voxel-Based Modeling of Transient Material Removal in Machining

John C. Miers[1], Tommy Tucker[2], Thomas Kurfess[1], Christopher Saldana[1,*]
[1]George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA
[2]Tucker Innovations, Waxhaw, NC 28173 USA
[*]Corresponding author: christopher.saldana@me.gatech.edu

Abstract
In the present study, a novel framework is presented that models the transient interaction between cutting teeth of an arbitrary end mill geometry and a workpiece. In this framework, the workpiece geometry is modeled using a voxelized representation that is dynamically updated as material is locally removed by each tooth of the cutting tool. A ray casting approach was used to mimic the process of the cutting faces of the tool intersecting the workpiece material. This ray casting approach was used to calculate the instantaneous undeformed chip thickness. The resulting voxel based model framework was validated by comparison of predictions with experimentally measured milling forces. The effect of voxel size on model predictions was evaluated and the ability of the model to describe the interactive effects of sequential machining passes was also investigated. Implications of this voxel based model framework in terms of utility for predicting local surface finish and computational scalability of complex cutting configurations are briefly discussed.

Keywords: Machining, Simulation, Voxel, Ray casting

1 Introduction

Advances in discrete machining process models to support simulation, optimization and verification are critical toward maintaining progress toward fully model-driven process design. The need for discretized part models, such as voxel-based models, is further emphasized by a drive toward hybrid processing frameworks wherein additive and subtractive methods can be interleaved within a production sequence [1, 2, 3]. Discretized models can readily be used to accurately represent product geometry and support simulation at intermediate stages of these processes and these have been finding further application due to their inherent compatibility with general purpose graphics processing unit (GPGPU) compute platforms [4, 5, 6, 7] . Unfortunately, most work in process simulation and process modeling have not focused on these voxelized descriptions. One area of interest in particular is in time-based modeling of material removal during machining, as well as in mapping forces dissipated from such voxel-based models. The objective of the current research is to develop a generalized discrete framework for the simulation of the interaction between an arbitrary cutting tool and the body of material being machined.

In mechanistic models, calculation of chip thickness is often approximated using Martellotti's original approximation of chip thickness, which considers motion of cutting flutes as they travel along a trochoidal path [8]. The form of the mechanistic model has been adapted and improved over the years, but one common factor is that the cross-section of the chip area is calculated from the product of the instantaneous chip thickness and the length of the discrete disk of the cutting tool [9]. Recent improvements in identifying the cutting force coefficients used have been made by Altintas [10] and Azeem et. al [11]. New efforts are being made into more accurately predicting the cutting forces with cutting force coefficients that are distributed along the depth of cut of the tools axial immersion [12]. The limitations of the basic mechanistic model are apparent from the fact that the unique models have been used for every tool-workpiece engagement scenario and variation of the cutting tool geometry.

Childs helped develop the enveloped cutter geometry commonly used in modern CAM software [13]. Geometric models of cutting tool geometry and how they interact with the workpiece geometry were developed in [14, 15, 16, 17, 18]. The major drawback of these models that represent both the tool and workpiece with surface geometry, is that the cutter engagement regions require extremely

complex boundary equations for estimating the regions of contact from the projection of the workpiece surface onto the surface of the cutting tool. In fact, once the circumstances of the cutter engagement expand beyond even some of the simplest prismatic cuts against rectangular workpiece volumes, the equations of the boundary must be solved with numerical solvers [19]. Other formulations of workpiece and end mill geometry have been proposed such as the z-map technique employed in Ref. [20]. This 2.5D method works well for medium to low complex part geometries, but since the model cannot handle overhangs or similar geometries that occur in the z-axis direction. As such, this model cannot be extended to handle 5-axis machining. The formulation of the tool geometry proposed by Engin and Altintas in Ref. [18] is important in that it is one of the few models that attempts to directly account for the helical flutes of the tool in the definition of the tool geometry. The incorporation of these models with the development of CAM software was carried out by again Altintas [15], and separately by Armarego and Deshpande [21]. Improvements in these computational methods were made by El-Mounayri et al. [22] with the incorporation of generalized geometric modeling of the workpiece with the tool.

A major limitation of traditional formulations of the mechanistic model is that it has to be specialized for each tool geometry and unique workpiece engagement circumstance. Due to this, the model works only for the cases it has been validated against and was only able to provide limited insight to how to improve general machining practices. This lack of robustness was partially solved by the advent of using envelope surfaces of the tools in the model to allow for the local engagement of the tool with the workpiece to be calculated from the projection of the workpiece geometry onto the tool surface. However, even with this technique, instantaneous local engagement cannot be predicted nor can chip thickness. The geometric tool model, developed in Ref. [19], does take steps to incorporate the outermost helix of the cutting flute in the cutter engagement region mapping, but it still suffers from same need for a numerical solver when the equations of the engagement region's bounds become too complex for analytical solutions. As such, it is clear that surface models do not present a computationally efficient method of calculating the interaction between the milling tool and the workpiece material. Another drawback of traditional machining models is they become more complex with the increasing complexity of the tool and workpiece contact or simply with the increase in the complexity of the tool path itself. This increased complexity means the boundary equations become more difficult and ultimately the computational load of the model increases. Ultimately, as the complexity of the interaction in surface based models increases the need for numerical solvers and computational power grows.

One growing alternative is the use of voxels in the simulation of the machining process. The emergence of automated path planning software for 5-axis CAM solutions, such as SculptPrint [4], is an indicator of the advantages that can be obtained by the choice of a different data structure from the traditional surface representation models. Not only do voxels allow for true solid modeling of the workpiece volume, but they also increase the ease and effectiveness of creating offsets of the part surfaces for the development of tool paths and analyses. Other works in the realm of voxel based machining models include Refs. [23, 6, 5]. The approach used in the present work uses a voxelized workpiece to represent the workpiece material with a floating point point-cloud of the tool.

As the advances in modern computing continue to grow, demand for completely digital manufacturing processes is becoming a common demand of industries. Thus, leveraging the recent leaps and bounds in computer hardware for the calculation of machining simulations seats well with current trends in industry. Moreover, due to the improvements in specialized computing hardware for graphics processing, the goal of simulating the interaction between a workpiece and a given tool is attainable in a discrete fashion. Leveraging the parallel processing power of multi-core CPU and GPGPU compute, voxel-based simulation of the interaction of a discrete workpiece and an arbitrary tool is possible. The use of voxel modeling for path planning in automated 5-axis machine tool paths has already been established [6, 5]. Fitting with the trend in digital manufacturing, a model of tool-

workpiece interaction based on a voxel framework can co-opt robust computer graphics processes for accelerating computation in this significant area of opportunity.

With the maturation of CNC machining and the rise of hybrid processing methods, increasingly more complex toolpaths are being investigated. The specific reasons for these complex paths are numerous, but include improved material removal rates, chip thinning, and insulating against chatter or thermal effects. Simultaneously, these tool paths and their underlying tool-workpiece engagements are challenging to analyse with traditional models. It is precisely due to these drawbacks that there is a push in some literature and companies now for voxel based workpiece representations and path planning CAM software [23, 4]. These models may be able to better represent intermediate stages of manufacturing, useful for hybrid manufacturing process planning. The purpose of this research is to describe a model for simulating the interaction between a cutting tool and workpiece in the machining process. The objective of the model is to predict cutting forces and workpiece geometry resulting from each tool pass.

2 Methodology

2.1 Workpiece Model

As a cubic element, every voxel represents a finite volume of material with specific and equal side lengths. Voxels can be easily stacked and grouped in 3-dimensions, face-to-face, in aligned arrays, allowing them to be readily interpreted by computers. This allows voxels to naturally function as the simplest tessellation of 3D space. A voxel with the logical value of 'true', or in binary 1, can be thought of as a filled voxel that represents solid or space, whereas, a voxel with a value of 0 can be thought of as representing unfilled or empty space. Locating voxels by their center point avoids inherent directional bias in representing part geometry, calculating chip thickness, and subsequently cutting forces. Therefore, the voxels in current model are located by their body center point or centroid. The representation of the voxelized blanks with 3D arrays of binary data allows for the physical spacing of the voxels to be stored as a single number that represents the isotropic voxel length, w. This makes recovering the location within the part straightforward, in that the location is defined by the 3 indices of the particular voxel multiplied by the predefined voxel length. Furthermore, finding the global location of a voxel is as simple as adding the offset of the workpiece's lowest 3 voxel indices.

2.2 Cutting Tool Model

Given the large variety of cutting tool geometries used in machining, it is important that any sufficiently adaptive simulation be able to represent the range of cutting tools commonly used in machining operations. For the purpose of practicality this model will be limited to the representation of unibody helical mills. That being said the model is still capable of fully representing simple cylindrical mills, ball mills, bull-nosed mills, tipped mills, and tapered mills, as well as any valid combination of these geometries with continuous curvature.

Representation of cutter geometry is of importance at all levels of machining operations, however the level of accuracy required does differ across representations. For modern computer aided machining (CAM) programs, tools are represented by a simplification of their outermost geometry [24]. This geometry is used to identify the radially symmetric envelope of the cutting tool so that the intersection of this boundary and the part surfaces can be calculated and graphically displayed. While envelope geometry does offer a simplified representation of the cutter geometry that is computationally efficient, it lacks the necessary geometric description needed to analyze the cutting mechanics.

Calculation of the cutting tool and workpiece interactions, requires accurate representation of tool geometry. As one of the primary assumptions of orthogonal and oblique cutting models, only the outermost edge of the helical flute is assumed to be in direct contact with the machined surface of the workpiece. Accordingly, by approximating the cutting edge of the tool to be infinitely sharp,

the primary cutting surface of the tool can be modeled with the rake face leading to this cutting edge. Models for the combination of the cutter envelope and outermost edge of the helical flutes have been developed [18], but these fail to account for the remaining rake face geometry.

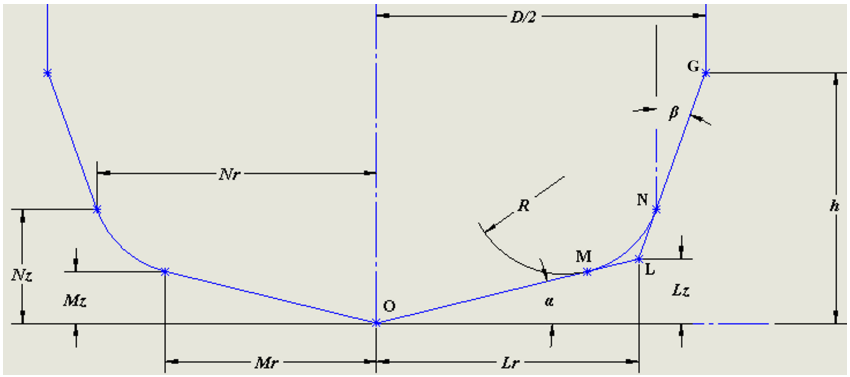

Figure 1. Milling cutter envelope geometry.

The tool model developed for this research shares some of the nomenclature employed by J. J. Childs in Ref. [13]. Starting at the tool tip specific points along the envelope of the cutter are calculated and used to anchor the helix of the tool at specific radii and distances along the tool axis. In the model presented the user defines at most five geometric parameters, *D, R,* α, β, *& h* that allow for the complete description of the tool envelope as seen in Figure 1. This reduced number of geometric parameters, allows for a more intuitive tool generation for the user by imposing curvature continuity amongst the sections of the milling tool when tools with cutter radii are used. To simplify the prior knowledge required by the user the present model does not allow for the generation of form cutters, unlike the generalized model presented by Engin and Altintas [18]. As seen in Figure 1, the five parameters can describe a general endmill of arbitrary geometry so long as the radii *R* is tangent with both line OM and line NG. In cases where the radii equal zero, this tangency is carried out by M and N residing on the same point. Calculating the center of curvature for the radii portion of the tool as the region of the cutter envelope that is defined by *R,* is handled by sorting the tool type from the user inputs.

For the user specified number of flutes and helix angle, the helical outermost edge of the tool is generated by first dividing each section of the helix into an appropriate number of discrete points with arc length ds. The angle in terms of radians of each discrete point wrapped around the tool for the first flute of the tool is calculated as the tool's spindle speed (ω) in rpm multiplied by the total time of the simulation (τ_{total}) in seconds:

$$\Phi_{referenceflute} = mod\left(\frac{\omega}{60seconds} * 2\pi * \tau_{total} - \gamma, 2\pi\right) \quad (1)$$

where the mod ensures that the tool angles are between 0 and 2π. To simplify and group the following information about the cutting tool, the 'structure' convention will be used for the tool points from this point forward. Movement ($P.tip$) along any arbitrary direction can be defined by the product of the feed rate in distance traveled per minute with a normalized vector of the feed direction ($\vec{e}_{feedrate}$) and the total time of the motion step (τ_{total}) as seen in equation (2):

$$P.tip = \frac{Feedrate}{60seconds} * \vec{e}_{feedrate} * \tau_{total} + C \quad (2)$$

where C is the user-defined starting point. The above equation just defines the location of the tool tip over the time τ_{total}. To apply the motion to the rest of the tool, the angular location of each flute with respect to the tool axis must be defined. As such, with the user defined number of cutting flutes N_{flutes}, the angular coordinates of each flute with respect to the tool's axis are defined by:

$$\Phi_{flutes} = mod\left(\Phi_{referenceflute} + (\Gamma - 1) * \frac{2\pi}{N_{flutes}}, 2\pi\right)$$
$$for\ \Gamma = 1\ to\ N_{flutes} \tag{3}$$

where again the mod limits the realm of possible angles between 0 and 2π. With the necessary position of all of the cutting flutes defined, the physical location of the discrete points for each flute can be defined as:

$$P.coord = r * \sin(\Phi_{flutes}) * \overrightarrow{e_\iota} + \mathrm{r} * \cos(\Phi_{flutes}) * \overrightarrow{e_J} + z * \overrightarrow{e_k} + P.tip \tag{4}$$

where $\overrightarrow{e_\iota}, \overrightarrow{e_J}, \overrightarrow{e_k}$ are the unit vectors for the x, y, and z directions respectively. By adding the movement of the tool's tip, $P.tip$, to equation (4), the location and translation of the tool points in time is defined. With a user defined rake angle (α_{rake}), the rake line can be created by specifying a secondary point interior of the tool. The line between the interior point and its corresponding outermost flute point is the rake line used to represent the local rake face of the tool.

The rake line is made by offsetting the tool's axis, calculating the vector between the outermost cutting edge of the tool and the offset axis, and finally calculating the vector point for the rake as the sum of the outermost cutting edge with the product of the rake face length (L_{rake}) and the normalized vector between the offset axis and the outermost cutting edge as shown in equation (5).

$$Offset\ Axis \overset{\text{def}}{=} t_{axis} = P.tip - (P.tip - P.coord) * \tan(\alpha_{rake})$$
$$P.rake = P.coord + L_{rake} * \frac{t_{axis} - P.coord}{\|t_{axis} - P.coord\|} \tag{5}$$

The mathematical description of the milling tool and its motion above is sufficient for 3-axis milling operations; however, in modern CNC machining operations, multiple rotational axes are often used. While most multi-orientation machine tools are limited to 5-axes, since there is not a consensus amongst machine tool manufacturers as to which of the primary axes should be allowed to rotate, this model will incorporate three additional rotational axes for a total of six degrees of freedom in the specification of tool movement relative to the workpiece, or simply the ability to model a 6-axis machine tool.

Let angles μ, η, & λ represent the rotations about the x-axis (A), y-axis (B), and z-axis (C), respectively. The combined rotation matrix for this motion at time τ_ξ can be defined as follows:

$$c_{\sim_\xi} = \cos(\sim_\xi) \quad \& \quad s_{\sim_\xi} = \sin(\sim_\xi)$$
$$\boldsymbol{R}_\xi = \begin{bmatrix} c_{\eta_\xi} c_{\lambda_\xi} & -c_{\eta_\xi} s_{\lambda_\xi} & s_{\eta_\xi} \\ c_{\mu_\xi} s_{\lambda_\xi} + s_{\mu_\xi} s_{\eta_\xi} c_{\lambda_\xi} & c_{\mu_\xi} c_{\lambda_\xi} - s_{\mu_\xi} s_{\eta_\xi} s_{\lambda_\xi} & -s_{\mu_\xi} c_{\eta_\xi} \\ s_{\mu_\xi} s_{\lambda_\xi} - c_{\mu_\xi} s_{\eta_\xi} c_{\lambda_\xi} & s_{\mu_\xi} c_{\lambda_\xi} + c_{\mu_\xi} s_{\eta_\xi} s_{\lambda_\xi} & c_{\mu_\xi} c_{\eta_\xi} \end{bmatrix} \tag{6}$$

Then to calculate tool orientation at time ξ the following equation is used:

$$P.coord_\xi = \boldsymbol{R}_\xi * (P.coord_\xi - P.tip_\xi) + P.tip_\xi$$
$$P.rake_\xi = \boldsymbol{R}_\xi * (P.rake_\xi - P.tip_\xi) + P.tip_\xi \tag{7}$$

With the complete models of the milling tool and the workpiece defined, the interaction between the workpiece and the cutting tool can be thought of as the deletion of voxels by the rake line of the tool once it passes into their boundary.

2.3 Ray Casting Model

The successful simulation of the machining process is largely dependent on the approach to calculating the interaction between the cutting tool and the workpiece. As the milling tool is fed forward and rotated about its axis, the rake lines will move into contact with different voxels of the workpiece. Through this superposition of the two component models, the process of removing workpiece material is mimicked by the value of the voxel in contact with the tool being changed from one to zero effectively simulating their removal from the workpiece.

In order to efficiently calculate which voxels are pierced by a rake line of the tool, a lightweight and highly optimized algorithm is needed. Although there are other computer graphics based techniques for calculating intersections, many rely on iteratively refining the local geometry around the intersection to improve the accuracy of the calculation until a point of sufficient accuracy is reached. Since the workpiece is represented down to the desired level of accuracy by voxels to begin with, there is no need to refine the local geometry in real time with the simulation. For the same reason, calculating the intersection using these methods that rely on surface geometry is also inefficient. Instead, a method is needed that can efficiently and accurately calculate which voxels have been pierced by a local rake line. As an advantage of the simplicity in representing the workpiece as a framework of cubic elements and the tool motion as distinct sets of lines at specific locations in time, the calculation of their intersection is also simple.

Ray casting is one of the most common methods used to render images and lighting. Due to the growth in the fields of computer generated images (CGI) and video games in recent decades, ray casting has become a highly developed and optimized technique. Moreover, due to the large recent advances in graphics processing unit (GPUs) hardware that can specifically accelerate 3D kernel computations by parallelizing operations across a high number of cores; uniformly discretized data such as voxelized arrays allow for very efficient computational times [25]. Typical ray casting implementations follow four primary steps. First, a ray of light is projected from a given view orientation. The projected ray is then sampled along its length for objects that may be pierced by the straight line. Next, the shading caused by this intersection is calculated and finally the objects that are pierced as well as their calculated shading are assembled along the ray back to the source to determine what is displayed along that path in the image [26]. In this proposed model the rake lines from the tool are used in place of rays of light, and the application is to determine the interaction between the tool and the workpiece. As such, no rendering steps are required.

In the proposed model a modified version of the algorithm developed by Amanatides and Woo is used to calculate the voxels pierced by each ray of the tool [27]. The advantage of this model is that even when the ray is cast through a very large number of voxels there is no need to recalculate any floating point numbers for the duration of the ray traversal. This allows the model to be approximately 10-25% faster, for small and large numbers of voxels respectively, than Bresenham based algorithms. Moreover, their algorithm proved faster than the original Cleary and Wyvill algorithm for large numbers of voxels [28]. The Amanatides and Woo algorithm is performed in two parts for each ray that is cast: ray initialization and discrete voxel traversal.

The ray initialization step checks to see if a given ray even intersects the region of the voxelized space and if so calculates the length along the ray at which the first intersection will occur. This is achieved by comparing the normalized components of the distance between the ray's origin and the bounds of the voxelized space. If the max of one component distance is less than the min of another component or vice versa, the ray will not intersect the voxelized space. This ray initialization model was introduced by Smits [29] and formalized into implementation by Williams et al. [30]. Since in most milling operations the majority of the tool is generally not in contact with the workpiece at any given instance in time, this fast initialization allows the model to avoid unnecessary calculations.

For a ray that does intersect with the voxelized space, the next step of the interaction model is to calculate what voxels the ray pierces. This is achieved by sampling points along the array and checking if they are within a solid voxel. Blind sampling along the array would be inefficient and unable to guarantee that all voxel intersections were identified. Instead, the proposed model uses a voxel traversal solution that breaks the sampling down into voxel steps in the component directions of the space. The algorithm traverses voxels according to which adjacent voxel is closest along the ray's direction. This discrete movement limits the number of floating-point calculations that need to be used during a given ray traversal. Unlike traditional ray casting algorithms, this ray will not terminate at the end of the voxel volume, but rather at the outermost point of the cutting flute. Therefore, the Amanatides and Woo algorithm [27] had to be further altered to allow the termination of the traversal once the last voxel pierced by each rake line was reached. Thus, conditions for the termination of the voxel traversal are needed. The conditions for termination of the ray are defined as follows:

$$\begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix}_{min} = \frac{\min(P.rake, \quad P.coord) \; - \; Bound_{min}}{w} \tag{8}$$

$$\begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix}_{max} = \frac{\max(P.rake, \quad P.coord) \; - \; Bound_{min}}{w} \tag{9}$$

$$condition \begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix} \geq \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix}_{min} \wedge \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix} \leq \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix}_{max} \tag{10}$$

where the minimum boundary of the workpiece is $Bound_{min}$. The conditions ensure that the voxels being traversed lie within the bounding box defined by the end points of the current ray. Keeping the conditions in terms of array indices is both appropriate given the discretization of the workpiece and efficient. Further conditions are held that require the discrete traversal to stay within the global bounds of the voxelized space. The ray casting technique described can be seen below in Figure 2. The material removal is simulated by setting voxel values to false after they have been pierced.

The last part of the tool and workpiece interaction model is the calculation of chip thickness along each rake line. Since the model of tool and workpiece interaction already steps discretely along the component directions, the model can also keep track of how many voxels set to false along each direction as well. Chip thickness, H, is calculated by summing the number of voxels traversed for each component direction and then taking the Pythagorean distance of each component total multiplied by the voxel size, w. To do this, an additional functionality is added to the traversal while loop, that is to keep a count of the number of times a step has been take for each component direction. The following equation summarizes this calculation:

$$H_{\xi,\Gamma,\zeta} = \sqrt{\left(N_{x_i} * w\right)^2 + \left(N_{y_j} * w\right)^2 + \left(N_{z_k} * w\right)^2} \tag{11}$$

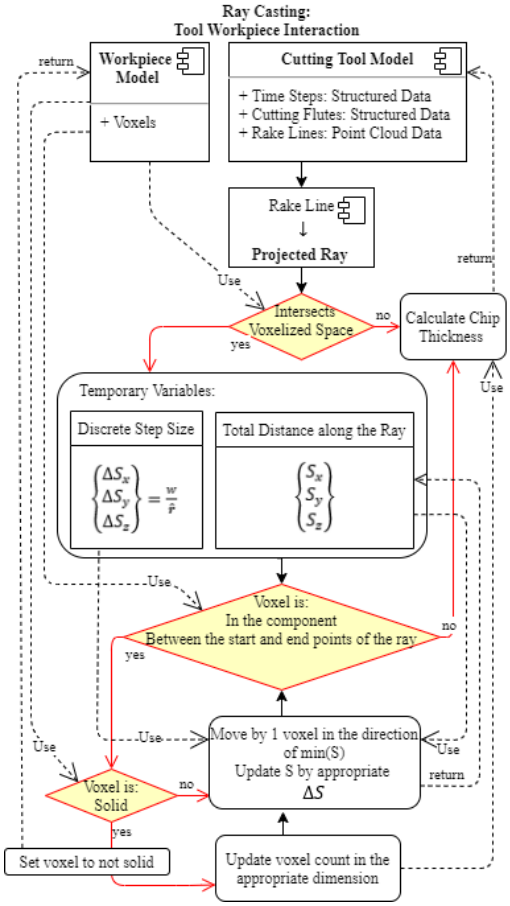

Figure 2. Block diagram of the tool and workpiece interaction algorithm.

2.4 Mechanistic Model

The cutting of metals is a plastic-flow process, but the modeling of this deformation is complex. Past research has focused on modeling the deformation zone with both thin-zone and thick-zone models. Merchant's orthogonal cutting thin shear-plane analysis is arguably the most famous analysis in these areas but requires a few fundamental assumptions that are not all strictly true for milling operations. These include that the tool tip is perfectly sharp, no built-up edge occurs, and the stress is uniformly distributed in the shear plane [31]. The model presented in this paper is based on the mechanistic model with alterations to its formulation based on the discrete nature of the data set. Similarly, the method used in the present model is based on experimentally defined cutting pressure coefficients and the discrete breakdown of the tool at increments in time and in increments along the tool's axis. However, in the present model the discretization is not only of the tool, but also of the workpiece. Because of this, some sampling and smoothing of the output data is needed to enforce a smooth force trace from the interaction of the milling tool and the workpiece material. As seen in Figure 3, the movement of a rake line through a field of voxels even in 2D results in periods where the rake line may not be in contact with a sufficient number of voxels to represent the true chip thickness of a given contact scenario.

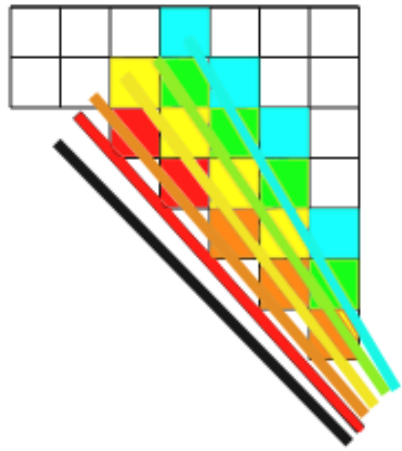

Figure 3. Illustration of how a rake line may move through and contact the voxelized workpiece in 2D.

The proposed alteration of the mechanistic cutting force model is calculated as in Ref. [19], where differential cutting forces generated can be calculated by the linear edge force model developed by Armarego [32]. The following equation shows how this linear edge force model is implemented in the proposed model:

$$\begin{pmatrix} dF_{r_{\xi,\Gamma,\zeta}} \\ dF_{t_{\xi,\Gamma,\zeta}} \\ dF_{a_{\xi,\Gamma,\zeta}} \end{pmatrix} = \begin{pmatrix} K_{rc} \\ K_{tc} \\ K_{ac} \end{pmatrix} * H_{\xi,\Gamma,\zeta} * w + \begin{pmatrix} K_{re} \\ K_{te} \\ K_{ae} \end{pmatrix} * ds \tag{12}$$

where $dF_{r_{\xi,\Gamma,\zeta}}, dF_{t_{\xi,\Gamma,\zeta}}$, & $dF_{a_{\xi,\Gamma,\zeta}}$ are the differential cutting forces in the radial, tangential, and axial cutting directions respectively. Additionally, ds is defined as the arc length of the flute between rake lines, $\zeta \; and \; \zeta + 1$, along the axis of the milling tool. The cutting force coefficients are divided amongst the cutting K_{rc}, K_{tc}, & K_{ac} coefficients of the shear zone and the ploughing K_{re}, K_{te}, & K_{ae} coefficients of the frictional contact between the tool and the workpiece [33]. Equation (12) defines the instantaneous elements of the force for every rake line, ζ, on every flute, Γ, at every point in time, ξ individually.

As discussed earlier, the sampling of chip thickness is carried out to smooth the data before computing the force prediction shown in equation (12) above. Associating this sampled chip thickness with the correct angle is accomplished by simultaneously extracting the index of the maximum chip thickness when finding the ideal (maximum) chip thickness within the sampling period. With the chip thickness associated with the correct rotational angle of the simulation, and for more complex geometries the angle off of perpendicular to the tool axis, the global forces for 3-axis milling operations can be defined by a simple transformation matrix.

$$\begin{aligned} &c_\kappa = \cos(\kappa_\zeta), \; s_\kappa = \sin(\kappa_\zeta), \\ &c_\Phi = \cos(\Phi_{flutes}), \quad \& \; s_\Phi = \sin(\Phi_{flutes}) \\ &\boldsymbol{R}_{i,j,k} = \begin{bmatrix} -s_\kappa * s_\Phi & -c_\Phi & -c_\kappa * s_\Phi \\ -s_\kappa * c_\Phi & s_\Phi & -c_\kappa * c_\Phi \\ c_\kappa & 0 & -s_\kappa \end{bmatrix} \end{aligned} \tag{13}$$

where κ is the angle from the tool axis, and Φ is the angle of the individual rake line about the tool's axis. It should be noted that since Φ is dependent on the time, ξ, flute, Γ, and rake line, ζ, the rotation matrix, $\boldsymbol{R}_{i,j,k}$, is as well. Multiplying equation (12) by the rotation matrix defined in equation (13), the value of the differential forces in the x_i, y_j, z_k directions of the global coordinate system. To extract these same differential forces for a 3+ axis machine tool the differential forces, $dF_{r_{\xi,\Gamma,\zeta}}, dF_{t_{\xi,\Gamma,\zeta}}$, & $dF_{a_{\xi,\Gamma,\zeta}}$ in the 3+axis tool coordinate system must first be transformed back to the 3-axis tool coordinate system by applying the inverse of the rotation matrix defined in equation (6). The differential forces are represented in Figure 4.

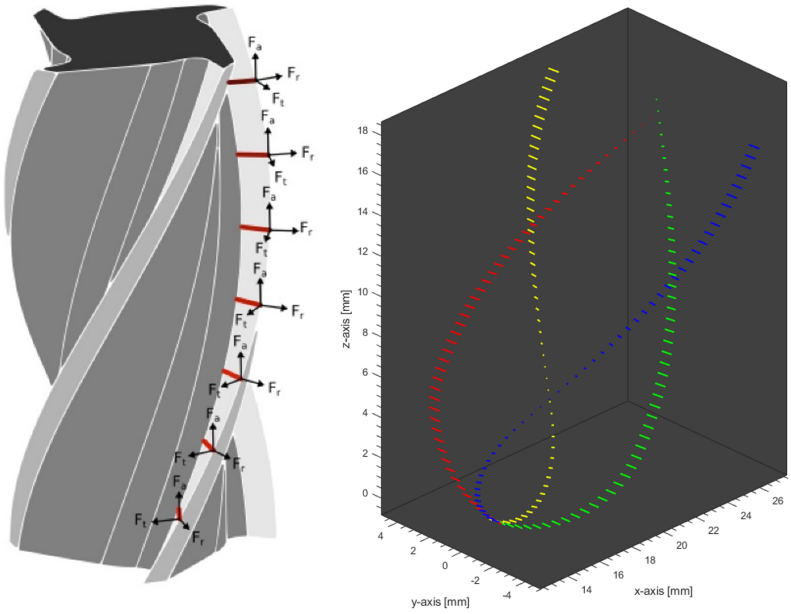

Figure 4. Example of a cutter with rake lines, shown in red, spanning the rake face of a flute, as well as the differential forces for each rake line (left). Modeled rake lines for each flute (right).

The final form of the differential force elements in the workpiece or global coordinate system are:

$$\begin{pmatrix} dF_{x_\xi} \\ dF_{y_\xi} \\ dF_{z_\xi} \end{pmatrix} = \boldsymbol{R}_{i,j,k} * \begin{pmatrix} dF_{r_{\xi,\Gamma,\zeta}} \\ dF_{t_{\xi,\Gamma,\zeta}} \\ dF_{a_{\xi,\Gamma,\zeta}} \end{pmatrix} \tag{14}$$

for 3-axis and 3+axis machine tool paths respectively. The final product is a series of lines running down the rake face of a milling cutter similar to the red lines spanning the rake face of the example cutter shown in. The spacing of these example rake lines does not represent the rake line spacing of the tools generated by the model.

3 Model Validation

To validate the model's ability to predict milling forces, results were compared with experimentally measured machining data from two different experimental test cases. The parameters for these tests are shown in Table 1. For the first test, peripheral milling was conducted in an up-milling operation on Ti-6Al-4V with a 4-flute cylindrical end mill. This data was collected in Ref. [34]. A rectangular workpiece blank $10.34\ mm\ \times 9.54\ mm\ \times 5.08\ mm$ was generated with a 0.005 mm voxel size. For each flute, the cutter was segmented along the length of the helix into 1,954 steps of equal arc

length. As an extra precaution, the simulation was run for multiple additional revolutions after moving into its full engagement with the workpiece. A half second of this fully engaged tool motion was simulated and decomposed into 32,504 discrete time steps. The cutting force coefficients for Ti-6Al-4V were the same as reported in Ref. [34] and are [K_{rc}, K_{tc}, K_{ac}, K_{re}, K_{te}, K_{ae}] = [317 N/mm^2, 1731 N/mm^2, 623 N/mm^2, 44.5 N/mm, 22.7 N/mm, 2.4 N/mm]. For the second test, a more complex tool geometry that involved down milling with a 2-flute ball mill in an $AlMgSi_{0.5}$ workpiece. The experimental data is found in Ref. [12]. The workpiece was generated to be a $4.65\,mm \times 4.01\,mm \times 4.01\,mm$ rectangular blank with a 0.00375 mm voxel width. The tool motion was subdivided into 1620 rake lines for each flute with 177496 discrete time steps. This allowed for the simulation to run for one full rotation of the cutter after it exited the workpiece to ensure that there was no longer any contact between the cutter and the workpiece at the end of the simulation. The values of the milling force coefficients used are taken directly from figures in Ref. [12] and are [K_{rc}, K_{tc}, K_{ac}, K_{re}, K_{te}, K_{ae}] = [125 N/mm^2, 575 N/mm^2, 150 N/mm^2, 6.25 N/mm, 12.5 N/mm, 1 N/mm].

Table 1. Experimental parameters.

Parameter	**Test 1**	**Test 2**
Workpiece	Ti-6Al-4V	AlMgSi0.5
Voxel size [mm]	0.005	0.00375
Workpiece size [mm x mm x mm]	10.34 x 9.54 x 5.08	4.65 x 4.01 x 4.01
Configuration	4-flute, up milling	2-flute, down milling
D [mm]	19.05	8.0
ψ [º]	30	20
α_{rake} [º]	12	12
SS [rpm]	500	4000
f_t [mm/tooth]	0.05	0.1
$r_{imm.}$ [mm]	9.525	4.0
$a_{imm.}$ [mm]	5.08	4.5

As seen in Figure 5(a), the agreement between the predicted milling forces and the forces measured in the first test is good. It is clear that the modeled machining forces compared well to the measured values both in terms of the magnitude and periodicity of each of the force components. There are small misalignments in the phase of the signal near the middle peak of force trace for the y-direction; however, this may be due to slight deflections of the cutter during the test. Given that similar misalignments of the force trace for the y-direction are found with the analytic simulation in the literature [9], this seems like a reasonable assumption. Additional minor variations are seen in the force along the feed-direction and the z-direction. These variations in the z-direction occur as the cutting tooth exits the workpiece material and are small in magnitude.

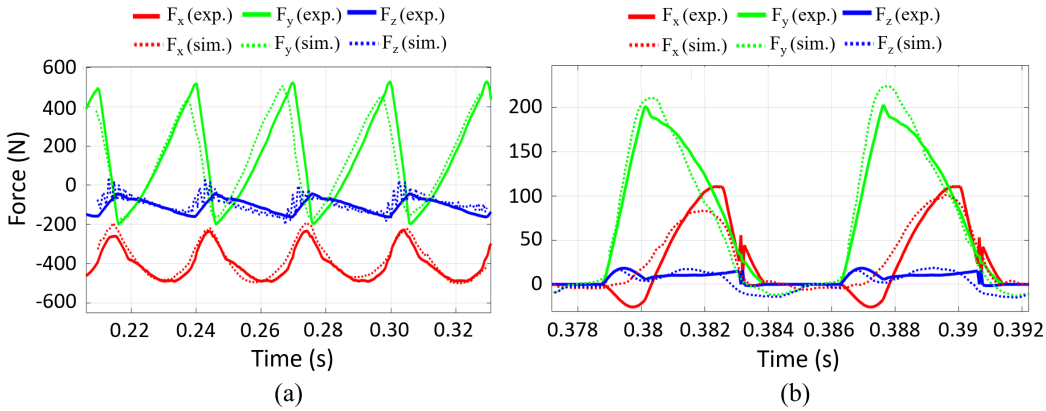

Figure 5. Comparison of milling forces calculated with the voxelized tool-workpiece interaction model and forces measured in Ref. [34] (top) and Ref. (bottom).

Figure 5(b) shows the results of the simulation compared with the results of the cutting experiment. According to Ref. [12], the discrepancies in the magnitude of the force along all three directions is primarily due to cutter runout in the experiment. Despite some lead and trailing edge fluctuations along the x-direction, the results show a good agreement with the measured force trace from the experiment. The initial sink in the forces along the x-direction is characteristic of using constant force coefficients for the length of the ball mill and can be corrected with a distributed cutting force coefficient relationship. The fluctuations in the predicted x-component force as the flute exists the material are not noticed in simulations with higher numbers of flutes or where the ball end of the mill is not fully immersed it is likely that they may be due to infinitesimal chip thickness at the trailing edge of the chip in down-milling that can be drowned out when more cutter teeth are making contact with the workpiece material. Moreover, this systematic error would likely decrease with decreasing voxel size.

4 Results

4.1 Effect of Voxel Size

The simulation's dependence on voxel size was tested based on sensitivity of material removal rate and machining forces to the model's voxel resolution. The test parameters are summarized in Table 2. The test consisted of a voxelized workpiece initially 2 mm × 2 mm x 1 mm in dimension, this created with three voxel sizes: 0.0025 mm, 0.01 mm, and 0.1 mm. The reason this milling scenario was chosen was to explore simulation accuracy when maximum chip thickness spanned over 10 voxels down to only 1 voxel. The resulting workpieces had size of 800×800×400 voxels, 200×200×100 voxels, and 20×20×10 voxels for each test case, respectively. This resulted in maximum chip load to be 120, 30, and 3 voxels wide, respectively. However, the simulation will never reach the maximum chip thickness since the tool is only partially radially immersed. Using the sinusoidal approximation of the chip thickness, the maximum chip thickness for all 3 test cases should be approximately 0.11 mm. Figure 6 presents the effects of voxel size on instantaneous undeformed chip thickness and estimated machining forces.

Table 2. Simulation parameters.

Parameter	Test		
	1	2	3
Voxel size [mm]	0.0025	0.01	0.1
Workpiece size [voxels]	800×800 ×400	200×200 ×100	20×20 ×10
Configuration	1-flute, upmilling		
D [mm]	15		
ψ [º]	30		
α_{rake} [º]	0		
SS [rpm]	1500		
f_t [mm/tooth]	0.3		
$r_{imm.}$ [mm]	1.8		
$a_{imm.}$ [mm]	1		

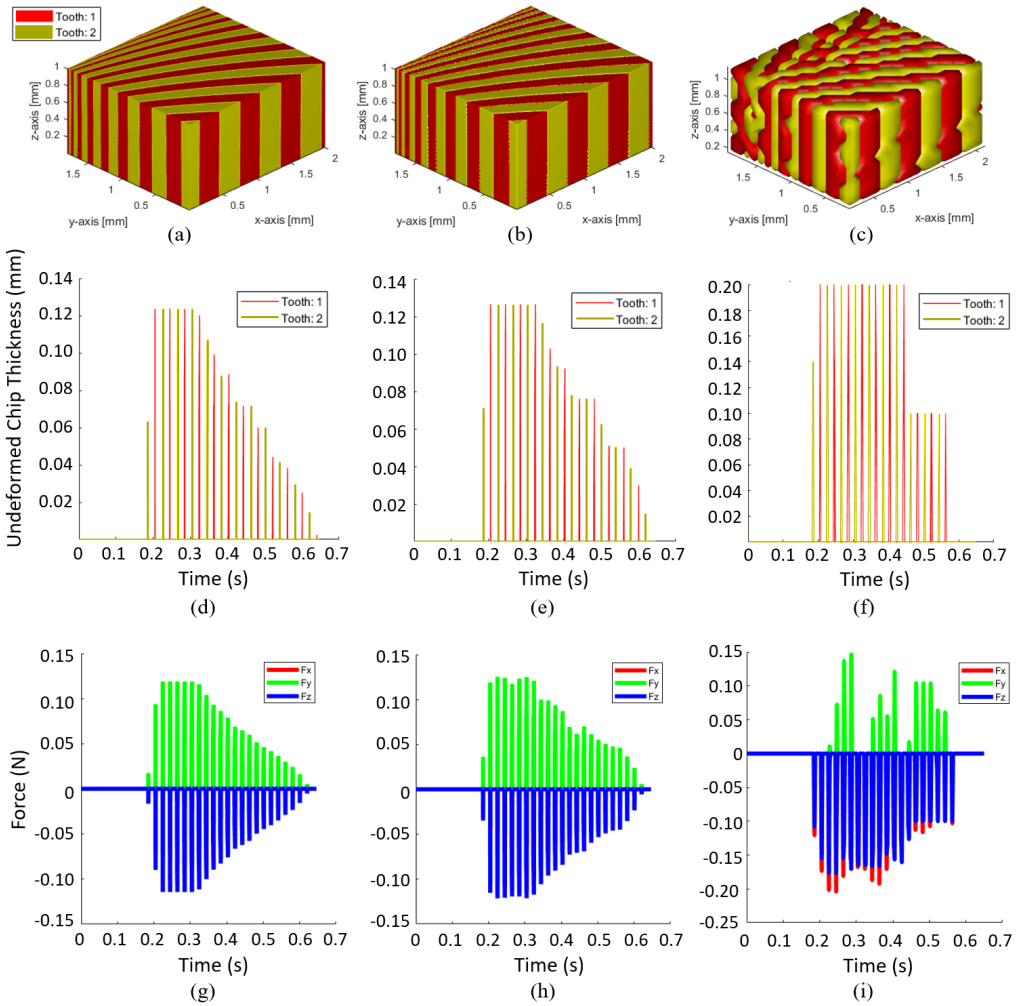

Figure 6. Undeformed chip shape, instantaneous undeformed thickness and forces for nominal undeformed chip thickness of: (a, d, g) 0.0025 voxels, (b, e, h) 0.010 voxels, and (c, f, i) 0.1 voxels.

For the voxel model, the accuracy of the prediction of milling forces is largely dependent on the accuracy of predicting the instantaneous chip thickness. Therefore, voxel size must be sufficiently small to allow be able to differentiate between the thickest and thinnest portions of the uncut chip thickness. Figure 6(a)-(f) shows the effect of voxel size on undeformed chip thickness. Comparison of Figure 6(a,d) and Figure 6(b,e) shows that small apparent differences in the simulated undeformed chip thickness resulted from variation in voxel size from of 0.0025 mm to 0.01 mm. However, at a voxel size of 0.1 mm, a clear deterioration in the visual representation of the chip thickness as well as deviation in the instantaneous chip thickness plot is evident. Similarly, the force data in Figure 6(g)-(h) shows that once voxel size reaches 0.1 mm, the expected force response is clear. These results are expected as the chip thickness varies from small to large throughout the cut. The ability for the model to accurately determine the material removal across this range of values is critical toward accurate simulation of the machining process. Thus, depending on the milling operation, the voxel size must be carefully chosen to allow for a balancing of the simulation efficiency, which prioritizes larger voxel sizes, and data resolution and model accuracy, which prioritize smaller voxel sizes.

4.2 Multi-Pass Machining

The present model also is well suited to investigate more complex configurations, such as the interactive effects of sequential machining passes, which can be difficult to model otherwise. To demonstrate these capabilities, two different scenarios were designed that incorporated roughing and finishing toolpaths. The simulation parameters for both tests are identified in Table 3. Both scenarios use a 30 mm diameter 2-fluted roughing tool, as well as 5 mm diameter 4-fluted finishing tool. The primary difference in the two tests is in the difference in feed rate and machining speed for the roughing operation. The second test has a much smaller chip load (feed rate), and as such will have less pronounced scalloping present in the workpiece. The resulting effects on predicted forces and undeformed chip thickness was evaluated using the model. To avoid any biasing of the force results and to better demonstrate the direct effect of the rough surface on the following tool path's chip thickness, the cutting force coefficients were set to 1 for all the shear elements and zero for all the ploughing elements, that is [K_{rc}, K_{tc}, K_{ac}, K_{re}, K_{te}, K_{ae}] = [1 N/mm^2, 1 N/mm^2, 1 N/mm^2, 0 N/mm, 0 N/mm, 0 N/mm]. The ploughing elements were set to zero since their application is not dependent on the calculated chip thickness. By limiting the shear cutting force coefficients to unity, the force prediction of the simulation becomes a direct representation of the effects of the calculated chip thickness in each direction of the workpiece coordinate system.

Table 3. Simulation parameters.

Parameter	Test 1		Test 2	
	Roughing	Finishing	Roughing	Finishing
# of cutting edges	4	2	4	2
D [mm]	30	5	30	5
ψ [º]	60	60	60	60
α_{rake} [º]	15	15	15	15
SS [rpm]	1000	6000	1000	6000
f_t [mm/tooth]	1.5	0.1	0.6	0.1
$r_{imm.}$ [mm]	3	0.5	3	0.5
$a_{imm.}$ [mm]	5	5	5	5

Figure 7(a) depicts the results from the roughing operation in test 1. From the figure, the material removal by each flute of the cutting tool is clear. The high feed per tooth is evident from the thick bands of each uncut chip shown. Near the back left edge of the part, the periodicity of the trochoidal tooth paths can just be made out from the curvature of the chips alternating from flute 1 and 2. As the roughing cutter moves through the workpiece along the x-direction, the different portions of the uncut chip are visible. From 0 mm to approximately 3 mm, the region of the tooth's

path through the workpiece generates the thicker region of the chip, whereas, the chips generated from approximately 4 mm to the end of the workpiece are the thin beginnings of the uncut chips for up-milling. The workpiece surface resulting from the tool pass of the 30 mm cutter at 1.5 mm/tooth is shown in Figure 7(b). The high feed rate of this tool pass has left the surface of the workpiece rough with the periodic scallops cresting above the planned workpiece surface. This is due to the motion of each cutting flute moving through the workpiece in a trochoidal path that will always result in a non-zero scallop height [35].

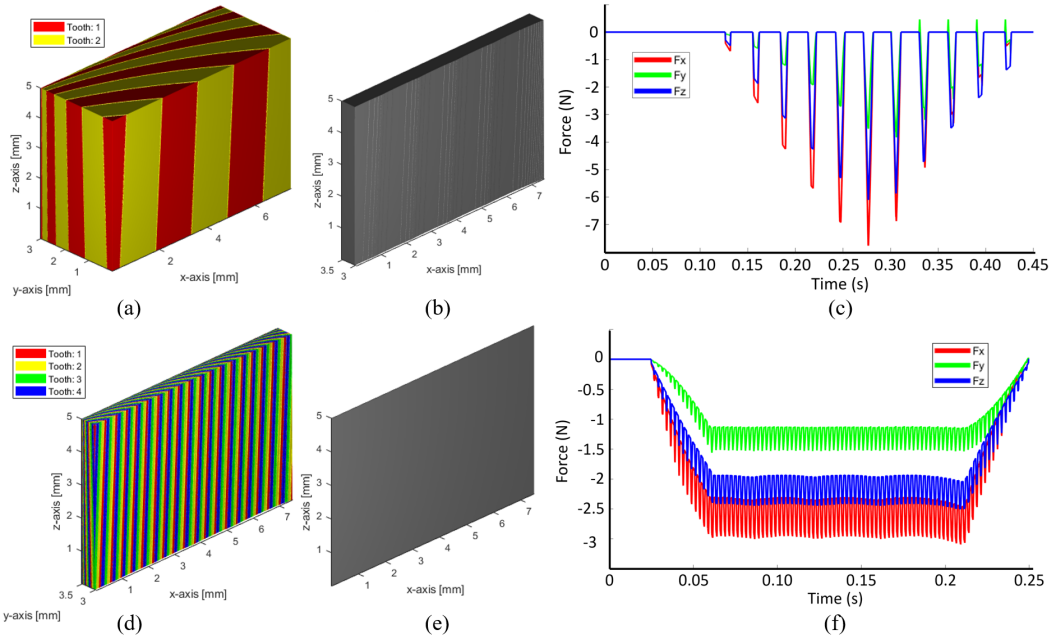

Figure 7. (a) Visualization of the material removal volume per tooth and (b) workpiece surface, and (c) forces predicted for the interaction of the 30 mm tool at (a-c) 1.5 mm/tooth and (d-f) 0.6 mm/tooth.

Figure 7(c) shows the predicted machining forces for the roughing toolpath for test 1. From the figure, the force ramps as it starts engaging the workpiece and peaks in the middle of the workpiece where it is at the longest chip engagement with the workpiece material. As the tool moves into the workpiece at around 0.125 seconds, the trend of the peak forces grows almost linearly with the tool traveling deeper into the workpiece. As the tool nears the latter half of the workpiece, the forces decline towards zero with the exception of the force in the y-direction. The force in the y-direction begins to exhibit small peaks of the force in the opposite direction due to the lagging entry of the cutting flute to the position of the center of the cutting tool. Figure 7(d)-(f) shows the effect of the finishing tool pass for the test 1. As can be seen the chip load is significantly smaller in magnitude and the workpiece surface is clearly smoother. The force dissipated shows periodic character in the maximum force levels and, it can be seen that the surface scallop topography left over from the roughing pass resulted in local increases in radial immersion of this finishing tool.

Figure 8(a)-(c) shows the results of test 2, which is carried out at a less aggressive roughing feed rate than in test 1. Figure 8(a) shows how this lower feed rate relative to scenario 1 allows for the cutter to experience smoother chip development through the milling process with the region of the uncut chip transitioning more gradually from the front of the workpiece (thick chip region) towards the back of the workpiece (thin chip region). Additionally, Figure 9(b) shows a work surface with a higher frequency of periodic surface scallops. This is due to a lower feed rate reducing the spacing between the scallops left by the cutting flutes on the surface of the workpiece compared to the higher feed rate test. Figure 8(c) shows the force dissipation and the increased number of cycles

in the force curve is due to the longer time taken to machine the workpiece at the lower feed rate. Figure 8(d)-(f) shows the effect of the finishing toolpath on the workpiece in test 2. The workpiece resulting from the finishing tool pass is again clearly smoother than the workpiece resulting from the roughing tool pass. When comparing the force predictions for this test case, it is clear that the forces evolved over smaller peak values when compared to the higher feed rate scenario, which is due to the increased material removal associated with the higher scallops of the first test case.

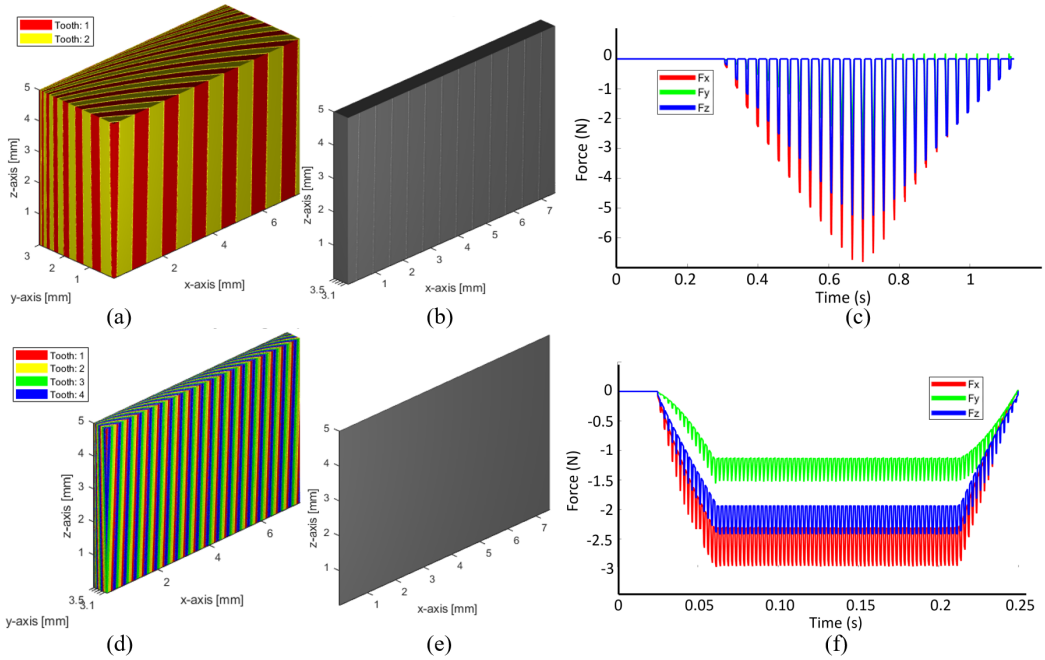

Figure 8. Forces predicted for the interaction between the 5 mm tool run at 0.1 mm/tooth with the workpiece resulting from: (a) the 30 mm tool at a feed of 1.5 mm/tooth, (b) the 5 mm tool at a feed of 0.1 mm/tooth.

5 Discussion

From the voxel size sensitivity test results, it is clear that there are limits to the model's accuracy as the voxel size approaches the order of magnitude of the maximum chip thickness due to the feed rate of the cutting tool. As these limits are approached, accurate prediction of the cutting force becomes limited primarily due to the breakdown of the calculation of the instantaneous chip thickness. From these tests, it is clear that voxel size must be sufficiently small for the accurate estimation of chip thickness at several locations along the arc of the chip. This relationship is expected due to the trochoidal path with which the cutting flute moves through the material.

In terms of the force estimation using the voxel-based model, the validation test against the experimental data on peripheral milling with a cylindrical end mill collected by Budak in Ref. [34] shows that it is appropriate to apply a mechanistic model designed for simplistic sinusoidal chip thickness estimations to the discrete data produced by this model. Furthermore, by accurately predicting the force data produced by a specific cutter in the experiment, it was shown that the model is able to sufficiently simulate the interaction of the cutting tool with the workpiece volume through considerations of only the rake face of the cutting tool. This is important as it validates the assumption that the tool geometry can be reduced to rake face geometry for the sake of the efficiency of the simulation.

The model validation showed that the model was capable of predicting forces, but for the sake of completeness, validation against down-milling operations was necessary. The additional test against the down-milling of $AlMgSi_{0.5}$ showed that the model was further capable of simulating

unusual milling operations such as the full diametrical immersion of the ball portion of a ball-mill. While the data collected by Gradiŝek et. al was not as free of cutter run-out as the Budak experimental data, it was considered important that proposed model still was able to show agreement. This is due to the fact that small discrepancies like cutter run-out are common in milling tests and real world milling operations. This is due to the general robustness of the milling operation due to the effective radial symmetry of the instantaneous tool motion about the spindle axis, assuming spindle speed is much greater than the feed rate. The irregularities of the predicted force and the force measured along the x-direction are interesting as lesser axial immersions of the ball mill with the same radial immersion, feed, and speed showed the same behaviour in the measured force in the x-direction as does the analytical model based on constant cutting force coefficients in Ref. [12]. However, the depth dependent cutting force coefficients were able to more accurately predict this measured force. This implies that the proposed model can be further improved by applying a similar regime of immersion-dependent force coefficients. Furthermore, the model already contains the requisite data for such applications since all of the points in space are already known. This means that for time-varying immersions, such as is in 5-axis freeform surface machining, the model would still be able to apply the correct cutting force coefficients.

From a generality perspective, the voxel-based simulation also is useful for modeling the interactive effects of sequential machining passes due to scalloping of the workpiece that commonly occurs in freeform surface machining and many 5-axis tool paths. This is especially critical for determining the appropriate model precision so to give a reasonably accurate prediction of tool-workpiece interaction. The data presented in Figure 7 and Figure 8 show that this voxel-based model is capable of modeling the effects of previous tool passes and also is able to predict how they will affect the current machining pass. However, it should be noted that ability to model the multi-pass surface results would be highly dependent on voxel size, tool diameter, and chip load. For linear tool paths, the distance between the lobes of the trochoidal motion of the cutting teeth is the same as the chip load [19]. Since the chip load is always significantly smaller than the diameter of the cutting tool for milling operation, the local path between any two consecutive lobes can be approximated as linear. Therefore, a simple relationship can be defined for the ratio of voxel size to tool diameter and chip load [36].

$$w < \frac{D}{2} - \sqrt{\left(\frac{D}{2}\right)^2 - \left(\frac{f_t}{2}\right)^2} \tag{15}$$

Equation (15) shows that for the simulation to be able to detect the scallops resulting from a tool path the voxel size must be significantly smaller than maximum height of the scallop from the new nominal surface of the workpiece. The particular ratio is still dependent on the tolerances of the part being made, because given the cubic geometry of the voxel the infinitesimal peak of the scallop cannot be simulated. Thus, a reasonable compromise can be found through examination of the required geometric tolerances of the part under study. The results of the subsequent passes in both Figure 7 and Figure 8 show that the surface scallops have an effect on the predicted cutting forces in both the x- and y-directions. As the chip load of the previous tool pass increases, the magnitude of the forces on the subsequent tool pass increase due to the larger scallop formation. From a general model precision consideration, the selection of voxel size should consider the above parameters to ensure accurate tool-workpiece simulation. Further, it should be noted that while the present model formulation is based on a representation employing constant voxel size, frameworks for variable precision based on hybrid dynamic tree representations [4] may provide further capability for enhanced resolution in vicinity of the machined surface and features. Expansion of the present analysis with variable precision model configurations may enable further scaling of the simulation to industrial part sizes.

6 Conclusions

In the present work, a voxel based process simulation model was presented. The validity of the model was tested against experimental data and found to be accurate within a tolerance that can be controlled with the specification of the voxel size. It was found that despite the discrete nature of the model's framework with a simple sampling regime, the continuum of measured results could be recovered sufficiently to have good agreement with the measured force data from the cutting experiment. By extension of the model in terms of axial engagement dependent coefficients, the proposed model could lend an even more in-depth analysis to sculpted surface machining where the ball mill depth will vary greatly across the tool path. On this point, the model was shown to be able to predict the effects of prior tool passes and their resulting surface roughness on the subsequent tool pass in both workpiece geometry and predicted forces. While the roughness directly resulting from the trochoidal motion of the tool's cutting teeth was discussed in terms of its generally low significance, these results showed that large surface roughness or varying surface geometry can be easily detected and accounted for in the simulation of consecutive tool passes. The nature of the voxel size to tool diameter and chip load was discussed and quantified in terms of how well the model can detect and predict the results of these unique elements of the workpiece geometry that result from the machining process. The nature of the relationship between the frequencies of the consecutive tool passes was discussed, and the constructive interference of the scalloped region of the force predictions of the subsequent pass at that location were described. The present results are of utility for understanding the criticality of model resolution on model accuracy. Analysis of voxel size sensitivity demonstrated limits to simulation accuracy as the voxel size approaches the order of magnitude of the maximum chip thickness due to the feed rate of the cutting tool. In this size regime, accurate prediction of cutting force and tool-workpiece interaction becomes limited due to breakdown of determination of instantaneous chip thickness. The present work highlights importance of sufficiently small voxel size to enable accurate chip thickness estimation during material removal and also related this criticality based on feed rate and tool diameter.

7 Acknowledgments

This work was partially supported by NSF CMMI-1646013, CMMI-1825640 and IIP-1631803.

8 Ethical Approval

Not applicable.

9 Consent to Participate

Not applicable.

10 Consent to Publish

Not applicable.

11 Authors Contributions

John Miers conducted experiments and simulations, analyzed results, and wrote the manuscript.

Thomas Tucker supported analysis of results, as well as wrote and edited the manuscript.

Thomas Kurfess supported analysis of results, as well as wrote and edited the manuscript.

Christopher Saldana conceived the study objectives, supported analysis of results, as well as wrote and edited the manuscript.

12 Funding

This work was partially supported by NSF CMMI-1646013, CMMI-1825640 and IIP-1631803.

13 Competing Interests

Not applicable.

14 Availability of data and materials

All simulation data is available upon request to the corresponding author.

15 References

[1] M. Praniewicz, T. Kurfess and C. Saldana, "An adaptive geometry transformation and repair method for hybrid manufacturing," *Journal of Manufacturing Science and Engineering,* p. 011006, 2019.

[2] Z. Zhu, V. G. Dhokia, A. Nassehi and S. T. Newman, "A review of hybrid manufacturing processes–state of the art and future perspectives," *International Journal of Computer Integrated Manufacturing,* pp. 596-615, 2013.

[3] S. T. Newman, Z. Zhu, V. Dhokia and A. Shokrani, "Process planning for additive and subtractive manufacturing technologies," *CIRP Annals,* vol. 64, pp. 467-470, 2015.

[4] T. M. Tucker, T. Kurfess and D. Konobrtskyi, "Hybrid Dynamic Tree Data Structure and Accessibility Mapping For Computer Numerically Controlled Machining Path Planning". United States of America Patent US 9971335 B2, 15 May 2018.

[5] J. Tarbutton, T. Kurfess, T. Tucker and D. Konobrytskyi, "Gouge-Free Voxel-Based Machining for Parallel Processors," *International Journal of Manufacutring Technology,* vol. 69, pp. 1941-1953, 2013.

[6] J. Tarbutton, T. Kurfess and T. Tucker, "Graphics Based Path Planning for Multi-Axis Machine Tools," *Computer Aided Design and Applications,* vol. 7, no. 6, pp. 835-845, 2010.

[7] R. Lynn, D. Contis, M. Hossain, N. Huang, T. Tucker and T. Kurfess, "Voxel model surface offsetting for computer-aided manufacturing using virtualized high-performance computing," *Journal of Manufacturing Systems,* vol. 43, pp. 296-304, 2017.

[8] E. Budak, "Analytical Models for High Performance Milling. Part I: Cutting Forces, Structural Deformations and Tolerance Integrity," *International Journal of Maching Tools and Manufacture ,* vol. 46, pp. 1478-1488, 2006.

[9] Y. Altintas, Manufacturing Automation, Cambridge: The Press Syndicate of the University of Cambridge, 2000, pp. 4-47.

[10] Y. Altintas, Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design, New York: Cambridge University Press, 2012.

[11] A. Azeem, H.-Y. Feng and L. Wang, "Simplified and Efficient Calibration of a Mechanistic Cutting Force Model for Ball-End Milling," *International Journal of Machine Tools and Manufacture,* vol. 44, no. 2004, pp. 291-298, 25 September 2003.

[12] J. Gradisek, M. Kalveram and K. Weinert, "Mechanistic Identification of Specific Force Coefficients for a General End Mill," *International Journal of Machine Tools and Manufacturing,* vol. 44, pp. 401-414, 2004.

[13] J. J. Childs, Numerical Control Part Programming, Alexandria, Virginia: Industrial Press, 1973.

[14] L. Zhen, Y. Li and S. Y. Liang, "A Generalized Model of Milling Forces," *International Journal of Macnufacturing Technology,* vol. 14, pp. 160-171, 1997.

[15] Y. Altintas and A. Spence, "End Milling Force Algorithms for CAD Systems," *Annls of the CIRP,* vol. 40, no. 1, pp. 31-34, 1991.

[16] M. Kaymakci, Z. M. Kilic and Y. Altintas, "Unified Cutting Force Model for Turning, Boring, Drilling, and Milling Operations," *International Journal of Machine Tools and Manufacture,* Vols. 54-55, pp. 34-45, 2012.

[17] W. Ferry and D. Yip-Hoi, "Cutter-Workpiece Engagement Calculations by Parallel Slicing for Five-Axis Flank Milling of Jet Engine Impellers," *Journal of Maufacturing Science and Engineerign,* vol. 130, pp. 1-12, 2008.

[18] S. Engin and Y. Altintas, "Mechanics and Dynamics of General Milling Cutters. Part 1: Helical End Mills," *International Journal of Machine tools and Manufacture,* vol. 41, pp. 2196-2212, 2001.

[19] S. D. Merdol and Y. Altintas, "Virtual Simulation and Optimization of Milling Operations Part I: Process Simulation," *Journal of Manufacturing Science and Engineering,* vol. 130, no. 051004, pp. 1-12, 2008.

[20] G. M. Kim, "Cutting Force Prediction Of Sculpted Surface Ball-End Milling Using Z-Map," *International Journal of Machne Tools and Manufacture,* vol. 40, pp. 277-291, 2000.

[21] E. J. A. Armarego and N. P. Deshpande, "Force Prediction Models and CAD/CAM Software for Helical Tooth Milling Processes. I Basic Approach and Cutting Analyses," *International Journal of Production Research,* vol. 31, no. 8, pp. 1991-2009, 1993.

[22] H. El-Mounayri, M. A. Elbestawi, A. D. Spence and S. Bedi, "General Geometric Modelling Approach for Machining Process Simulation," *International Journal of Manufacturing Technology,* vol. 13, pp. 237-247, 1997.

[23] D. Jang, K. Kim and J. Jung, "Voxel-Based Virtual Multi-Axis Machining," *International Journal of Advanced Manufacturing Technology,* vol. 16, pp. 709-713, 2000.

[24] K. F. Ehmann, S. G. Kapoor, R. E. Devor and I. Lazoglu, "Machining Process Modelling: A Review," *Journal of Manufacturing Science and Engineering,* vol. 119, pp. 655-663, 1997.

[25] S. J. Wou, Y. C. Shin and H. El-Mounayri, "Ball End Milling Mechanistic Model Based on a Voxel-Based Geometric Reresentation and a Ray Casting Technique," *Journal of Manufacturing Processes,* vol. 15, 2013.

[26] J. Pawasauskas, "Volume Visualization with Ray Casting," 18 February 1997. [Online]. Available: http://web.cs.wpi.edu/~matt/courses/cs563/talks/powwie/p1/ray-cast.htm#Introduction. [Accessed 21 October 2017].

[27] J. Amanatides and A. Woo, "A Fast Voxel Traversal Algorithm for Ray Tracing," *Proceedings in Eurographics,* pp. 1-10, 1987.

[28] B. Zalik, G. Clapworthy and C. Oblonsek, "An Efficient Code-Based Voxel-Traversing Algorithm," *Computer Graphics Forum: The Eurographics Association,* vol. 16, no. 2, pp. 119-128, 1997.

[29] B. Smits, "Efficiency Issues for Ray Tracing," *Journal of Graphics Tools,* pp. 1-12, 1999.

[30] A. Williams, S. Barrus, R. K. Morley and P. Shirley, "An Efficient and Robust Ray-Box Intersection Algorithm," *Journal of Graphics Tools,* 2005.

[31] F. Koenigsberger and A. P. Sabberwal, "An Investigation into the Cutting Force Pulsations During Milling Operations," *International Journal of Machine Tool Design and Research,* vol. 1, no. 1-2, pp. 15-33, 1961.

[32] E. J. A. Armarego and C. J. Epp, "An Investigation of Zero Helix Peripheral Up-Milling," *International Journal of Machine Tool Design,* vol. 10, pp. 273-291, 1970.

[33] M. A. Rubeo and T. L. Shmitz, "Milling Force Modelling: A Comparison of Two Approaches," *Pocedia Manufacturing,* vol. 5, no. 44, pp. 90-105, 2016.

[34] E. Budak, Y. Altintas and E. J. A. Armarego, "Prediction of Milling Force Coefficients form Orhtogonal Cutting Data," *Journal of Manufacturing Science and Engineering,* vol. 118, no. May, pp. 216-224, 1996.

[35] M. E. Martellotti, "An Analysis of the Milling Process," *Transactions of the ASME,* vol. 63, no. 8, pp. 677-700, 1941.

[36] Tyson Tool Company Limited, "Technical-Mill-Formulas.pdf," [Online]. Available: http://www.tysontool.com/tech-mill-formulas.pdf. [Accessed 20 November 2018].