2D EXTENSION OF THE GAPSPACE MODEL FOR TOLERANCE ANALYSIS

Zhihua Zou & Edward Morse
Center for Precision Metrology, Department of Mechanical Engineering
The University of North Carolina at Charlotte, NC 28223, USA

Introduction

To insure interchangeable assembly and proper function of mechanical components, designers use tolerances to specify allowable limits for variations in geometric form and material properties. Effective use of tolerance analysis tools will allow the designer to balance acceptable variation in the part geometry with the requirement for production at lower cost. The GapSpace model described in the paper is an extension of the 1D model developed in [Morse 99]. Our 2-D extension bears some similarity to the Physical Constraints Faces Set method (PCFS) method from [Mullins 98] as both approaches use graph to represent the functional requirements of assembly. Because of the gap-based nature of the GapSpace model, the development of graph traversal rules and analysis methods is more direct.

The GapSpace model in 1D uses gaps to define the adjacency for a pair of opposing features, which are from different components and may potentially interfere when assembling. After identifying all gaps, an assembly graph is generated. Each assembly cycle in the graph represents a physical requirement for assembly, independently of how the components are dimensioned. The physical requirements are called Fits Conditions (FC) because they represent the necessary and sufficient conditions for the parts 'fitting' together in an assembly. Both worst case and statistical tolerance analysis can be implemented based on these FCs [Morse 99] [Zou 01].

In this paper we present the extension of these assembly cycles and Fits Conditions to assemblies of "two-dimensional" parts (i.e. prismatic parts with a constant polygonal cross-section).

The 2-D GapSpace Model

In this model, we only consider components that are polygonal, although they may be either convex or concave. This is because we use gap to describe a constant distance between two planar surfaces. We first introduce the ideas of Constraining Triangles and their relationship to the GapSpace Sine Law.

1. Constraining Triangle (CT)

If three faces in a polygonal component are constrained, the motion of the component may be restricted. We call the triangle generated by the three faces a constraining triangle, or CT. Because the constraint for each face is related to a gap between that face and the rest of the assembly, we can represent the Constraint Triangle in terms of these gaps. Equation (1) shows a validity test for a CT formed by three gaps. The directions of gaps g_1, g_2 and g_3 in Equation (1) are decided by the normals of the respective faces, pointing away from the material side of the part. The three gaps g_1, g_2 and g_3 in Figure 1 form a valid CT, called *CT_A*.

$$\sum_{i=1}^{3} a_i * \vec{g}_i = 0 \quad \text{where } a_i > 0,\ \overline{g}_i \text{ is gap associated with the face i} \qquad (1)$$

$$CTL = L2\sin(a1)\sin(a3) \qquad (2)$$

Each CT is associated with a value represented by the component's geometric parameters. This value is called the constraint triangle length (CTL). The CTL of CT_A in Figure 1 is shown in Equation (2). If the normal directions for two faces are opposite, their gaps generate a 1-D constraint triangle whose CTL is the distance between the faces. A CT is represented by its associated 3 (or 2, for the opposing case) gaps.

2. The GapSpace Sine Law

If a component is assembled with other components, the gaps (g_1, g_2, g_3) associated with any CT are related by the expression in Equation (3), where angles a1, a2 and a3 are those shown in Figure 1. The constant is the difference between the lengths of *CT_A* and the "outside" CT. The "outside" CT is formed when we treat other components as one pseudo-component (think of this as a subassembly). These outer faces form a CT which we call the *outside_pseudo_CT*.

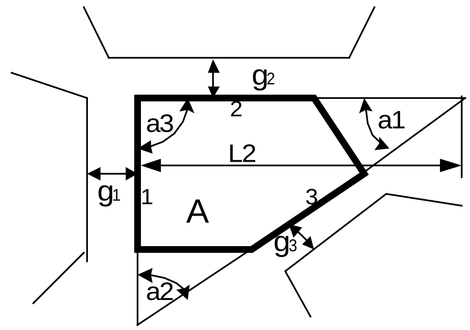

Fig 1. One part A is assembled with others

The equation (3) means that the value of left side keeps a constant wherever component A translates in any direction (no rotation) with respect to the rest of the assembly. As all coefficients are related to sine values, we call it the *GapSpace Sine Law*.

$$g_1 \sin(a1) + g_2 \sin(a2) + g_3 \sin(a3) = \textbf{\textit{constant}} = \mathrm{CTL(outside\_pseudo\_CT) - CTL(CT\_A)} \quad (3)$$

Combination of two components with a common gap

If CTs from different components share a common gap, the two CTs can be combined together by counting the common gap once and sum up all other gaps after making the coefficients for the common gap equivalent. In Figure 3, Part B has one CT (g_1, g_2, g_3) while C has one CT with (g_3, g_4, g_5). Using the Sine Law for both CTs as shown in Equation (4) we combine these two CTs as shown in Equation (5). The value of the left side of Equation (5) remains constant if either component B or C moves.

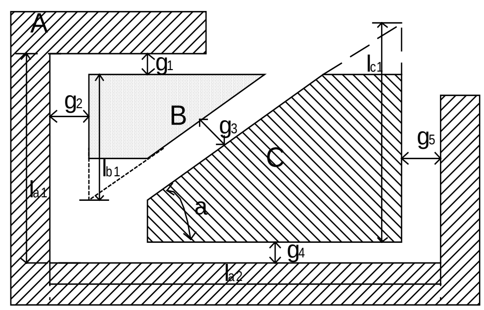

Fig 3. A real 2D assembly with 3 components

$$\begin{cases} g_1 \sin(\pi/2 - a) + g_2 \sin(a) + g_3 \sin(\pi/2) = c1 \\ g_3 \sin(\pi/2) + g_4 \sin(\pi/2 - a) + g_5 \sin(a) = c2 \end{cases} \quad (4)$$

$$\mathrm{g_1 cos(a) + g_2 sin(a) + g_3 + g_4 cos(a) + g_5 sin(a) = c1 + c2} \quad (5)$$

Liaison graph

The constraint triangle is used to describe the constraints on the dimensions of individual component. As we showed above, the constraint of two components with a common gap is also possible. The next step is to find a way to combine the constants for all components in an assembly. An assembly graph is used to capture the assembly relationships, and a method for inducing assembly-level constraints is presented.

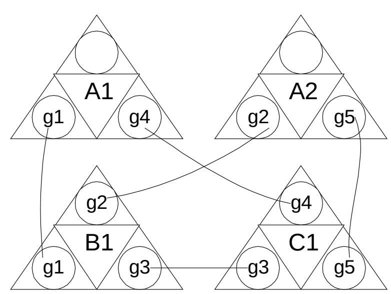

Fig 4. Assembly graph of Fig 3

In an assembly graph, each CT for a component will be one **node** of the assembly graph. Each node has 2 or 3 **ports** that reflect the gaps related to the CT. Any two ports belonging to different CTs in different components should be connected by an arc if they include the same gap. There is no arc between CTs from the same component. Each arc is associated with a single gap, but a gap may map many arcs in the graph. The assembly in Figure 3 has the assembly graph shown in Figure 4, where the CTs A1 and A2 are from Part A, B1 is from Part B, and C1 is from Part C. The CTs are connected by arcs representing the gaps in the assembly.

Fits Conditions (FCs)

After the assembly graph is constructed, the next step is to search the graph to find a constant expression (or expressions) for the assembly, called the Fits Condition (FC). A Fits Condition is a subgraph of the assembly graph with some restrictions. The sub-graph should satisfy the following conditions to be a FC:

1. The sub-graph is comprised of one or more assembly cycles (explained below).
2. If one port in any node is included in the sub-graph, any other ports in the node should be included in an assembly cycle that is included in the sub-graph.
3. The sub-graph can't be divided into more sub-graphs that satisfy conditions 1 and 2.

The third condition above is to prevent the search of the assembly graph from finding redundant FCs. The basic element in a FC, the assembly cycle, is a general cycle in the graph theory plus these constraints:

1. The assembly cycle must enter from one port of a node, exit from another port of the node
2. The assembly cycle can pass through at most one node in the same component.

If we follow the definitions for FCs and assembly cycles, we are able to find that the assembly graph in Figure 4 has just one Fits Condition, and 2 assembly cycles are included in this Fits Condition.

The properties of FCs

While a Fits Condition is comprised of assembly cycles, it is also comprised of Constraint Triangles that include gaps. A FC may be represented by the weighted sum of the included gaps if we repeatedly perform the combination of CTs as shown in Equations (4) and (5). The combination can be aborted when we've reduced the problem to two (possibly pseudo-) components, because these components or sub-assemblies will share common gaps. For example, in the Equation (5), after the CTs in B and C are combined, it is not necessary to combine the CTs from part A. The Equation (5) is already an FC after the CTs in parts B and C have been combined.

Some important properties of FCs follow:

1. An FC can be represented by weighted sum of included gaps. The value of FC is independent of the relative positions of components in the assembly.
2. If and only if all FCs for an assembly are non-negative, the components can be assembled without interference. Fits Conditions are so named because they guarantee fitting among components when the value is nonnegative.
3. The value of a FC is represented by the geometric parameters of related Constraint Triangles inside. Every Constraint Triangle has a contribution coefficient (CC) for the FC, representing the weight of

its Constraint Triangle Length in the FC.

Analysis Example

For the assembly in Figure 4, the only FC is shown in Equation (5). The expression of this FC in terms of the components' geometric parameters (dimensions) is shown in Equation (6). The contribution coefficients and CT Lengths are listed in Table 1 for all CTs.

Table 1. CCs and CTLs

CT	Contribution Coefficients	CT Length
A1	cos(a)	l_{a1}
A2	sin(a)	l_{a2}
B1	-1	$l_{b1}\sin(a)$
C1	-1	$l_{c1}\cos(a)$

$$FC = \cos(a)l_{a1} + \sin(a)l_{a2} - \cos(a)l_{b1} - \cos(a)l_{c1} \qquad (6)$$

Some basic tolerance analysis questions are listed below. The answers may be found easily, given that we've already developed Equations (5) and (6).

Table 2: The questions that are answered by the 2D GapSpace model

Question	Answer
Is there interference in the assembly?	Only if an FC has a negative value.
What is the maximal value of a gap?	The minimum of the FCs divided by its coefficient if the FCs contain the gap
What is the relative sensitivity of each gap with respect to part dimensions?	Expressed as a ratio of the coefficients of the gap.

Conclusions

We have presented the GapSpace modeling technique in 2-D, which is a direct extension of 1-D GapSpace. The representation of an assembly in 2 dimensions using an assembly graph is shown. One or more FCs are found for the assembly graph. These FCs are represented by the weighted sum of the gaps, where the weights are determined using the *GapSpace Sine Law*. Each FC can also be represented by the components' dimensions. The tolerance analysis is easy to execute after both representations of FCs are available. The algorithms to search FCs and combination of two CTs have already been implemented using C++ programming by the authors. The limitations of the model described in the paper are that the components must be polygons and no rotation is considered. We are extending the method to more complicated components and have begun work to apply it to 3 dimensional assembly.

References:

[Morse 99] Morse, E. P.; "Models, Representations, and Analyses of Toleranced one-dimensional Assemblies", Ph.D. Thesis, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, 1999.

[Mullins 98] Mullins, S. H.; Anderson, D. C.; "Automatic identification of geometric constraints in mechanical assemblies", *Computer-Aided Design*, vol. 30, no. 9, pp715-726, August, 1998.

[Zou 01] Zou, Z. H.; Morse, E. P.; "Statistical Tolerance Analysis Using GapSpace", 7th CIRP Seminar, ENS de Cachan, pp 313-322, France, April, 2001.