

Finding the Relative Motion Space of an Assembly Using the GapSpace Model

Zhihua Zou & Edward P. Morse

*Center for Precision Metrology, Department of Mechanical Engineering
The University of North Carolina at Charlotte, NC 28223, USA*

1. Introduction

When mechanical parts are manufactured, there are inevitable variations in their dimensions and form. Tolerances are used to define the allowable variability of these geometric attributes. When components are assembled, clearance conditions between the parts allow them to move with respect to one another. All possible relative positions for a pair of parts in an assembly can be captured by defining a *region* in the space of parameters that describe the relative position of the parts. In this paper we refer to this region as the relative motion (RM) region between parts.

As assemblies become more complex, such as 2D non-orthogonal assemblies, the relative motion region is not obvious and there is not systematic way to calculate it. In some applications, the range of relative motion between parts may be the primary design function. In robotics, the workspace of the end-effector to the base is calculated through the kinematic parameters. Tolerances inside the joints may affect the accuracy of the robot. In robotics and other applications, it is useful to have a way to calculate the relative motion region between mechanical parts, especially when considering tolerances.

A GapSpace model for multi-dimensional assembly has been proposed by the authors [1, 2]. The primary use of the model is for tolerance analysis of mechanical assemblies. In the model, gaps are used to describe the relationship between pairs of adjacent mating features on different parts. An assembly graph is generated to represent the spatial relationship of parts in the assembly. Sub-graphs of the assembly graph are identified which determine the independent and sufficient conditions for non-interfering assembly. These subgraphs represent inequalities which are called the fitting conditions (FCs) of the assembly.

These fitting conditions are also the independent units for calculating the relative motion region because each fitting condition may be regarded as a spatial constraint of the parts involved. When the parts A and B are involved in two different fitting conditions, their relative motion should be the intersection of the two relative motion regions. This paper describes the algorithms use to calculate the relative motion region based on the assumption that all fitting conditions ($FC_i, i=1, \dots, n$) are known and all gaps ($g_j, j=1, \dots, k$) defined are only constrained by these fitting conditions. These assemblies may be called floating assemblies [3] where the extent of the "float" is the motion we discuss in this paper. For this discussion, all parts are modeled as rigid bodies.

The remaining sections are organized as follows: we first discuss the relative motion algorithm for one dimensional assembly, which is easily understood and straightforward; we next address the algorithm for multidimensional assembly followed by a two-dimensional example. We conclude with a short discussion of open issues and future work.

2. Relative Motion Region in 1D assembly

To calculate the relative motion region, a coordinate system for each part is necessary; we treat each part as a rigid body as is typical in robotics. The frame of the part A, or F_A , is the coordinate system attached to that part. The relative motion region $RM(A, B)$ describes the possible positions of F_B with respect to F_A . The relative motion region in 1D assembly can be represented simply by the "length", or extent of the region, presuming we don't care where the region is. This length is called the maximal relative motion or $MRM(B,A)$. It is straightforward to show that $MRM(B,A) = MRM(A,B)$ in a 1D assembly.

When an assembly has only two parts A and B, $MRM(B,A)$ takes on the minimal value of the assembly's fitting conditions (FCs). The proof is given in [1]. The value of a Fitting Condition is the sum of the gaps that occur in that Fitting Condition. This value is determined by the dimensions and tolerances of the parts in the FC.

When an assembly includes more than two parts, the maximal relative motion between parts A and B is not only controlled by the fitting conditions directly including A and B, but also those fitting conditions that indirectly connect them. As indicated above, each fitting condition puts one spatial constraint on the parts it contains. The relative motion between any two parts is bounded by this FC value. If any two parts, say A and B, are involved with more than one FC, MRM between them must be bounded by the minimal value of these FCs, or

$$MRM(A, B) \leq \min_i FC_i \text{ where } A \in FC_i, B \in FC_i \quad (1)$$

Equation (1) is an inequality because MRM may be controlled by an intermediate part, say C. If the sum of $MRM(A, C)$ and $MRM(B, C)$ is smaller than the value calculated by Eq. (1), $MRM(A, B)$ should be bounded by the sum also. Figure 1 shows a situation where this may occur, and a solution is proposed in [1]. We now introduce an algorithm which finds the MRM for any two parts in an assembly graph – this is shown below in Algorithm outline 1.

To use the algorithm, we assume all fitting conditions' values are non-negative. This permits the use of the Floyd Algorithm for finding a minimal spanning tree, as it relies on this assumption [5]. The physical meaning of a negative FC is that an interference condition exists in the assembly. In this case the parts can't move, and their relative motion is nonexistent (for our rigid body assumptions).

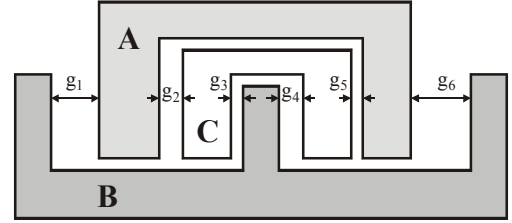


Figure 1: $MRM(A, B)$ depends on part C

Name: SearchMRM(G)
Input: assembly graph G
Output: $MRM(A, B)$; A, B are any different parts inside the assembly
Algorithm:
1. Find all FCs in G and calculate their values
2. For any two parts A and B in G, set the MRM to $+\infty$.
3. For any two parts in G, update MRM as the value of the right side in Equation (2)
4. Generate an undirected weighted graph G2 where
a. Each part is a node in G2
b. If MRM between any two parts is smaller than $+\infty$, add an arc between them and set the weight of the arc as the MRM.
5. Find the minimal spanning tree for A and B using Floyd's algorithm, the sum of the weights of the tree is the MRM between the parts.

Algorithm outline 1: Determining the MRM of any two parts in a 1-D assembly

3. Relative Motion Region in multidimensional (2D and 3D) assembly

Because the current GapSpace model can only handle assemblies with polygons or polyhedrons (either convex or concave) we only discuss the relative motion regions between these kinds of parts in this paper. We only consider the translation of the parts, so the relative motion region will have the same dimension as the space of the assembly. Because 2D and 3D assemblies have similar properties, all examples and proof will be done in 2D for easy visualization and explanation.

In 2D, the relative motion region between 2 parts is a subset of 2 dimensional spaces which may be convex or concave. When we talk about relative motion region between part A and part B (abbreviated as $RM(A, B)$), our meaning is that the possible trace generated by any point in part A after fixing part B without violating the non-interference criterion. If one frame is attached to each part, $RM(A, B)$ is the movable space of the origin of F_A after fixing F_B . We should note that $RM(A, B)$ is symmetric with $RM(B, A)$. $RM(A, B)$ could be found by rotating $RM(B, A)$ by 180 degrees around the origin of Frame A. For brevity, we won't belabor the difference in the two expressions, although they are distinct.

Let's look at a simple assembly in Figure 2 and check the relative motion regions. Part B and C are not connected directly. Without difficulty, $RM(B, A)$ and $RM(A, C)$ can be formed as shown in Figure 3. $RM(B, C)$ can be calculated from $RM(B, A)$ and $RM(A, C)$ by using Minkowski sum (or M-sum) [4]. The process matches the step 5 in calculating the sum of arcs in the algorithm SearchMRM(G) of 1D as M-sum is similar to a linear sum in 1D.

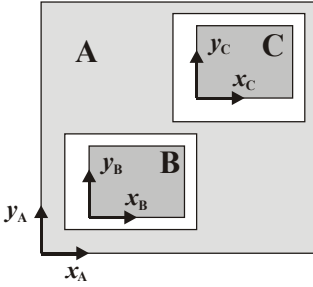


Figure 2: 3-part assembly

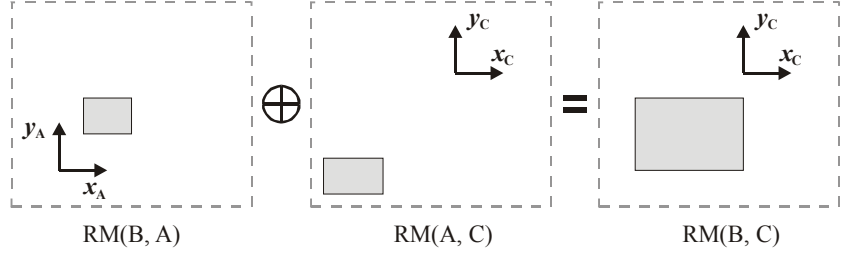


Figure 3: Minkowski sum of two RMs

Two questions come up: 1. Can we use the M-sum of $RM(B,C)$ and $RM(C,A)$ to get $RM(B,A)$? 2. If not, what is the strategy to choose the basic relative motion regions to get others when using the M-Sum? Three steps (a, b, and c below) are proposed to solve the problems.

- Find the relative motion region $RM(A,B)_{FC}$ of any two parts constrained by a single FC where both parts are located in the FC. This step matches the step 1 in SearchMRM(G) which is used to get the value of FC in 1D.
- Find the direct combined relative motion Region $RM(A,B)_{DC}$ for a pair of parts A and B. This tests all FCs that include both A and B. This step is similar to step 3 in SearchMRM(G) using Equation (2) to calculate MRM.
- Find the relative motion region $RM(A, B)$ for the whole assembly. The corresponding parts of SearchMRM(G) are steps 4 and 5.

3.1 The relative motion region $RM(A, B)_{FC}$ constrained by a single FC, where $A \in FC, B \in FC$.

The problem is very easy in one dimension as the relative motion is the value of the FC. This is not the case in 2D because the gaps are not parallel, and the maximal possible value for each gap in the FC may be different. There is no known geometric method for solving the problem, so we use the following method based on the GapSpace model.

Algorithm: RMinOneFC(FC, A)

Input: Fitting condition FC, component A; (gaps in FC, features information related gaps)

Output: Relative motion region $RM(X, A)$ in FC, X is any part in FC

- Generate a local coordinate system for each component. (The features of related gaps are known). Frame A is regarded as base coordinate system.
- Define variables for each Frame origin, where each is associated with 2 variables (x_i, y_i) . If there are m 2D parts in the FC, $2(m-1)$ variables are defined.
- Set any gap Y with its maximal value in the FC and other gaps with zero. Generate an equation relating these gaps. A total of $2m-2$ equations exist. Solving the $2m-2$ equations with $2m-2$ variables, one possible position of frames is generated.
- Set different gaps with maximal value to get the possible set of positions of frames (a total of $2m-1$ positions for each component).
- $RM(X, A)$ is the convex hull of the $2m-1$ locations of Frame X.

Algorithm outline 2: Algorithm to calculate the relative motion region in one FC

When the relative motion between A and B is considered, we assume both A and B function as 2D parts in the FC. Otherwise the problem becomes easier if either of them functions as a 1D part (in a 1D part, only opposing parallel planes interact with other features in the assembly). This is because the relative motion of the two parts need only be considered in that direction. As discussed earlier, the physical meaning of $RM(A, B)$ can be envisioned as the possible locations that the origin of F_A can have when F_B is fixed.

In each fitting condition, only these 2D parts are considered for easy explanation. The relative motion region of all parts relative to part A can be found using the algorithm in outline 2, considering the FCs only. In this circumstance, the number of gaps is $2m-1$ according to property 2 in [2]. We believe the relative motion region is convex – hence step 5 – but have yet to prove this. More research is needed to address the problem. According to the physical meaning of our use of the M-sum, Equation (2) is guaranteed if parts A, B and C are part of the FC.

$$RM(B, A)_{FC} \subseteq RM(B, C)_{FC} \oplus RM(C, A)_{FC} \quad (2)$$

If the right side of Equation (2) is generated with no additional constraints (no other FCs) between B and A, this corresponds to the equivalence of the left and right sides.

3.2 Direct combined relative motion Region $RM(A,B)_{DC}$ for all pairs of parts (part A and Part B) when considering all FCs that include the pair.

The relative motion regions for individual FCs are combined by applying Equation (3). To calculate the intersection of regions, these regions must be in unique coordinate system. As we discussed in Section 3.1, frame B is the base frame of $RM(A, B)$.

$$RM(A, B)_{DC} = \bigcap RM(A, B)_{FC_i}, \text{ where } (FC_i, i=1,..n), \text{ where } A \in FC_i, B \in FC_i. \quad (3)$$

If the frames are not changed for the different FCs, the intersection calculation is evaluated in frame B.

3.3 The relative motion $RM(A, B)$

As we saw in 1D, the relative motion between two parts may be indirectly constrained by FCs that don't contain both A and B. So it is also important to consider all indirect links which may affect the relative motion regions in a multidimensional assembly.

We introduce the method to calculate the relative motion region $RM(A_1, A_n)_{path}$ through paths in which there are a sequence of parts $\{A_2, A_3, \dots, A_{n-1}\}$. Suppose that only neighboring parts (A_i, A_{i+1}) share a single fitting condition, and no other parts in the sequence share this FC (later we will explain that the condition is not required). $RM(A_{n-1}, A_n)$ is independent from $RM(A_{n-2}, A_{n-1})$ since A_{n-2} and A_n are assumed to not share any FCs, and their M-sum represents $RM(A_{n-2}, A_n)$. Evaluating the entire sequence to A1 leads to the following:

$$RM(A_1, A_n)_{path} = RM(A_1, A_2)_{DC} \oplus RM(A_2, A_3)_{DC} \oplus \dots \oplus RM(A_{n-1}, A_n)_{DC} . \quad (4)$$

As the M-sum is associative and commutative, the order of the elements in right side doesn't matter. The result of Equation (4) will be in the Frame A_n .

The method to get the $RM(A,B)$ in the whole assembly can be described in the following way. First, an undirected weighted graph needs to be generated, which is called RMGraph. In the RMGraph, each part is a node and for any two parts sharing an FC, one arc is added between them. The weights of the arcs are the combined regions calculated from Equation (3). The next step is to find the all spanning trees between two nodes in RMGraph, say A and B. Using Equation (4), we can calculate the relative motion region of $RM(A,B)$ for each path. The last step is to intersect all relative motion regions of these paths. As we explained above, the result of Equation (4) is located in the last frame (B in our case), so all regions are in the same frame. The intersecting calculation is done in the common coordinate system. Thus step can be written

$$RM(A_1, A_n) = \bigcap_i RM(A_1, A_n)_{path(i)} . \quad (5)$$

When we combined the RM in Equation (4), we assumed that only neighboring parts share FC. But this is not necessary since the FC that include non-neighboring parts will generate another path, which may make the original path redundant in some cases. This result follows from Equations (6) and (2).

$$(A \oplus B) \cap (A \oplus C) = A \oplus B \quad \text{if } B \subseteq C \quad (6)$$

Let's look at the example RMGraph shown in Figure 4. Suppose parts A, B and C are in FC1 and C and D are contained in FC2. According to Equation (4), we have

$$RM(A, D)_{\{B,C\}} = RM(A, B)_{FC1} \oplus RM(B, C)_{FC1} \oplus RM(C, D)_{FC2} , \quad (7)$$

$$\text{and } RM(A, D)_{\{C\}} = RM(A, C)_{FC1} \oplus RM(C, D)_{FC2} . \quad (8)$$

Using Equation (5) to combine (7) and (8), we get Equation (9) where the path in Eq. (7) is redundant.

$$RM(A, D) = RM(A, D)_{\{B, C\}} \oplus RM(A, D)_{\{C\}} = RM(A, D)_{\{C\}} \quad (9)$$

The reason is that the shorter path ($A \rightarrow C$) is always considered from another tree. When the regions in Equation (7) and (8) are intersected, the smaller region will appear in the final result. The M-sum operation can be time-consuming, so the removal of redundant calculations in the method described above is desirable. It is helpful to use logical manipulation of the RM equations in Equation (5) – as we've done in Equation (9) – *before* calculating the M-sum.

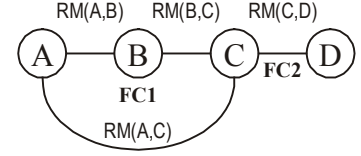


Figure 4: An RM graph

4. Example

Figure 5 shows an assembly with 4 parts. The assembly graph for this assembly is shown in Figure 6, along with the subgraphs that correspond to the FCs. We will not attempt to explain the assembly graph and the finding of the fits conditions in this paper. Figure 7 shows the RMGraph for this assembly.

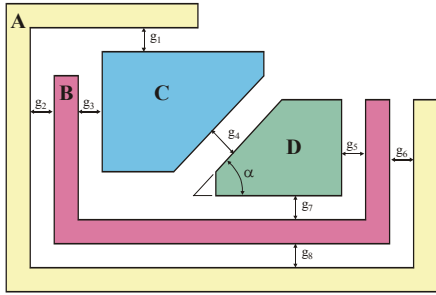


Figure 5: Four-part assembly.

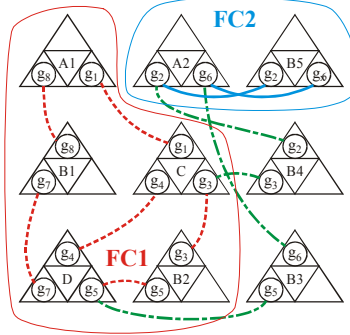


Figure 6: Assembly graph

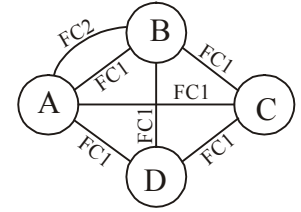


Figure 7: RM graph

We use logical manipulation to simplify the RM after finding all paths. From this manipulation we get:

$$\begin{aligned} RM(A, B) &= RM(A, B)_{FC1} \cap RM(A, B)_{FC2} \\ RM(A, C) &= RM(A, C)_{FC1} \cap \{RM(A, B) \oplus RM(B, C)_{FC1}\} \\ RM(A, D) &= RM(A, D)_{FC1} \cap \{RM(A, B) \oplus RM(B, D)_{FC1}\} \\ RM(B, D) &= RM(B, D)_{FC1} \cap \{RM(B, A) \oplus RM(A, D)_{FC1}\} \\ RM(C, B) &= RM(C, B)_{FC1} \cap \{RM(C, A)_{FC1} \otimes RM(A, B)\} \\ RM(C, D) &= RM(C, D)_{FC1} \cap \{RM(C, B)_{FC1} \oplus RM(B, A) \oplus RM(A, D)_{FC1}\} \end{aligned} \quad (10)$$

The region generated by the individual FCs are listed in Figure 8, where the frames A and B are coincident and frames C and D are in the parts' left-bottom corners. The dimensions are shown in approximately the same scale as the parts in the assembly of Figure 8. Figure 9 shows the relative motion regions between parts according to the formulas above.

5. Discussion and Conclusion

This paper addresses the relative motion regions of the components of a floating assembly based on the GapSpace model. The effect of tolerances on the motion is not discussed, although the values of fitting conditions depend on the dimensions *and* tolerances. The relative motion region for the toleranced case will be the nominal regions (as developed here) plus the range due to the tolerances – this may appear as an offset of the nominal regions. The difficulty is that the tolerances also affect the distance between the region and the fixed frame when the assembly becomes complex. One possible method is to use Monte Carlo method to sample the dimensions and tolerances, and

then calculate the relative motion range using the method in the paper. But this would be time-consuming and would not take advantage of the power of the GapSpace model. More research is needed for this issue.

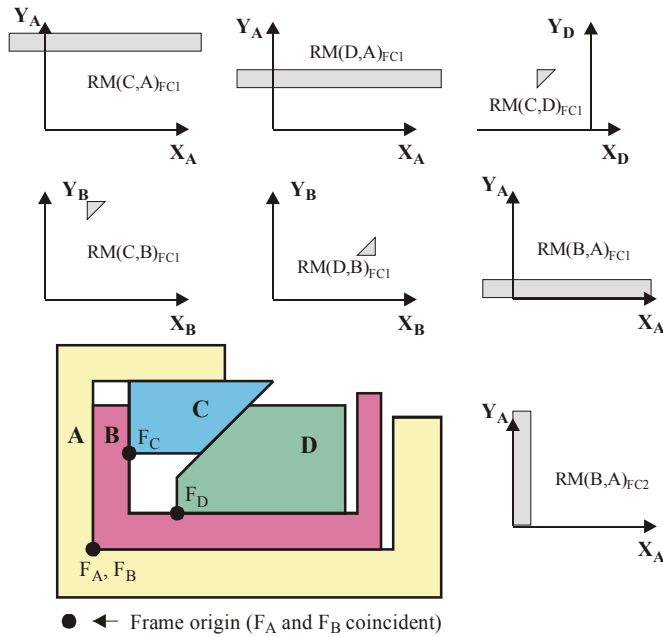


Figure 8: Relative motion regions from individual FCs

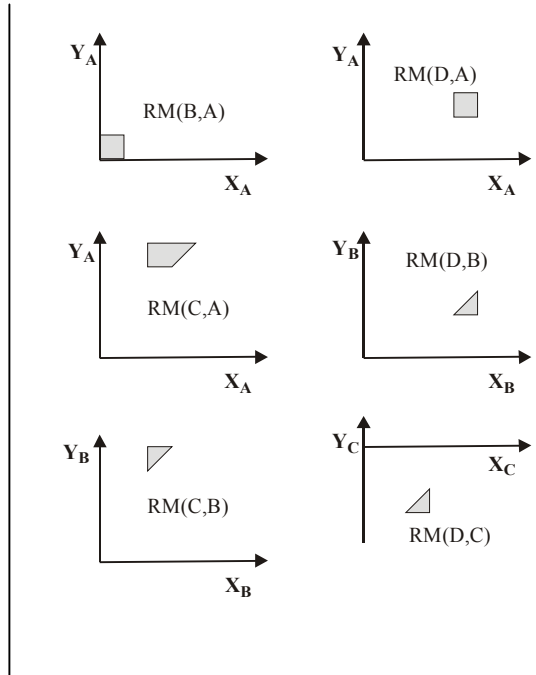


Figure 9: Relative motion regions

The technique described here may also be used to calculate the uncertainty of the end-effector of robots when each joint exhibits some uncertainty due to the kinematics of the design. As each joint is similar to a fitting condition in our discussion, the uncertainty of sequencing arm links will be calculated by the M-Sum. If there are parallel links, their uncertainties region can be calculated by intersection. This is the reason why robots with multiple sequential links have large uncertainties, although the workspace is large, and parallel machines (robots) have high precision with small workspace. Although only translations are considered in the paper, the potential rotation of parts can be approximated by linearizing the effects of the rotations. This requires additional work.

This paper proposes a method to calculate the relative motion region when the parts are assembled. With the help of the GapSpace model, the fitting conditions are captured and their values are computed, based on the geometric information of parts. The relative motion region is generated by M-summing the sequencing fitting conditions and intersecting parallel fitting conditions after the relative motion region in one fitting condition is found. This result may be used for reverse engineering where a designer may need to ensure that the possible motion between parts is satisfied. An additional application is robotics, as the accuracy of the end-effector is one key parameter of a robot.

6. References:

- [1] 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.
- [2] Zou, Z.Z.; Morse, E. P.; "Assembleability Analysis Using GapSpace Model for 2D Mechanical Assembly", *Proceedings of DETC*, DFM-34187, Montreal, CA, 2002.
- [3] Robinson, D. M., "Geometric Tolerancing for Assembly", Ph.D. Thesis, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, 1998.
- [4] Kaul, A; "Minkowski sums: A Simulation Tool For CAD/CAM", *Computers in Engineering*, Volume 1, ASME 1992.
- [5] Sedgewich, R., *Algorithms in C++*, IDSN 0-201-51059-6, Addison-Wesley Publishing Company, Inc., 1992.