

DETC2018-86344

## AN INTERACTIVE MANUFACTURABILITY ANALYSIS AND TOLERANCE ALLOCATION TOOL FOR ADDITIVE MANUFACTURING

Hannah D. Budinoff \*

Sara McMains

Department of Mechanical Engineering  
University of California, Berkeley  
Berkeley, California 94720  
Email: hdb@berkeley.edu

Alberto Rinaldi

Department of Mechanical Engineering  
University of California, Berkeley  
Berkeley, California 94720

Department of Mechanical and Process Engineering  
ETH Zürich  
Zürich, Switzerland

### ABSTRACT

*Geometric tolerances for new products are sometimes assigned without specific knowledge of the cost or feasibility of manufacturing them to the assigned tolerances, which can significantly drive up production costs and lead to delays and design revisions. We present an interactive tool that quickly estimates the manufacturability of assigned tolerances for additive manufacturing and a compact visualization to present this information to the designer. The designer can use the system to explore feasible build orientations and then adjust specified tolerance limits if all tolerances are not simultaneously achievable at a single orientation. After the designer is satisfied that the range of feasible orientations has been fully explored, a physical programming approach is used to identify a single orientation to best satisfy the designer's preferences. The calculation and visualization of the results is done in real-time, enabling quick iteration. A test case is presented to illustrate the use of the tool.*

### NOMENCLATURE

- $\mathbf{B}$  build vector
- $h$  layer thickness
- $\Delta$  specified geometric tolerance value
- $\delta$  geometric deviation on a single layer
- $\varepsilon$  error associated with a particular tolerance

$\lambda$  part cost parameter

$\theta$  polar angle relative to feature

$\phi$  azimuthal angle relative to feature

### INTRODUCTION

During the design process for a new product, designers make a myriad of decisions that determine the final configuration of the product. The decisions are often deeply interconnected. Decisions regarding the acceptable level of geometric accuracy of the part, which is often communicated by geometric tolerances, can significantly impact the product's cost, performance, and service life. The processes chosen to manufacture the product can substantially influence the geometric accuracy of the part. Despite this important interconnectivity, designers often make design decisions such as tolerance allocation without understanding how that decision affects the product costs or whether a target manufacturing process can manufacture to all their specified tolerances. If we could make the connections between design specifications and manufacturing constraints explicit and explorable, we hypothesize that designers could more efficiently analyze manufacturing trade-offs during the design stage, and consider changes to their design (or manufacturing process) in order to achieve a less costly product.

There are two important phases of this trade-off analysis: exploring feasible options to assess what levels of each objective

\* Address all correspondence to this author.

(such as cost, production time, or geometric error) are achievable; and choosing a single, most preferred choice from these feasible options. Often, no one choice can optimize all objectives simultaneously, so designers instead select a preferred solution once they feel that they cannot, with further analysis, find a better solution [1]. Exploring what level of attainment is possible for each objective, and learning how that attainment affects other objectives, is an important part of the trade-off analysis, as it enables designers to refine their goals and preferences, which might be vague or even unachievable. For example, designers might specify tight tolerances on features that are not critical to the product's performance, not realizing how much this allocation increases the cost of the product.

A tool that helps analyze manufacturing trade-offs should allow designers to fully explore and understand feasible process plans *before* choosing a single, most preferred manufacturing approach to use. However, it is also important for such a tool to minimize the mental load required from designers, who, as discussed earlier, are already making countless interconnected design decisions. A design tool should allow designers to explore feasible options quickly and help them to choose a best option, while requiring minimal unproductive, clerical interactions that don't help designers learn or communicate their preferences more accurately to a manufacturer.

Advances in technology bring new manufacturing processes for designers to consider. Additive manufacturing (AM), originally used primarily for prototyping, is increasingly being used for one-off and small batch production, which makes conformance to design specifications of growing importance for parts made using AM. In AM, process parameters such as the build orientation used to create a part can have a large influence on the quality of the produced part, making it difficult to predict what tolerances are actually achievable. The build orientation also impacts the time and cost required to build the part.

We present a system for AM that can rapidly analyze part geometry and predict errors related to geometric tolerances, based on layer thickness and build orientation. The system also quantifies part cost and production time. Cost and error information is presented using novel visualizations, which allow designers to explore feasible build orientations. Initial assumptions about designer preferences regarding part quality and build time serve as the basis of a ranking system, based on physical programming. An interactive GUI is also provided to enable the designer to selectively refine and relax specific tolerances.

Our main contributions include: an efficient ranking system based on intuitive interpretations of designer preferences with minimal initial designer input; compact visualizations to help the designer quickly interpret interconnections between manufacturing objectives; and a fast, interactive GUI for tolerance refinement and relaxation.

## RELATED WORK

Several previous studies have focused on recommending a particular build orientation for AM. Some efforts have focused on minimizing cost or support volume [2, 3] and do not consider geometric error, which is a key consideration of designers choosing between different process plans. A few researchers [4–7] have built systems to minimize error corresponding to some types of geometric tolerances, but most previous research in this area has focused on an average error metric. Some researchers [8–10] developed algorithms to optimize build direction to minimize surface roughness, while others [11–13] have minimized other average error metrics, such as volumetric error.

These average error metrics are less directly related to standard geometric tolerances used by designers to specify how much error is acceptable on key features, which limits the utility of these approaches. In industry, designers typically use geometric dimensioning and tolerancing (GD&T), defined by standards such as ASME Y14.5 [14] and ISO 1101 [15] to specify nominal geometry and acceptable geometric variation. These standards have historically been used for parts produced using traditional subtractive manufacturing methods, so although they are still necessary for parts produced using AM, there are some gaps related to new AM capabilities [16]. A draft standard relating to product definition for AM, ASME Y14.46 [17], has been released to ensure that these gaps are filled. ASME Y14.46 doesn't supersede ASME Y14.5 but instead supplements it, using the same tolerances whenever possible and adding additional controls as needed. Because GD&T remains the standard approach for specifying nominal geometry and allowable variation, it is important to understand AM manufacturability in this context.

Previous approaches are also limited because of how they deal with multi-objective optimization for AM. Typically, relative importance of the different objectives (minimize error, minimize support volume, minimize build time, etc.) is assigned by the researcher or provided by the designer as scalar weights, which isn't intuitive and can be complicated by the need to combine objectives with widely varying units. An approach called physical programming eliminates the need for scalar weighting of each objective [18]. Instead of a meaningless scalar weight, it asks the designer to break each objective into ranges of different relative desirability, which is more intuitive. Because we seek to create an interactive design tool, rather than a non-interactive orientation optimization algorithm, it is crucial that the input required from the designers is easy to understand.

Physical programming has been used as the basis of an interactive system [19]. Previous orientation optimization systems for AM are not interactive, and assume that the designer's preferences are well informed and inflexible, which is not consistent with the iterative, evolving nature of design. Barnawal, Dorneich, Frank, and Peters [20] showed that providing graphical feedback, particularly in the form of interactive, 3D visualizations, helped improve the manufacturability of designs and improve

designer confidence. Interactive AM manufacturability systems do exist [21, 22], but these systems don't provide information regarding GD&T. An interactive system that allows designers to iterate and refine their tolerance allocation while learning about the manufacturability of their design is still needed.

Physical programming has been applied to many different engineering problems in the past, as summarized by Ilgin and Gupta [23], but has not to our knowledge been utilized for tolerance allocation. As summarized by Chase [24], tolerance allocation often uses a theoretical, general equation to connect part quality and manufacturing cost. New methods are needed for tolerance allocation for AM because it is possible to improve part quality by changing the orientation of the part without necessarily increasing the build time or cost.

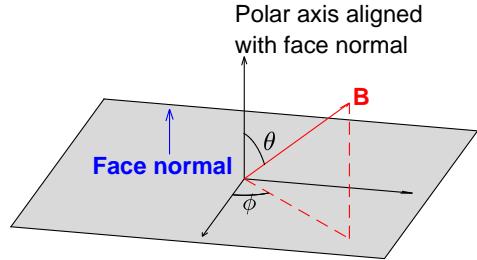
## BACKGROUND

The goal of our trade-off and tolerance relaxation/refinement tool is to elucidate the relationships between achievable geometric errors and other AM process parameters, so we begin by presenting the mathematical basis of our error calculations. Basing these calculations on mathematical equations, derived from analysis of the basic stair-stepped geometry of a part made using AM, allows our model to be generally applied to any layer-based, AM process. The mathematical basis also enables quick computation for all possible orientations. After the error calculations summary, we present a summary of the physical programming ranking used by our tool. Rather than asking designers to sort through thousands of data points to find a best option, we turn their preferences into minimization goals and constraints, which reduces the need for unnecessary user interaction.

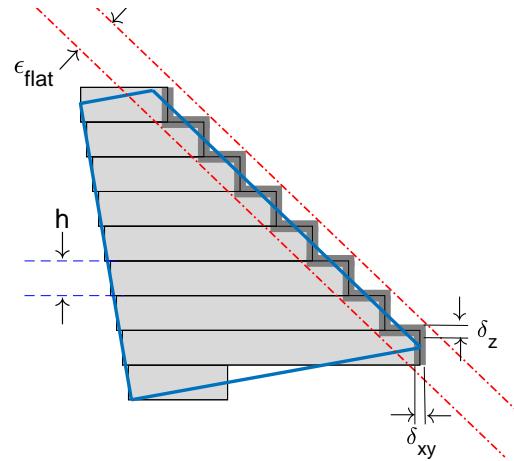
## Mathematical basis of error calculations

Error calculations for flatness, cylindricity, angularity, perpendicularity, and parallelism [5, 6, 25] are used in this paper. The detailed derivation of these error calculations can be found elsewhere, so only key equations are summarized here. For brevity, we describe the most common error regime, but when the normal vector of the feature face or datum face are aligned or closely aligned to the build vector,  $\mathbf{B}$  (the direction normal to the deposited layers), other equations are needed to calculate the error [5, 25]. We also use similarly modified versions of error calculations for cylindricity tolerances presented by Paul and Anand [6], whose equations presented only the common error regimes. All calculations are based on a spherical coordinate system, where the polar axis is aligned with the tolerated feature's face normal, and each orientation of  $\mathbf{B}$  can be described by a unique combination of the polar angle,  $\theta$ , and the azimuthal angle,  $\phi$ , as shown in Fig. 1.

For the flatness and orientation tolerances, the error is calculated as the distance between two parallel planes containing all



**FIGURE 1.** SPHERICAL COORDINATE SYSTEM ALIGNED WITH FEATURE FACE NORMAL FOR SPECIFYING BUILD ORIENTATION  $\mathbf{B}$  PARAMETERIZED BY  $(\theta, \phi)$ .



**FIGURE 2.** ILLUSTRATION OF FLATNESS ERROR,  $\epsilon_{flat}$ , FOR A SIMPLE TRAPEZOID (INPUT GEOMETRY SHOWN IN BLUE).

points on the feature surface, whereas for cylindricity, the error is calculated as the distance between concentric cylinders containing the surface, consistent with the definitions described by ASME Y14.5 [14]. This error, denoted  $\epsilon_{flat}$  for error associated with a flatness tolerance,  $\epsilon_{ang}$  for an angularity tolerance,  $\epsilon_{par}$  for a perpendicularity tolerance, and  $\epsilon_{perp}$  for a parallelism tolerance, depends on the layer thickness,  $h$ , and the orientation of the tolerated face's normal relative to  $\mathbf{B}$ . The deviation on each layer also affects the total error. The layer-level deviation orthogonal to  $\mathbf{B}$  is denoted  $\delta_{xy}$  and deviation parallel to  $\mathbf{B}$  is denoted  $\delta_z$ . These errors are seen graphically as the dark gray lines in Fig. 2.

The basic error equation for flatness and orientation errors is shown in Eq. 1. The error equation for cylindricity, shown in Eq. 2, is identical except that, instead of a normal vector, the cylinder axis vector is used, which is offset by  $\pi/2$  from the cylinders' face normals.

$$\varepsilon_{flat}, \varepsilon_{ang}, \varepsilon_{perp}, \varepsilon_{par} = (h + \delta_z) |\cos \theta| + \delta_{xy} \sin \theta \quad (1)$$

$$\varepsilon_{cyl} = (h + \delta_z) \sin \theta + \delta_{xy} |\cos \theta| \quad (2)$$

In addition to errors caused by layer-level deviations and the stair-step effect, it is important to consider how supports might change the error on the part. Whether support material is needed depends on the angle between the feature normal and **B**. The threshold angle depends on the AM process and material being used. Following prior work by Budinoff and McMains [25], we add an additional deviation term,  $\delta_s$ , to  $\delta_z$  if support is needed on the feature face. We assume support material must be manually removed and will slightly mar the surface, either with leftover support material or gouges from the removal process. This term can be turned off for processes where supports can be dissolved or are not needed.

Although we have incorporated some simple errors (related to deviations on each individual layer) into our mathematical framework, there are other process factors that can affect the quality of the produced part, such as warping, material defects, etc. These errors are difficult to predict because they tend to be specific to a given AM process, material, and even particular machine. The associated calculations can also be time intensive, making them impractical to calculate in near-real-time. For these reasons, we do not attempt to include them in this general framework. Previous studies [26, 27] have shown that orientation and layer thickness significantly influence the quality of the part itself. The effect these two factors have on geometry is more predictable than that of other process parameters. We will add complexity in future work. One example of planned future work is using different shapes to approximate the layers' profile. (For example, for FDM, the edge of each layer tends to have a rounded shape [27] that would be better approximated with a circular profile.)

In addition to the geometric errors, designers considering AM to produce their parts must also consider manufacturing cost. Predictive models have been developed to optimize part orientation to minimize cost [12, 13, 28]. In order to provide quick, real-time estimates of cost at thousands of orientations, we use cost estimates partially based on simplified estimates of build time. Following Armstrong, Barclift, and Simpson [22], with modifications to consider the volume of support material needed, the total cost to produce a part,  $\bar{C}$ , is calculated as:

$$\bar{C} = \rho C_m [\eta_s(V_s) + \eta_p(V_p)] + \bar{t} C_t \quad (3)$$

where  $\rho$  is the material density,  $C_m$  is material cost,  $\eta_s$  and  $\eta_p$  are support and infill density, respectively,  $V_s$  and  $V_p$  are the support material volume and part volume, respectively,  $\bar{t}$  is the total manufacturing time, and  $C_t$  is the cost per time charged for the manufacturing process. The total manufacturing time,  $\bar{t}$ , can be calculated as:

$$\bar{t} = \kappa \left( \frac{\eta_s(V_s) + \eta_p(V_p)}{vw\bar{h}} \right) + \frac{H}{\bar{h}} (t_r) \quad (4)$$

where  $\kappa$  is a complexity factor,  $v$  is the extrusion rate,  $w$  is the width of the filament,  $h$  is the layer thickness,  $H$  is the height of the STL file's bounding box in orientation **B**, and  $t_r$  is the additional time required to reset and move to the next layer.

Rather than using the total estimated cost,  $\bar{C}$ , directly in our optimization, we instead calculate a cost parameter,  $\lambda$ , which is used so we can provide generally applicable physical programming bounds in the next section. For a given orientation,  $\lambda$  is calculated as:

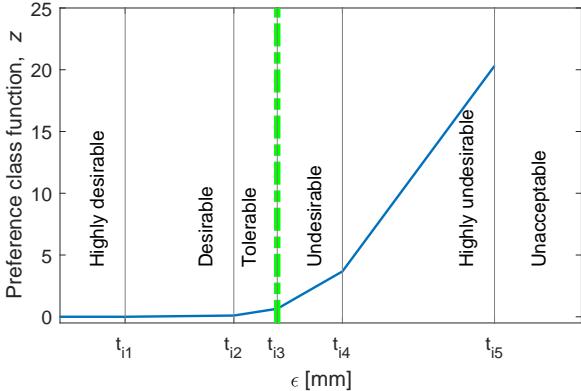
$$\lambda = \frac{\bar{C} - \min(\bar{C})}{\max(\bar{C}) - \min(\bar{C})} \quad (5)$$

where  $\bar{C}$  is the cost corresponding to that orientation, and  $\max(\bar{C})$  and  $\min(\bar{C})$  are the maximum and minimum costs for all orientations, respectively.

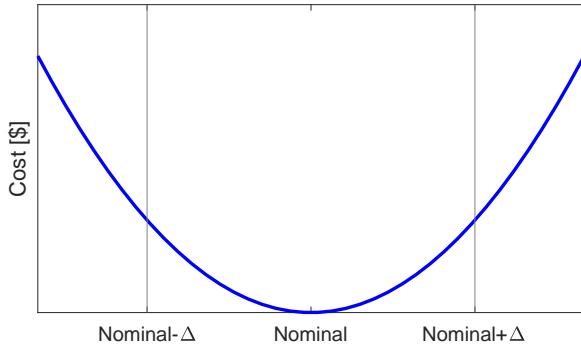
## Physical programming

The basic approach of physical programming was introduced by Messac [18]. For our approach, we used linear physical programming [29]. In physical programming, instead of asking a designer to set weights to turn a multi-objective problem into a single-objective problem, meaningful boundaries are determined between different values of each objective.

Physical programming includes several preference types, called preference class functions. If the goal is to minimize, the class type is 1; if the goal is to maximize, the class type is 2; and if the goal is to achieve a particular range, the class type is 3. There are soft and hard versions of each class, denoted S and H respectively, that can be used to characterize different levels of attainment by the designer. For each class function, a certain desirability is assigned to different ranges of objective values. For a soft class function in which the goal is to minimize the value of the objective, denoted 1S, the range of objective values is divided by boundaries,  $t_{is}$  where  $s = 1, 2, \dots, 5$  and  $i$  is equal to the index of the objective in question. These boundaries separate what range of objective values are highly desirable (HD), desirable (D), tolerable (T), undesirable (U), highly undesirable (HU), and unacceptable. An example of a 1S class function is shown in Fig. 3.



**FIGURE 3.** PREFERENCE CLASS FUNCTION. GREEN DOTTED LINE INDICATES VALUE OF  $\Delta$ .



**FIGURE 4.** TAGUCHI QUADRATIC QUALITY LOSS FUNCTION, AFTER [31].

Because a designer's goal will always be to minimize error, the class function 1S is an obvious choice to represent each tolerance error minimization objective. The shape of this class function also closely resembles the Taguchi loss function (Fig. 4), which describes the increased cost of quality loss associated with imperfection and error, especially with regards to design specification and tolerances [30]. Although the 1S class function is a good initial first guess at the designer's preferences for all tolerances, there may be errors that the designer would not consider relaxing. The designer has the option to make these tolerances mandatory, in which case they are represented by a 1H preference class function. We have also chosen to represent the goal of minimizing cost, as represented by minimizing  $\lambda$ , using a 1S class function.

Rather than asking designers to set their initial preference boundaries for each tolerance objective, we instead automatically position the boundaries of the class function relative to the initially specified tolerance,  $\Delta$ . These class boundaries are summarized in Table 1 in the row for optional tolerances,  $\varepsilon_{op}$ . These

**TABLE 1.** PREFERENCE CLASS FUNCTION PARAMETERS WHERE  $t_{is}$  SEPARATE THE DIFFERENT PREFERENCE REGIONS.

	Class	HD	D	T	U	HU
		Boundary	$t_{i1}$	$t_{i2}$	$t_{i3}$	$t_{i4}$
$\varepsilon_{op}$	1S	0.3 $\Delta$	0.8 $\Delta$	$\Delta$	1.3 $\Delta$	2 $\Delta$
$\lambda$	1S	5%	30%	75%	90%	100%
		Acceptable				
$\varepsilon_{mnd}$	1H	0	$\Delta$			

initial assumptions will be evaluated in future work to determine if they match initial preferences of designers.

After being presented with the assumed boundary positions, the designer is given the opportunity to adjust the boundaries, if desired. For designers with vague ideas regarding which tolerances are truly critical, no changes to the boundaries are needed. However, the designer has the option to change the position of the bounds if the initial guess doesn't match her preferences. The designer can choose what value of error defines the transition between the desirable region, tolerable region, undesirable region, and so on. The designer also has the option to change each tolerance to mandatory,  $\varepsilon_{mnd}$ , as described above. This phase of refining the preference structure further justifies the use of physical programming: the designer can more easily answer what range of error is tolerable to them than determining what scalar weight to assign to each of many tolerances.

Once class function types are determined and boundaries are calculated, weights can be calculated for each soft function, defining the shape of the class function for each objective, following [29]. Using these weights,  $\tilde{w}$ , and by calculating the deviational variables, denoted  $d_{is}$ , the overall minimization problem can be formulated. (Only a subset of the full physical programming implementation used by Messac [29] is described here since there are no maximization, value, or range goals needed for our application.) Our main physical programming objective for class 1S and 1H functions is summarized as:

$$\min_{d_{is}} J = \sum_{i=1}^{n_{sc}} \left[ \sum_{s=2}^5 \tilde{w}_{is} d_{is} \right] \quad (6)$$

subject to

$$\mu_i - d_{is} \leq t_{i(s-1)}; \quad d_{is} \geq 0; \quad \mu_i \leq t_{i5} \quad (1S) \quad (7)$$

$$\text{or} \quad (7)$$

$$\mu_i \leq t_{j,max} \quad (1H)$$

where  $n_{sc}$  is the number of soft class objectives,  $n_{hc}$  is the number

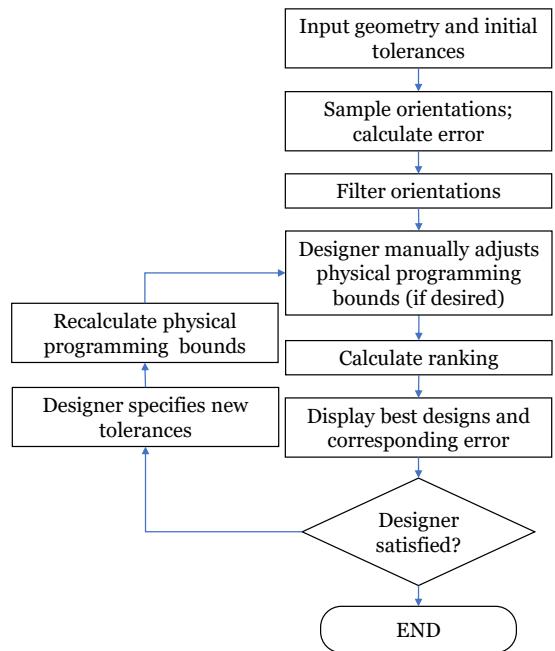
of hard class objectives,  $i = 1, 2, \dots, n_{sc}$ ,  $j = 1, 2, \dots, n_{hc}$ ,  $s = 2, \dots, 5$ ,  $x$  is the design variable vector, and  $\mu_i = \mu_i(x)$  are the design objectives.

## TOLERANCE EXPLORATION AND RELAXATION

In a typical physical programming problem, Eq. 6 would be optimized using a commercial optimization code, resulting in a single, Pareto-optimal set of objective values. For our problem, because the error calculations in Eq. 1 and Eq. 2 have a quick computation time, it is trivial to generate thousands of solutions to use for data visualization. Designers can learn more about the trade-offs between objectives by examining the range of possible orientations and the errors associated with them, exploring how each objective is related, and determining the range of errors associated with each objective. Our data visualizations (described in the next section) allow designers to learn from a wide range of solutions without needing to examine each orientation individually. This visualization and exploration can help designers to refine preferences to better identify a most preferred orientation. Interactive physical programming has been explored before [19], but the process can be somewhat tedious since only one Pareto-optimal point is generated at a time. Because of the quick computation time of our problem, hundreds of Pareto-optimal points can be generated.

The interaction with the designer begins with entering geometry and tolerance information. Then, orientations are sampled (randomly distributed points on a unit sphere as well as orientations parallel to each toleranced face's normal vector) and error associated with each specified tolerance is calculated. We use all generated orientations for plotting, as shown in the next section, including non-Pareto-optimal points, because it results in a more cohesive visualization, but the orientations are then filtered to remove dominated, non-Pareto-optimal points. Only the non-dominated points are ranked according to the physical programming weights, and the single most-preferred orientation is found and presented to the designer. This process is summarized in Fig. 5.

If the designer is satisfied with the first orientation selected for them and does not need to refine or relax tolerances, the interaction ends. If the first orientation is not satisfactory, the designer can use a GUI we developed to selectively relax the specified tolerances. The designer is presented with several data visualizations, described in the next section, to help them analyze the data. Once the designer is satisfied with the relaxed tolerances as input into the GUI, the physical programming bounds and weights are recalculated and the process can restart (see loop in Fig. 5). Because all the preference ranges are calculated from the specified tolerances, the new tolerance will change all the preference ranges for that updated tolerance. Additionally, the designer can opt to change some of the tolerances to mandatory (class function 1H) or to change the 1S boundaries.



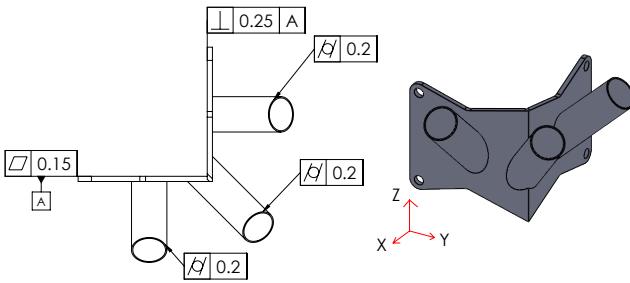
**FIGURE 5.** PROCESS FLOW FOR INTERACTION WITH TOOL.

This tolerance exploration and relaxation process minimizes initial input required from the designer by making informed guesses about preferences regarding error and cost. After quickly generating candidate orientations and presenting these orientations to the designer, the system helps the designer to explore what-if scenarios by guiding them through selectively relaxing or refining tolerances. Once the designer has explored the data enough that he is satisfied he cannot achieve a better result, the physical programming ranking identifies the orientation that best meets his needs. This exploration process will be illustrated in the next section. The designer can then indicate that the part must be built in this orientation, as detailed in ASME Y14.46 [17].

## EXAMPLE APPLICATION

To illustrate the use of the system, we evaluate a simple part and its associated geometric tolerances. The example part is a triple flag pole bracket. A flatness tolerance was assigned to one face of the baseplate, which mounts on a building. A perpendicularity tolerance and three cylindricity tolerances were also defined. Figure 6 shows the geometry and geometric tolerances. For the physical programming calculations, the preference region boundaries were set based on guidelines described earlier.

Normal vectors and STL information were input into the system. Although the STL file was used to determine the height of the part at each orientation, it was not used directly for error



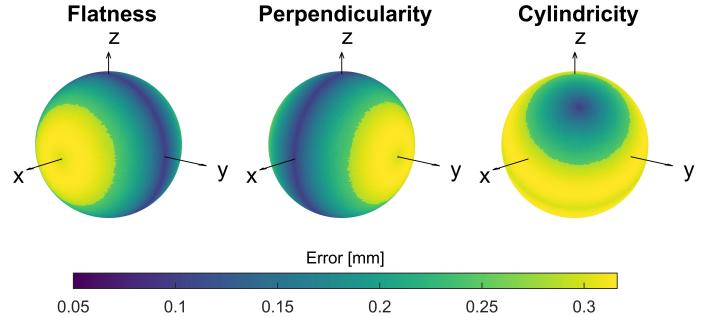
**FIGURE 6.** EXAMPLE PART AND GEOMETRIC TOLERANCES.

calculation. Depending on the resolution of the STL file, an additional error is introduced, as discussed by Paul and Anand [6]. We plan to incorporate this error into our predictions in future work. The layer-level deviations,  $\delta_{xy}$ ,  $\delta_z$ , and  $\delta_s$  were set to 0.10, 0.05, and 0.05 mm, respectively while the layer thickness,  $h$ , was set to 0.20 mm. The angle at which the support penalty was added was 45 degrees. These values were chosen to reflect layer-level error and layer thickness for an FDM process.

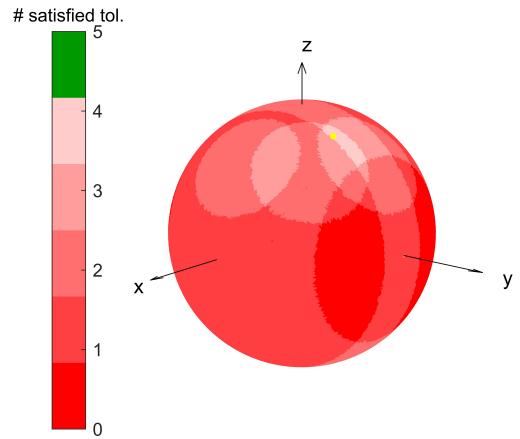
The error corresponding to each tolerance was calculated using the method described by Budinoff and McMains [25]. The error associated with the flatness tolerance, perpendicular tolerance, and middle cylinder cylindricity tolerance is shown in Fig. 7 as a function of the orientation. (The other cylindricity errors are not plotted here due to space constraints.) The coordinate axes in these figures corresponds to the coordinate axes shown in Fig 6. These plots illustrate the error functions described earlier (Eq. 1 and Eq. 2). These maps are made by evaluating the error at each orientation and plotting the error as a dot whose color is determined by the magnitude of the error at that orientation. The slight fuzziness of the coloring of these plots is due to the discrete nature of the points. Regions with increased error due to supports being needed are visible in the left, right, and sides of the three plots for flatness, perpendicularity, and cylindricity respectively.

In this example, the initial specified tolerances were not all simultaneously achievable at any particular tolerance, as shown in Fig. 8. The color scaling of this sphere, developed by Rinaldi [32], shows orientations where all tolerances are met as green, and orientations where some tolerances are not met as shades of red. Even though there is no orientation where all tolerances are simultaneously satisfied (visualized as no green region on the sphere), using the physical programming ranking of the Pareto-optimal points, we can show the designer a particular orientation that best meets her needs. For this example, that orientation was calculated to be [0.19, 0.47, 0.86] (shown as a yellow square in Fig. 8 and Fig. 9). The object rotated to this orientation is shown in Fig. 12(i).

The errors at this orientation are summarized in Table 2,



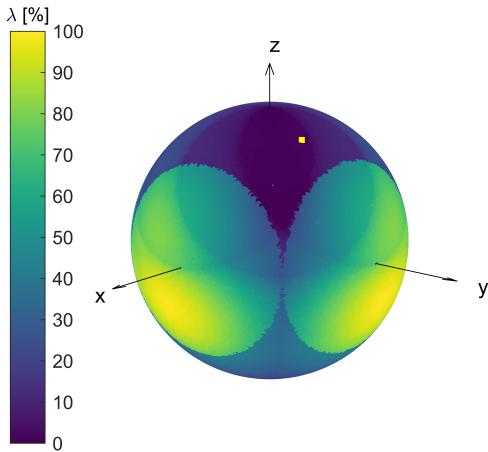
**FIGURE 7.** GEOMETRIC ERRORS ON EXAMPLE PART.



**FIGURE 8.** NO ORIENTATION SATISFIES ALL 5 INITIAL TOLERANCES. PARETO-OPTIMAL ORIENTATION [0.19, 0.47, 0.86] IS MARKED WITH A YELLOW SQUARE.

along with the error at the initial orientation shown in Fig. 6, [0, 0, 1], the orientation of the part used by the designer during geometric modeling. Comparing the naive, initial orientation to the orientation output by the program, [0.19, 0.47, 0.86], the errors associated with the flatness and perpendicularity tolerances have increased, but both are in the tolerable range. The cylindricity error has decreased for all cylinders. At this orientation, the inside of the cylindrical bosses no longer need support, which decreases the error on those features. The cost parameter has increased only slightly.

At this point, consider the case that the designer chooses to relax the tolerances slightly so that the error associated with this “best” orientation falls within the newly specified tolerances. However, the designer wants to better understand how relaxation of particular tolerances affects the number of feasible orientations. Figure 10 shows the GUI, with a set of relaxed tolerances for the example problem.



**FIGURE 9.** EXAMPLE PART COST PARAMETER. PARETO-OPTIMAL ORIENTATION  $[0.19, 0.47, 0.86]$  IS MARKED WITH A YELLOW SQUARE.

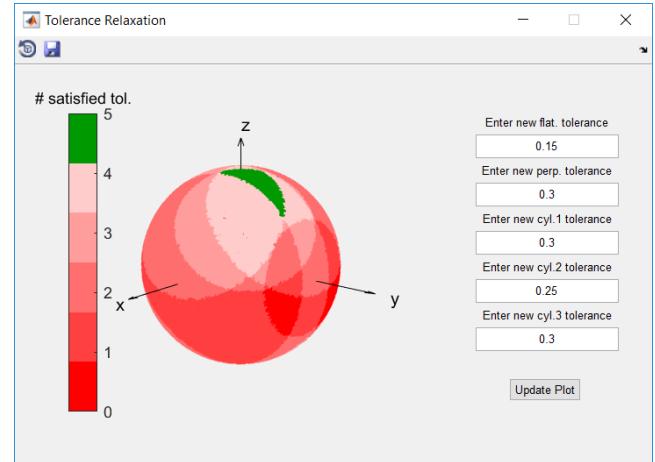
**TABLE 2.** ERRORS AT NAIVE & OPTIMAL ORIENTATIONS.

<b>B</b>	$\epsilon_{flat}$	$\epsilon_{perp}$	$\epsilon_{cyl1-3}$	$\lambda$
$[0,0,1]$	0.10	0.10	0.25,0.28,0.25	0%
$[0.19, 0.47, 0.86]$	0.15	0.21	0.24,0.18,0.18	1%
$[-0.10, -0.34, -0.94]$	0.13	0.18	0.24,0.21,0.20	2%

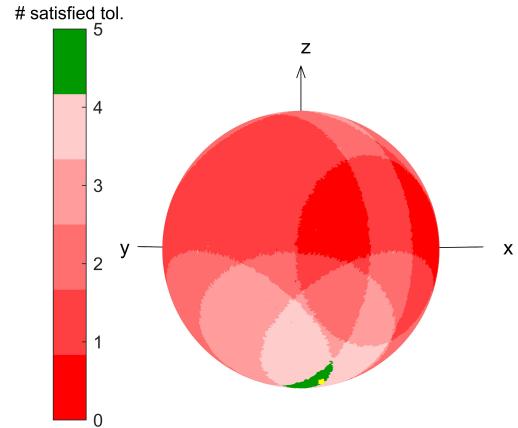
After exploring the effect of relaxing different tolerances, the designer decides to relax the cylindrical tolerances to 0.25 mm each. The designer does not switch any tolerances to the 1H class and the boundaries of the 1S class functions are unmodified. Based on these inputs, the feasibility plot is regenerated (Fig. 11). A yellow square at the very bottom of the sphere in Fig. 11 represents the new preferred orientation,  $[-0.10, -0.34, -0.94]$ , output by the program, based on the revised tolerances. At this new preferred orientation, there is slightly increased error of the cylindrical tolerances, but the flatness and perpendicularity error has been improved over the previous orientation (summarized in Table 2). The part is displayed in this new preferred orientation in Fig. 12(ii). At this new orientation, all five errors are within the new tolerances, as opposed to the initial, naive orientation, where only the flatness and perpendicularity error satisfied the original tolerance allocation.

## DISCUSSION

The analysis of the example part demonstrates the system's utility for exploring the manufacturability of parts and their as-



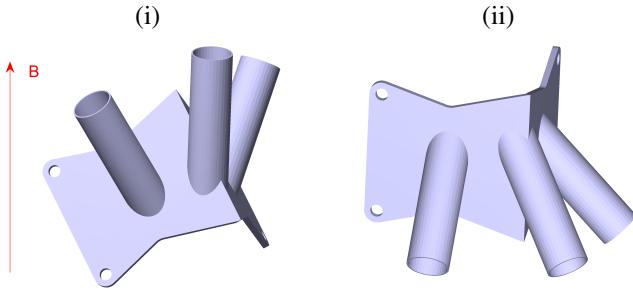
**FIGURE 10.** TOLERANCE RELAXATION GUI.



**FIGURE 11.** FEASIBLE ORIENTATIONS VISIBLE WITH REVISED TOLERANCES. PARETO-OPTIMAL ORIENTATION  $[-0.10, -0.34, -0.94]$  IS MARKED WITH A YELLOW SQUARE (BACK SIDE OF SPHERE WITH SHOWN).

sociated geometric tolerances. This tool supports a more objective, informed decision making process. Rather than subjective or guess-and-check approaches for choosing an orientation, the tool allows a designer to visualize each objective for all possible orientations at once, which aids in faster evaluation. These visualizations and the interactive nature of the tool make trade-offs between geometric accuracy and production time explicit.

Our visualization allows for quick analysis showing not only the feasible orientations but also the *almost* feasible orientations, where some but not all specified tolerances are met. It is hoped that this visualization, along with the interactive GUI, will help designers quickly explore the relationships between each objec-



**FIGURE 12.** EXAMPLE PART AT PARETO-OPTIMAL ORIENTATIONS (i) [0.19, 0.47, 0.86] & (ii) [-0.10, -0.34, -0.94].

tive. However, there are other options for presenting designers with visual analyses of manufacturability information. Rinaldi has previously used a 2D projection of the orientations [32]. We plan to explore other forms of data visualization, including parallel coordinates, in the future.

In order to examine the effectiveness of the data visualizations and the tool itself, we hope to test the proposed tool with actual designers in the near future. We assume that designers, when presented with manufacturability information, will be willing to selectively relax less critical tolerances in the manner described in this paper. Because the tool can allow the designer to examine ranges of options, rather than immediately selecting one orientation based on assumed preferences as previous tools have done, we hypothesize that the designer will feel more satisfied with the final, optimized orientation.

One benefit of our tool is its generality: it can be applied to many different AM processes, incorporating machine-specific deviations, layer thickness, and support generation cut-off angles into the error calculations. Experimental verification, which is planned as future work, will help refine these mathematical analyses.

Another benefit is the tool's computational speed. Our error calculations can be computed quickly, enabling designers to interactively explore trade-offs and tolerance allocation during the highly iterative design stage. Previous optimization tools for AM process planning are slow and not designed to be interactive. In our tool, the initial computation of the error is in near real-time (generating 10,000 orientations and calculating error for this example took 1.5 seconds on a laptop) and the relaxation GUI can update the feasibility sphere in real time, because the same orientations and error calculations are re-used and only plotting color changes are made, after comparing the already-calculated-error to the revised tolerances. Calculation of cost is slower (150.7 seconds for 10,000 orientations) due to the calculation of support volume, but this calculation is only performed once and is not re-calculated during tolerance relaxation. The computation time could be reduced using GPU techniques [33].

The tool also requires minimal initial input from the designer, making educated guesses about the designer's goals. The designer can explore feasible solutions and then refine these assumed preferences. The time the designer spends interacting with the tool is spent productively refining the preference structure and exploring the data, not tediously defining preferences or guessing at objective weights.

## CONCLUSION

Designers must analyze and optimize often conflicting objectives and requirements while designing products and preparing for production. As part of design for manufacturing, designers seek to minimize cost while maximizing quality of the parts that are produced. In order to do this, trade-offs of quality and cost must carefully analyzed. As AM becomes an increasingly viable option for production parts, it is necessary to develop tools specifically for AM to assist designers in trade-off analysis and subsequent decision making.

We have presented a tool that allows designers to predict achievable tolerances for parts produced using AM. Designers can explore which orientations enable them to meet all specified tolerances while also considering the printing time and associated cost. A ranking system based on meaningful, easy-to-understand interpretations of designer preferences was implemented to minimize demands on the designer's time and patience. Compact data visualizations were presented to quickly convey a large amount of information regarding objectives and their interconnectivity. Additionally, an interactive, iterative GUI for tolerance relaxation was presented so designers can learn and shape their preferences based on what-if scenarios of tolerance refinement and relaxation.

## ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1106400. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Alberto Rinaldi thanks Prof. Dr. Kristina Shea from the Engineering Design and Computing Laboratory of ETH Zürich for her approval for this work, and the Zeno Karl Schindler Foundation for the financial support of his time at University of California, Berkeley.

## REFERENCES

- [1] Miettinen, K., Ruiz, F., and Wierzbick, A. P., 2008. "Introduction to Multiobjective Optimization: Interactive Approaches". In *Multiobjective Optim.* Springer, Berlin.

- [2] Barclift, M., Armstrong, A., Simpson, T. W., and Joshi, S. B., 2017. “CAD-Integrated Cost Estimation and Build Orientation Optimization to Support Design for Metal Additive Manufacturing”. In Proc. ASME 2017 IDETC/CIE Conf.
- [3] Luo, Z., Yang, F., Dong, G., Tang, Y., and Zhao, Y., 2016. “Orientation optimization in layer-based additive manufacturing process”. In Proc. ASME 2016 IDETC/CIE Conf.
- [4] West, A. P., Sambu, S. P., and Rosen, D. W., 2001. “Process planning method for improving build performance in stereolithography”. *Comput. Des.*, **33**(1), pp. 65–79.
- [5] Arni, R., and Gupta, S. K., 2001. “Manufacturability Analysis of Flatness Tolerances in Solid Freeform Fabrication”. *J. Mech. Des.*, **123**(1), pp. 148–156.
- [6] Paul, R., and Anand, S., 2011. “Optimal part orientation in Rapid Manufacturing process for achieving geometric tolerances”. *J. Manuf. Syst.*, **30**(4), pp. 214–222.
- [7] Das, P., Chandran, R., Samant, R., and Anand, S., 2015. “Optimum Part Build Orientation in Additive Manufacturing for Minimizing Part Errors and Support Structures”. *Procedia Manuf.*, **1**, pp. 343–354.
- [8] Ahn, D., Kim, H., and Lee, S., 2009. “Surface roughness prediction using measured data and interpolation in layered manufacturing”. *J. Mater. Process. Technol.*, **209**(2), pp. 664–671.
- [9] Frank, D., and Fadel, G., 1995. “Expert system-based selection of the preferred direction of build for rapid prototyping processes”. *J. Intell. Manuf.*, **6**, pp. 339–345.
- [10] Thrimurthulu, K., Pandey, P. M., Reddy, N. V., and Venkata Reddy, N., 2004. “Optimum part deposition orientation in fused deposition modeling”. *Int. J. Mach. Tools Manuf.*, **44**(6), pp. 585–594.
- [11] Moroni, G., Petrò, S., and Polini, W., 2017. “Geometrical product specification and verification in additive manufacturing”. *CIRP Ann. - Manuf. Technol.*, **66**(1), pp. 157–160.
- [12] Xu, F., Loh, H., and Wong, Y., 1999. “Considerations and selection of optimal orientation for different rapid prototyping systems”. *Rapid Prototyp. J.*, **5**(2), pp. 54–60.
- [13] Alexander, P., Allen, S., and Dutta, D., 1998. “Part orientation and build cost determination in layered manufacturing”. *Comput. Des.*, **30**(97), pp. 343–356.
- [14] ASME, 2009. Dimensioning and Tolerancing Y14.5.
- [15] ISO1101, 2017. Geometrical product specifications (GPS).
- [16] Ameta, G., Lipman, R., Moylan, S., and Witherell, P., 2015. “Investigating the Role of Geometric Dimensioning and Tolerancing in Additive Manufacturing”. *J. Mech. Des.*, **137**(11), p. 111401.
- [17] ASME, 2017. Product Definition for Additive Manufacturing Y14.46.
- [18] Messac, A., 1996. “Physical programming - Effective optimization for computational design”. *AIAA J.*, **34**(1), pp. 149–158.
- [19] Huang, H. Z., Tian, Z. G., and Gu, Y. K., 2004. “Reliability and Redundancy Apportionment Optimization Using Interactive Physical Programming”. *Int. J. Reliab. Qual. Saf. Eng.*, **11**(03), pp. 213–222.
- [20] Barnawal, P., Dorneich, M. C., Frank, M. C., and Peters, F., 2017. “Evaluation of Design Feedback Modality in Design for Manufacturability”. *J. Mech. Des.*, **139**(9), p. 094503.
- [21] Nelaturi, S., and Shapiro, V., 2015. “Representation and analysis of additively manufactured parts”. *CAD Comput. Aided Des.*, **67-68**, pp. 13–23.
- [22] Armstrong, A. P., Barclift, M., and Simpson, T. W., 2017. “Development of a CAD-Integrated Cost Estimator to Support Design for Additive Manufacturing”. In Proc. ASME 2017 Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.
- [23] Ilgin, M. A., and Gupta, S. M., 2012. “Physical programming: A review of the state of the art”. *Stud. Informatics Control*, **21**(4), pp. 359–366.
- [24] Chase, K., 1999. Tolerance allocation methods for designers. Tech. Rep. 99-6, Brigham Young University.
- [25] Budinoff, H., and McMains, S., 2018. “Prediction and visualization of achievable orientation tolerances for additive manufacturing”. *Procedia CIRP*.
- [26] Mahmood, S., Talamona, D., Goh, L. K., and Qureshi, A. J., 2016. “Fast Deviation Simulation for Fused Deposition Modeling Process”. *Procedia CIRP*, **43**, pp. 327–332.
- [27] Boschetto, A., Giordano, V., and Veniali, F., 2012. “Modelling micro geometrical profiles in fused deposition process”. *Int. J. Adv. Manuf. Technol.*, **61**(9-12), pp. 945–956.
- [28] Tedia, S., and Williams, C. B., 2016. “Manufacturability analysis tool for additive manufacturing using voxel-based geometric modeling”. In Solid Free. Fabr. Symp. An Addit. Manuf. Conf.
- [29] Messac, A., 2015. *Optimization in Practice with MATLAB®: For Engineering Students and Professionals*. Cambridge University Press, New York, NY.
- [30] Taguchi, G., 1986. *Introduction to Quality Engineering*. UNIPUB/Kraus International.
- [31] Kalpakjian, S., and Schmid, S., 2001. *Manufacturing Engineering and Technology*, 4th ed. Prentice-Hall, Upper Saddle River, NJ.
- [32] Rinaldi, A., 2017. “Design methodology for geometric dimensioning and tolerancing in additively manufactured parts”. Master’s thesis, ETH Zurich.
- [33] Khardekar, R., and McMains, S., 2006. “Fast Layered Manufacturing Support Volume Computation on GPUs”. In Proc. ASME 2006 IDETC/CIE Conf.