Computer-Aided Design 46 (2014) 129–137

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Geometric computation and optimization on tolerance dimensioning

Songgang Xu *, John Keyser

Department of Computer Science and Engineering, Texas A&M University, United States

HIGHLIGHTS

- LPGUM model is extended to propose a new complete representation of tolerances.
- A new analysis method for the tolerance estimation via geometric computations.
- A new optimization method for improving a dimension scheme to reduce tolerances.

ARTICLE INFO

Keywords:
Tolerance analysis
Process planning
Geometric optimization

ABSTRACT

This paper presents an efficient geometric method of tolerance analysis for optimizing dimensioning and providing an optimal processing plan for a discrete part. Geometric primitives are used to represent part features, and dependencies in the dimensions between parts are represented by a topological graph. The ordering of these dependencies can have a significant effect on the tolerance zones in the part. To obtain tolerance zones from the dependencies, the conventional parametric method of tolerance analysis is decomposed into a set of geometric computations, which are combined and cascaded to obtain the tolerance zones in the geometric representations. Geometric optimization is applied to the topological graph in order to find a solution that provides not only an optimal dimensioning scheme but also an optimal plan for manufacturing the physical part. The applications of our method include tolerance analysis, dimension scheme optimization, and process planning.

1. Introduction

In parts manufacturing, the quality of finished parts is determined by both design and manufacturing tolerances, which determine the dimensional and geometric properties of the part [1]. Engineers are required to select appropriate machining processes and equipment so that the requirements of the design tolerances are met. This process is usually called the *process planning*. The traditional way to design an appropriate process plan is a time consuming task that requires the engineer to iterate the process of designing, testing, and modifying the solution until all the design requirements are satisfied. Recently, computer aided process planning (CAPP) has drawn considerable attention as a way to simplify this process. Key techniques related to CAPP include tolerance modeling, tolerance analysis, and tolerance allocation. The tolerances determined in the design phase serve as the basis for the manufacturing tolerances, which are used to reflect individual manufacturing operations. The focus of this paper is on design tolerances.

1.1. Prior work

A key part of tolerance modeling is representing the zone within which geometric characteristics of a model may vary. These tolerance zones are usually represented as simple geometric entities that are guaranteed to bound the features of the model [2,3]. They can be thought of as a representation of the uncertainty in the geometric position. Geometric variations can also be modeled as higher dimensional geometric objects, such as polytopes or dual-cones, which represent the region as intervals of the coefficients of their algebraic parameterization [4–6]. Tolerances of part features such as form, orientation, and size can be represented in this way, though further computation on such models can be very complicated.

Tolerance analysis covers the techniques that compute the variations of tolerances for the worst case estimation or statistical expectation. Broad reviews of the area are available [7,8]. Though many techniques can be employed for tolerance analysis in the design phase, tolerance propagation is the core part of process planning. Tolerance propagation refers to the determination of one tolerance zone based on others. When manufacturing one feature of a part, engineers have to use other features as the geometric or dimensioning references. Thus, tolerances in the references must

* Corresponding author.
E-mail addresses: sxu@tamu.edu (S. Xu), keyser@cse.tamu.edu (J. Keyser).

0010-4485/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cad.2013.08.025

be propagated to the feature referring to it. Typically the propagation involves a series of computational geometry operations. However, because the tolerances are modeled in different spaces (world space and local part-centric space), the representation and the transformation of tolerances are not straightforward. For high-dimensional representations, the computation of tolerance propagation relies on computational techniques in high dimension that are notorious for the difficulty with implementation and robustness. Several lines of prior research have proposed methods addressing propagation.

Tolerance charting methods, which evaluate tolerance propagation based on engineers' experience, are a traditional way of handling tolerances. Computer-aided tolerance charting methods [9] have been developed to reduce the iteration of physical trial-and-error runs. Shortcomings of tolerance charting are that it cannot deal with complex spatial tolerance propagation issues or geometrical tolerances. To overcome this problem, methods for modeling tolerance propagation in higher dimensions have been presented [10–13]. One kind of method uses small displacements torsor (SDT) [14] to model the process planning [10]. Another kind of method, Technologically and Topologically Related Surfaces (TTRS), forms any part as a tree representing the succession of surface associations [11]. Tolerance information can be tracked along the stacking chain in the graph. The stacking up of parts could be simulated by a Monte Carlo method to estimate tolerance propagation [12,13]. This has the advantage of simplicity and flexibility, however the drawbacks of such methods are that it can be very time consuming and have poor computational accuracy at small to medium sample sizes. A third kind of method tries to find a feasible assembly plan on a graph structure by representing the related parts with the consideration of tolerances [15]. This method bridges the gap between its generalized tolerance model and previous models so that it could be incorporated with the previous assembly planning methods. Though this method formulates a general framework for tolerance estimation, its contribution to the optimal planning is limited.

Regardless of the representation of the zones themselves, the most common and straightforward implementations represent the tolerance zones independently. Unfortunately, as tolerances are propagated the dependencies between zones is then lost, causing over-estimation of the zones. One way of exploiting dependencies between the tolerance zones is by analyzing sensitivity to parametric variations [16,15]. Recently, a new method for describing dependencies of geometric uncertainties has been proposed [17,18], though to our knowledge it has not been implemented in any general way, including by the original authors, prior to our work presented here. This method, called LPGUM (Linear Parametric Geometric Uncertainty Model), uses a first-order approximation of the uncertainty zones of geometric primitives. The dependencies of uncertainties are derived from sensitivity matrices. Despite the promise of LPGUM and similar methods, they have to this point been shown to handle only very simple operations on very basic primitives, and thus have not been practical for modeling tolerance zones of real parts. Further, no tolerance propagation model has been designed, and thus tolerance zones cannot be cascaded. As a result, these methods were not (yet) suitable for tolerance analysis during the process planning phase.

1.2. Our work

In this paper, we present a new geometric model for tolerance modeling and propagation, geared towards the tolerance analysis in process planning. Our aim is to decompose a big chunk of analytic computations of conventional tolerance analysis into a series of geometry computations. We first decompose the part into basic geometric primitives, or features. Because those primitives are to be manufactured in a common part, we know they are related and that there are dependencies between them, which could be represented by a graph structure. We first decouple the primitives into several co-related primitive groups. In each group, the geometric position of a certain primitive (*target primitive*) is decided by the remaining primitives (*reference primitives*). This allows us to use the LPGUM model [17] to model the tolerance zone for the target primitive, because all its variations have been obtained. Next, we formulate a method for cascading the decoupled primitive groups, so that the tolerances can be transferred between groups. Using those cascading techniques, we could obtain the tolerance zones for all primitives by traversing the embedded graph structure representing the dependencies. The tolerance zones thus obtained provide a worst case estimation on dimensions, represented as a geometric polytope. Finally, we provide a computational optimization method which can improve the quality of the existing process plan so that the tolerance of the parts could be minimized as much as possible. The optimization problem itself is an NP hard problem. We propose an efficient approximated dynamic programming solver which utilizes the optimal substructure.

To summarize the contributions of this paper:

– we expand the LPGUM model to propose a complete representation of tolerances that is suitable for use in process planning and can exploit the dependencies within the dimensioning scheme.
– building on our tolerance model, we describe the tolerance analysis based on geometric computations that provide a worst-case estimation and a straightforward geometric representation of the tolerance zone.
– we describe a tolerance optimization approach that can improve an existing dimensioning scheme so that relative tolerances can be relaxed and thereby reduce manufacturing cost. The optimization result, which is a dependency graph of dimensions, indicates the processing order of features on the part.
– although the tolerance optimization problem is NP hard, we propose an efficient approximated solver with much lower complexity.

In order to simplify the tolerance analysis model, a group of basic geometric primitives, points, lines, and planes, are selected, and these are used to represent all features on the part, both linear and nonlinear [19]. This decomposition into basic primitives occurs before the tolerance analysis. Dimensions are used to build the dependencies among those primitives. Specifically, each dimension is associated with at least two primitives; one of them is the target primitive, $\mathbf{t}$, which is specified relative to the reference(s), denoted by a set $\{\mathbf{m}_i\}$, $1 \leq i \leq n$. This dependency between the references and the target could be represented by a map $\{\mathbf{m}_i\} \rightarrow \mathbf{t}$. The variation zone of $\mathbf{t}$ is caused by tolerances in the dimension(s) and in the propagated variation of the reference(s). We introduce two notations of tolerance zones of the target. One is the relative tolerance zone, denoted by $\mathcal{R}(\mathbf{t}, \{\mathbf{m}_i\})$ or $\mathcal{R}(\mathbf{t})$ for short, which shows the tolerance zone of $\mathbf{t}$ by assuming positions of references are exact (i.e. the tolerance zone is based solely on the tolerance in the dimensioning). The other is the global (real) tolerance zone, denoted by $\mathcal{Z}(\mathbf{t}, \{\mathbf{m}_i\})$ or $\mathcal{Z}(\mathbf{t})$, which shows the tolerance zone for $\mathbf{t}$ with consideration of all variations of dimensions, including that propagated from references.

This paper is organized as follows. Section 2 explains the computation of relative tolerance zones on primitives. Section 3 explains cascading of relative tolerance zones. Section 4 explains the optimization on an augmented dependency graph of dimensions. Section 5 shows the experimental results.

Note that, our work will be described and analyzed in only 2D. We can think of no fundamental reason why our methods would not extend to 3D (and some contributions are entirely dimension-independent), but we have not yet implemented all parts of our system in 3D, and thus will leave such extension to future work.