A sequential constraint solver to simulate assembling operations for tolerance analysis

Pasquale Franciosa\textsuperscript{a*}, Salvatore Gerbino\textsuperscript{a} and Stanislao Patalano\textsuperscript{b}

\textsuperscript{a}University of Molise - School of Engineering, Via Duca degli Abruzzi - 86039 Termoli (CB) - Italy
\textsuperscript{b}University of Naples, Federico II - School of Engineering, P.le Tecchio 80 - 80125 Naples - Italy

Abstract

In the variational modeling of assemblies it is important to define the location of a part both in absolute terms and with respect to the position/orientation of other assembled parts. The present paper proposes a programming optimization approach to solve this problem. The algorithm, by using the heuristic Nelder-Mead technique - combined with a penalty function - simulates and solves sequential assembly strategies to find the optimal geometric configuration of a rigid part with variational features satisfying all the assembly constraints in the given sequence. The algorithm best aligns mating features avoiding, at the same time, feature-to-feature interferences, and automatically calculating the amount of movement the part being assembled must obey to satisfy assembly constraints, at that state of the assembly process. Thus, different assembly sequences can be simulated also including variational features.

Keywords: sequential constraint solver; rigid assemblies; assembly simulation; constrained optimization; tolerance analysis

1. Introduction

Most manufactured products are assemblies made of tens of individual parts, thus assembly design is a crucial task to be accomplished when designing or re-designing a product. It is also well-known that assembly design has significant impact on many downstream activities such as process planning, production planning and control, and packaging [1]. These activities are often strictly related to the way to assemble the product's components. Some products are assembled in a simultaneous way, by positioning all parts together at the same time, but in many other cases it is necessary to assemble parts one-by-one, in a sequential way. In both cases, when doing tolerance analysis, the assembly sequence strongly influences the final assembly geometric configuration, because variation at level of the single part features propagates through the assembly [2, 3, 4]. Tolerance stack-up is strictly related to the specific constraint status among part features involved in the assembly.

"Constraint solving" topic is covered in different engineering fields, from Dynamics and Kinematics applications to assembly sequence analysis through functional analysis [5, 6 and 7]. For example, in the Kinematics field, closed loop mechanisms are solved by using the well-known Newton-Raphson (N-R) method, which calculates the roots of a set of non-linear equations in a simultaneous way. However, this method has several drawbacks. First of all, the Jacobian matrix, involved in the calculation, needs to be calculated at each iteration: this task may become computationally very huge when the number of variables increases. Moreover, N-R method is not robust when handling
over-constrained assemblies, which are quite common in industrial applications. In this case, the initial guess required in N-R method plays a relevant role. For under-constrained assemblies N-R technique may be unstable due to lack of geometric constraint: Jacobian matrix becomes rectangular and the calculation of its pseudo-inverse is required [5].

Looking at the solid modeling community [8, 9 and 10 to cite a few], graph-based approaches and algebraic methods are the most commonly used to solve geometric constraint problems, and are dominant in 2D CAD applications. They have been also extended, more recently, to 3D cases where handling constraints and finding solutions is more complex. From 2D CAD point of view, the algebraic approach by D-Cubed, the so-called Dimensional Constraint Manager, DCM, is de facto an industrial standard in constraint-based sketching. The more recent 3D version of this software, 3D DCM, based on a fast non sequential solver, is used to constraint parts in assemblies and mechanisms. Similar solution is offered by Ledas Geometric solver, LGS 3D, a variational geometry engine used by several CAx systems. Working with CAD geometries, also Screw Theory is used to calculate the constraint status of an assembly based on the choice of kinematic joints used to assembly parts [11].

When redundant constraints are introduced, the related assembly equations may become dependent to each other. This is a very crucial issue to be faced out. In fact, with an ideal rigid-part assembly there is no guarantee that all constraint relationships are properly satisfied. Therefore, under the hypothesis of ideal rigid-part assembly, it is important to calculate whenever a given assembly, for a given set of constraint relationships, is feasible or not. The answer to this question is not trivial at all if we consider also variations of assembly features. In fact, as shown in [12], variational constraint features need a search contact algorithm in order to best align them, avoiding at the same time, feature-to-feature interferences. One of the first contribute to the assembly modeling among variational features was offered by [13]. The author adopted a mathematical programming approach to model constraint relationships. The general idea may be stated as follows: given an "object" part being positioned with respect to a set of "target" parts, the constraint features should be aligned as closely as possible and interferences should be avoided. Turner gave a solution to this issue, but he limited his research only to 2D mating features under the small displacement hypothesis.

Chase and his group at Brigham Young University posed the basis for a more general approach, named Direct Linearization Method (DLM), to simulate variational assemblies [14]. The whole assembly is modeled with a graph representation, in which edges correspond to joining features, whereas vertices are parts being assembled. Then, equations are written for each independent loop. Assembly constraints for each vector loop may be expressed as a concatenation of homogeneous rigid body transformation matrices, which results in a set of non-linear equations. These equations are linearized by using Taylor’s series expansion. DLM procedure allows to solve into a closed form any mechanical assembly for a given set of tolerances: no Monte Carlo simulation is strictly required. However, 3D tolerance zones are not fully-integrated. In addition, it does not allow to simulate different assembly sequences as assembly constraints among mating features are modeled through "linearized equivalent" joints, not allowing to model non-linear constraint conditions (see contact constraints).

A contact search algorithm was proposed in [15] to simulate 2D and 3D assembly operations accounting shape errors, modeled by natural mode shapes. In [16] a framework for a constrained optimization method was described to simultaneously solve geometric 2D constraints. Authors stated that their approach gives stable results both for under- and over-constrained problems. More recently, in [12], authors proposed a more general approach, also working for 3D feature constraints, accounting the best alignment among plane-to-plane and cylinder-to-cylinder features. Authors proposed a sequential solver to best align mating features of an assembly made of only two parts. How to extend the proposed procedure to more complex assembly and other assembly features (see, for example, plane-to-cylinder) was not covered. Moreover, how to avoid feature-to-feature interference when working with cylinder-to-cylinder alignment was also omitted.

Starting from the preliminary results of Franciosa et al. [12], the present paper describes a new assembly constraint solver, able to simulate sequential assembly strategies, under the hypothesis of ideal rigid parts, and to model both "mate" (assuring the best alignment geometric condition) and "contact" (avoiding feature-to-feature interference) feature constraints. The proposed algorithm successfully works with nominal and variational features. In the latter case, both small and large displacement hypotheses are supported.

The paper is arranged as follows: Section 2 depicts the general methodology; Section 3 presents the assembly constraint modeling approach; Section 4 reports case studies, and, finally, Section 5 draws discussions and conclusions.

2. Methodology Overview

During assembly operations, an "object" part must be moved to satisfy a set of geometric constraints between
its features and the mating features of the "target" parts. For a given assembly sequence, when introducing a new assembly constraint, two criteria must be obeyed at the same time: (I) calculate the best alignment between the actual mating features, and (II) keep all constraints already met for next alignments.

The rigid-motion of the object part is parameterized by a 4x4 assembly homogenous matrix, depending on the six degrees of freedom (three translations and three rotations), which are aimed to be determined.

A programming optimization approach is here used to find the optimal geometric configuration of the "object" part satisfying all the assembly constraints in the given sequence. Mate and contact geometric conditions are treated as equalities and inequalities, respectively.

The optimization problem is solved through the heuristic Nelder-Mead technique, combined with a penalty function. Mating features are best aligned avoiding, at the same time, feature-to-feature interferences, and automatically calculating the amount of movement the object part must obey to satisfy assembly constraints, at that state of the assembly process.

The proposed procedure may be successfully applied to solve both nominal and variational assemblies (variational features were already treated in [12]). In the latter case the effects of the 3D tolerance stack-up, depending on the assembly sequence, can be calculated. The sequential solver algorithm was embedded in SVA-TOL software, which has been developed at University of Molise - Italy, in cooperation with University of Naples - Italy, to do tolerance analysis of rigid-part assemblies.

3. Assembly Constraint Modeling

3.1. Materials and Methods

During assembly operations an Object Part (OP) must be moved to satisfy constraints of Target Parts (TP), which are assumed locked (no motion allowed).

We use Degrees of Freedom (DoFs) - three translations and three rotations - to model and parameterize assembly constraints [17]. Thus, the directions of constraint of any kinematic joint are only related to those DoFs along/around which motions are not allowed. The remaining DoFs are invariant. For example, for a plane-to-plane constraint, with z normal, rotation around z axis and translations along x-y axes are invariant. The remaining DoFs correspond to the directions of constraint.

In the present paper, we distinguish between "mate" and "contact" geometric conditions, holding between planar and cylindrical features.

"Mate" condition requires that two assembly features come in contact (at least one contact-point) and keep that geometric configuration with respect to invariant DoFs. When the mate condition is met, assembly features cannot detach and they keep the same relative orientation during the next assembly operations. Thus, for example, plane-to-plane mate condition assures that the plane features involved in the mate cannot move far away to each other, neither rotate out the plane. "Contact" condition, instead, only assures that two assembly features do not penetrate to each other. This means that the two features may detach, losing their relative geometrical configuration, in the next steps of the assembly process.

In real industrial applications contact constraints (often called NC-blocks) are used to limit rigid-motion displacements which may arise during the positioning of parts on the fixture frames. For example, looking at Fig. 1a, a two-part assembly is showed related to an intermediate assembly state with OP in the mate condition "matei-1" (plane-to-plane) and contact condition "contactk" (plane-to-cylinder). After specifying "matei" condition (Fig. 1b), while keeping the previous
mate, the lateral contact can be lost. Notice that contact constraints make the solution process non-linear since the number of contact points is a priori unknown and, then, for that specific status of the assembly, they need to be recalculated step-by-step.

In the present paper we propose a sequential solver, which allows to solve constraint conditions one-by-one in an iterative way. When solving the "i-th" assembly constraint (mate or contact), the OP must be moved accounting all constraints already met. Therefore, for a given assembly sequence, the motion of OP is captured by the 4x4 assembly transformation matrix, $T_{TP,OP}$, defined as in equation (1), where $R_{TP,OP}$ and $d_{TP,OP}$ are the 3x3 rotational matrix and the 3x1 position vector, respectively. Looking at Fig. 1, $T_{TP,OP}$ expresses the location of the coordinate frame, $OP$, with respect to that one belonging to TP, $\Omega_{TP}$ (see equation 1).

$$
T_{TP,OP}(\alpha, \beta, \gamma, \Delta x, \Delta y, \Delta z) = \begin{bmatrix} R_{TP,OP} & d_{TP,OP} \\ 0 & 1 \end{bmatrix}
$$

(1)

The assembly matrix depends on the six DoFs (the triplets $[\alpha, \beta, \gamma]$ and $[\Delta x, \Delta y, \Delta z]$ define the rotational and translational DoFs, respectively) initially unknown. Thus, the rotational matrix and the position vector can be parameterized as in equation (2):

$$
\begin{align*}
R_{TP,OP}(\alpha, \beta, \gamma) &= R_x \cdot R_y \cdot R_z \\
(d_{TP,OP}(\Delta x, \Delta y, \Delta z) &= [\Delta x, \Delta y, \Delta z]^T
\end{align*}
$$

(2)

where $R_x$, $R_y$, and $R_z$ are the rotational matrices around x, y and z axes of the coordinate frame $\Omega_{TP}$. Under the small displacement hypothesis, Franciosa et al. [12] proposed a linearized expression of relationship (2). However, this assumption is acceptable only when parts are very close each-others before assembling them. Therefore, in the present paper, in order to not loose generality we did not linearize the rotational matrices. When the assembly transformation matrix is calculated based on the optimization approach (see Sections 3.2 and 3.3), OP, that is originally defined with respect to the assembly coordinate frame, $\Omega_{0}$, is repositioned by applying the 4x4 homogeneous matrix, $T_{0,0}$, stated in equation (3).

$$
T_{0,0} = T_{0,TP} \cdot T_{TP,OP} \cdot T_{TP,0}^{-1}
$$

(3)

That is, OP is firstly expressed in the target coordinate frame; then, once the assembly constraint is solved, OP is moved, accordingly; finally, it is transformed back in the assembly coordinate frame. As stated above, the present paper focuses on mate and contact geometric constraints between planar and cylindrical features. The following constraints (each of them can be seen either as a mate or a contact condition) may arise (see Fig. 2): (P-P) Plane-to-Plane, (P-C) Plane-to-Cylinder (or cylinder-to-plane), and (C-C) Cylinder-to-Cylinder.

3.2. Single-Constraint Modeling

Planar and cylindrical features are parameterized by a unit vector and a point. Looking at Fig. 2, we want to best align Object Feature (OF), defined by the normal vector $N_{OF}$ and the point $P_{OF}$, with respect to Target Feature (TF), defined by the normal vector $N_{TF}$ and the point $P_{TF}$.

Since TP is always assumed locked during the assembly operation, only OF is iteratively updated by the assembly matrix, $T_{TP,OP}$, as stated in relationship (4).

$$
\begin{align*}
N_{OF} &= R_{TP,OP}(\alpha, \beta, \gamma) \cdot N_{OF} \\
P_{OF} &= R_{TP,OP}(\alpha, \beta, \gamma) \cdot P_{OF} + d_{TP,OP}(\Delta x, \Delta y, \Delta z)
\end{align*}
$$

(4)

The optimization problem for mate condition corresponds to find-out the minimum of the scalar function, "J", here called alignment function and defined as in equation (5).

$$
\min_{\alpha, \beta, \gamma} \left\{ J(\alpha, \beta, \gamma) \right\} = \begin{cases} \\
\|N_{TF} + N_{OF}\| \text{ for } P-P \text{ constraint} \\
\|N_{OF} \cdot N_{TF}\| \text{ for } P-C \text{ constraint} \\
1 - \|N_{OF} \cdot N_{TF}\| \text{ for } C-C \text{ constraint}
\end{cases}
$$

(5)
For example, looking at plane-to-plane (P-P) constraint, equation (5) states that the relative angle is minimum when the norm of the resultant vector between NTF and NOF becomes minimum: J function drives the relative orientation between mating features.

Calculating the minimum distance between two features is a well-known topic covered in the computational contact community from which we have inherited the "mapping distance" operator [18], "Md", which gives the relative minimum distance between OF and TF, taking into account the boundary of the same features. For example, looking at Fig. 2a, the mapping distance, from the object to the target plane, can be calculated only for those points of OF whose projection lies inside the boundary of TF (shaded area in Fig. 2a). The unit vector "Nn" is here used to calculate the sign of the mapping distance operator. Thus, for a cylindrical feature, assumed as "pin", Md operator is positive if the mapping point lies "outside" TF, with respect to Nn unit vector, and becomes negative for any mapping point "inside". The interested reader may refer to [18] for more mathematical details.

Mate condition requires that at least one point of the object plane belongs to the target one. This means that the mapping distance operator is zero. On the other hand, in the case of contact condition, one should avoid that object and target features penetrate each other ("no penetration"). That is the distance of the point closest to the target feature must be equal to zero (when features keep contact) or greater than zero (when features detach). Often, it is of interest calculating the minimum distance just assuring no penetration among assembly features (no matter about best alignment among assembly features). In this case, the optimization problem can be formulated imposing Md operator being minimized with Md ≥ 0 ("minimum distance").

\[
\begin{align*}
\text{mate} & \rightarrow \left\{ \min_{\alpha, \beta, \gamma} \langle J(\alpha, \beta, \gamma) \rangle \right\} \\
M_d(\alpha, \beta, \gamma, Ax, Ay, Az) & \geq 0
\end{align*}
\]  

\[(6a)\]

\[
\text{contact (no penetration)} : \\
M_d(\alpha, \beta, \gamma, Ax, Ay, Az) \geq 0
\]

\[
\text{(6b)}
\]

\[
\text{contact (minimum distance)} : \\
\left\{ \min_{\alpha, \beta, \gamma, Ax, Ay, Az} \langle M_d(\alpha, \beta, \gamma, Ax, Ay, Az) \rangle \right\} \\
M_d(\alpha, \beta, \gamma, Ax, Ay, Az) \geq 0
\]

\[
\text{(6c)}
\]

3.3. Sequential Constraint Modeling

The sequential constraint algorithm iteratively solves assembly constraints. Fig. 3 shows the main steps of the proposed algorithm. For every assembly constraint ("Njoint" is the total number of assembly constraints) the constrained optimization problem is built-up. With respect to the "i-th" assembly constraint, "Ji" function is aimed to be minimized. Here, constraint functions (that is, "h1", "h2", "g1", "g2" and "g3") assure that both actual constraint ("i-th") and previous constraints ("k-th") are properly met. When working with contact conditions, "g1" and "g3" constraint functions avoid any feature-to-feature penetration (see equations 6b-c) during all the assembly process. Instead, the constraint function g2 is introduced to keep the same relative orientation of two mating features. Here, "t" is the alignment function already calculated for the mate constraint k-th. Thus, the "optimal" alignment function "Ji" generated at that state of the assembly process is stored in "t", and it will be adopted in the next assembly steps to keep the state of the relative orientation between those mating features. Finally, the position and orientation of the OP is updated according to equation (3).

The scientific literature offers several numerical algorithms to solve the optimization problem stated in step (2) of Fig. 3. Penalty method and Lagrange multipliers are two well-known techniques adopted to handle constrained optimization problems [19].

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Get object and target features related to constraint i-th</td>
</tr>
<tr>
<td>2</td>
<td>Build-up the constrained optimization problem</td>
</tr>
<tr>
<td>3</td>
<td>Solve the constrained optimization problem</td>
</tr>
<tr>
<td>4</td>
<td>Update the position and the orientation of the object part</td>
</tr>
</tbody>
</table>

\[
T_{op} = T_{TP} = T_{TP} \left[ \begin{array}{c}
\Delta x_{op}, \Delta y_{op}, \Delta z_{op}
\end{array} \right] \bigg|_{\text{contacts}}
\]

Fig. 3. Sequential constraint solver algorithm
The first one is less accurate but more stable than the second one. On the other hand, the penalty method does not require the calculation of the Jacobian matrix (or partial derivatives), which is usually a computationally huge task. That is why, in the present paper, we adopted the penalty method, despite it is generally less accurate than Lagrange multiplier method. In this way, constraint satisfaction is monitored by the penalty function being small enough.

When working with penalty method, the constrained optimization problem (2) can be reformulated as an unconstrained problem, which is here treated with the Nelder-Mead method [20]. This method, that gives the right balance between numerical stability and computational time, was adopted in the present research as we were looking for a numerical procedure as stable and fast as possible to solve geometric constraints, also accounting variational features. However, other optimization routines could be implemented.

![Fig. 4. Case study: three-mate constraints](image)

**4. Case Studies**

The proposed sequential constraint solver was embedded in SVA-TOL software, to do tolerance analysis of rigid-part assemblies. SVA-TOL is fully written in Microsoft VB® programming language. Planar and cylindrical features are directly imported from SolidWorks® (by Dassault Systemes) CAD system, once picking them from the SolidWorks® graphical area.

**4.1. Two-part Assembly**

Fig. 4 shows a two-part assembly. Three mate constraints are defined. The aim of this study is to show how the proposed sequential solver allows to simulate different assembly sequences.

<table>
<thead>
<tr>
<th>Assembly Sequence ID</th>
<th>Constraint Sequence</th>
<th>(d_1) (mm)</th>
<th>(d_2) (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>mate_1 + mate_2 + mate_3</td>
<td>0.984</td>
<td>0.148</td>
</tr>
<tr>
<td>II</td>
<td>mate_1 + mate_3 + mate_2</td>
<td>0.736</td>
<td>0.670</td>
</tr>
<tr>
<td>III</td>
<td>mate_2 + mate_1 + mate_3</td>
<td>1.010</td>
<td>0.000</td>
</tr>
<tr>
<td>IV</td>
<td>mate_2 + mate_3 + mate_1</td>
<td>0.404</td>
<td>0.000</td>
</tr>
<tr>
<td>V</td>
<td>mate_3 + mate_1 + mate_2</td>
<td>0.000</td>
<td>0.593</td>
</tr>
<tr>
<td>VI</td>
<td>mate_3 + mate_2 + mate_1</td>
<td>0.000</td>
<td>0.604</td>
</tr>
</tbody>
</table>

Based on the variational-feature approach already proposed in [12], only a random configuration was generated (tolerances were modeled with a statistical normal distribution - natural tolerance range = 6).

![Fig. 5. Assembly geometry for different assembly sequences - variation scale factor = 100](image)
Mating features of the OP are supposed ideal (no input variation is assigned). The so-generated variational geometry was adopted for all assembly sequences. Distances "d1" (from point "P1", belonging to OP, to target feature "TF3") and "d2" (from point "P2", belonging to OP, to target feature "TF2") were monitored (see third and fourth columns in Table 1). Fig. 5 depicts the final assembly geometry for all six assembly sequences (only mating features are drawn for TP).

As expected, the final assembly configurations are strongly different to each other. As example, looking at "assembly sequence V", mate1 is a plane-to-plane type, while mate2 and mate3 become line-to-plane (two-contact points) and point-to-plane (one-contact point) types, respectively. Moreover, no penetration is assured at mating feature interfaces. Table 1 shows the six feasible assembly sequences.

4.2. Three-part Assembly

Figs. 6 and 7 show a three-part assembly. The aim is to analyze the minimum distance "Df" and the angle "Af", between the pin of the part C and the hole of the part A.

Mate conditions were established between part A and part B ("mate1", "mate2", and "mate3"). The pin/hole joint was modeled through a contact constraint ("contact") - no penetration allowed. Moreover, a "minimum distance" contact constraint ("contact2") was also defined to assure part C and part B were close as much as possible to each other. Tolerances were defined for each part (see Fig. 6).

Each tolerance was modeled with a statistical normal distribution (natural tolerance range = 6). Monte Carlo method was used to generate random variational features (number of simulation = 1000). The following assembly sequence was assigned: mate1 + mate2 + mate3 + mate4 + contact1 + contact2. Histograms of frequencies are reported in Fig. 8. Among 1000 assembly configurations only 908 are feasible, for which all assembly constraints are properly met (that is, the penalty function is small enough). Unfeasible assemblies can be solved only by accounting part deformation.

As expected, the minimum value of the functional requirement Df (Fig. 8b) is right zero, since assembly features cannot penetrate each-other. Moreover, the
maximum value is about 0.7 mm, which is lower than the maximum radial gap (1.3 mm) between pin and hole features.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig8}
\caption{(a) functional requirement $A_c$; (b) functional requirement $D_r$}
\end{figure}

5. Conclusions and Final Remarks

The paper described an assembly constraint solver able to simulate sequential assemblies, under the hypothesis of ideal rigid parts, and to model both mate and contact constraints. By using a programming optimization approach, the motion of the object part, with respect to target ones, was calculated by solving a non-linear constrained optimization problem. Equalities and inequalities were introduced to model mate and contact constraints, respectively. The motion of the object part was parameterized through a 4x4 homogeneous transformation matrix. The Nelder-Mead algorithm, combined with a penalty function, was adopted to solve the optimization problem. Other optimization algorithms could be also integrated to further improving calculation performances.

Two case studies were analyzed. The first one showed how the proposed constraint solver allows to simulate different assembly sequences. Then, a three-part assembly was studied to calculate functional requirements between assembly features.

At the present, the proposed methodology accounts planar and cylindrical features, which may vary within the assigned tolerance ranges. More assembly features are going to be included in the analysis. Future works will be devoted to automatically calculate simultaneous assembly strategies and over-constrained assemblies, also considering part deformation.

References

