Polyhedral-based Modelling and Algorithms for Tolerancing Analysis

Vincent Delos, Denis Teissandier, Alexander Malyshev, Grégory Nuel, Sonia García

Polyhedral-based modelling and algorithms for tolerancing analysis

Vincent Delos[a], Denis Teissandier[b], Alexander Malyshev[c], Grégory Nuel[d], Sonia C. García[b,*]

[a] *CNRS, I2M, UMR 5295, Talence, France*
[b] *Univ. Bordeaux, I2M, UMR 5295, Talence, France*
[c] *Open Cascade, Inc., Nizhny Novgorod, Russia*
[d] *CNRS, LPSM, UMR 8001, Paris, France*

Abstract

The cumulative stack-up of geometric variations in mechanical systems can be modelled summing and intersecting sets of constraints. These constraints derive from tolerance zones or from contact restrictions between parts. The advantage of this approach is its robustness for treating any kind of mechanisms, including the over-constrained ones. However, the sum of constraints, which must be computed when simulating the accumulation of defects in serial joints, is a very time-consuming operation. In previous papers, we proposed to virtually limit the degrees of freedom of the toleranced features and joints turning the polyhedra into polytopes to avoid manipulating unbounded objects. Even though this approach enables to process the whole mechanism, it also introduces bounding or cap facets which increase the complexity of the operand sets after each operation until becoming far too significant. In this work, we introduce algorithms summing, intersecting and testing inclusions. As they operate on sets of constraints using unbounded polyhedral objects, we identify the smaller sub-space in which the projection of these operands are bounded sets. Calculating the sum in this sub-space allows reducing the operands complexity significantly and consequently the computational time. Then, checking the final inclusion informs us not only about the compliance of the mechanism tol-

*Corresponding author
Email address: sonia.garcia-gomez@u-bordeaux.fr (Sonia C. García)

erances with respect to the functional specification but also to quantify how far we are from this target. Finally prismatic polyhedra integrate ISO and contacts specifications in a very natural way and are able to perform a full kinematic analysis of the mechanism. After presenting the geometric properties on which this approach rely, we demonstrate it on an industrial case. Then we compare the computation times, prove the robustness of the new method and show how to quantify the functional condition compliance with respect to a given set of tolerances.
Keywords: Tolerance analysis, Model reduction, DOF, Polyhedra, Minkowski sum, Intersection

---

1. Introduction

Geometric uncertainties influence products assembly, functioning, and aesthetics. The consideration of these uncertainties in mechanical design is known as geometric tolerance analysis. In general, the choice of a tolerancing scheme is not trivial, and even more, it impacts all the stages of the product life cycle; hence a highly active research in the tolerance management is done.

Different methods for tolerance analysis have been developed. Most of them are based on the discretization of features into several points and solve geometric constraints with these points [1].Among these methods we can find CLIC [2], Robustness Analysis [3], Jacobian Matrices [4], TTRS [5],[6], SDT [7].

The main drawback of these models is that they consider each movement limit to be independent [8]. This implies that more than one simulation can be required to validate the fulfillment of a design requirement. Furthermore, over-constrained mechanisms cannot be treated with the aforementioned models.

Methods based on operations on sets of geometric constraints can face these issues. These methods model the movement constraints imposed by the tolerance zone of each toleranced feature in an abstract deviation space. The

combination of these sets, through Minkowski sums and intersections, allows simulating the deviations propagation in an assembly. Some models manipulate the sets of constraints through their frontiers [9, 10, 11]. The method based on polytopes discretizes the non-linear features to obtain linear constraints and manipulate only linear objects. Although the other models are initially able to handle quadratic constraints, they finally linearize the sets of boundaries because of the complexity of summing convex non-linear constraints. The effects of this linearization are discussed in [12]. The method proposed in [13] and [14], instead of dealing with the frontiers of the sets, manipulate their interiors by statistic treatments. A review of some methods based on sets of constraints is presented in [15, 16] and a comparison with the parametric approaches is given in [17].

Although the methods based on sets of constraints are robust enough for treating most of the cases in mechanical design (included over-constrained assemblies), the complexity of the Minkowski sums makes them time-consuming. We found that such complexity, in the context of geometric tolerancing, is correlated with how the degrees of freedom (DOFs) of the joints and the degrees of invariance of the toleranced features are considered and treated. For example, the sum of the geometric constraints derived from two planar surfaces can be computed in a 3-dimensional space. In the general case, the sets belong to spaces of different dimensions demanding special treatment.

The previous approach, presented in [8], proposes to compute sums in a 6-dimensional space by introducing some additional constraints, called caps. These constraints virtually bound the displacements related to the DOFs of the tolerances joints and the degrees of invariance of the toleranced features. We found that this solution entails an increase of the polytopes complexity due to the propagation of the DOFs along the tolerance chains. This complexity worsens after each sum until becoming far too significant and consumes most of the computational resources.

This paper introduces a new strategy to handle sets of geometric constraints facing the problems entailed by unconstrained displacements during a simula-

tion of tolerances propagation. The new strategy, based on the work presented in [18], proposes to decompose each set of geometric constraints into the sum of a bounded set, representing the limits imposed by the tolerance zones, and an unbounded set, representing the DOFs. When summing two operand sets modelling geometric constraints, only their bounded parts can be considered isolating the rest. As they usually belong to different spaces, we propose to identify the sub-space in which the projection of the operand subsets is bounded. This sub-space is characterized by the displacements that define the relative position between the two involved features. Calculating the sum in this sub-space allows reducing the complexity of the operands significantly and consequently the computational time. This decomposition and the calculation of this sub-space is proposed to be done through kinematic analysis using screws theory.

The article is subdivided into three major parts: firstly, section 2 presents an overview of the general way to model the geometric constraints and a description of the method presented previously (cap-based method). Section 3 introduces the way to operate on polyhedral objects in a tolerancing chain, providing the mathematical background of all the operations required to do a tolerance system reduction (sum and intersection), and explaining the way to know if the functional requirement is accomplished by checking and quantifying the inclusion of the resulting polyhedron inside the functional one. Finally, section 4 illustrates an application in a spectrometer as an example to compare the results and the calculation time using the polyhedra and the cap-based methods; the application is made taking as hypotheses: i) no form defect in surfaces, ii) no local strain due to the contact and iii) no deformable parts.

2. Overview: How to operate in 6D

In mechanical design, a tolerance zone represents the limits of the manufacturing defects for a given feature. When the feature is considered as a discrete set of points, this restriction is applied to each of them. These geometric constraints can be modelled as algebraic constraints expressed at a common 3D

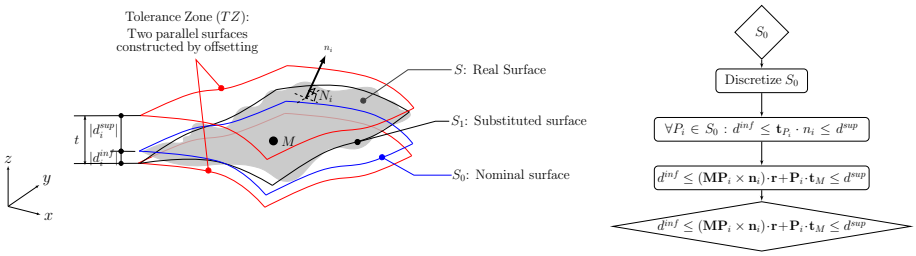


Figure 1: From geometric to algebraic constraints [19]

point $\boldsymbol{M}$; which is assumed to be rigidly linked with the toleranced feature. Figure 1 illustrates this process.

Considering manufacturing defects as small displacements [20], each constraint represents a half-space in 6-dimensional space of deviations [10]:

$$\bar{U}_k^+ = \left\{ \boldsymbol{x} \in \mathbb{R}^6 : b_k + a_{k_1} x_1 + ... + a_{k_6} x_6 \geq 0 \right\} \tag{1}$$

where $x_1 = r_x$, $x_2 = r_y$, $x_3 = r_z$ are the rotation variables, $x_4 = t_x$, $x_5 = t_y$, $x_6 = t_z$ are the translation variables, a_{k_j} $(1 \leq j \leq 6)$ are scalar parameters depending on the geometry of the toleranced feature and the location of the point $\boldsymbol{M}$. The constants b_k are related to the width of the tolerance zone t or the value of the clearance in the contacts. In the common case where the tolerance zone is centered with respect to its nominal surface, all the b_k are equal to $t/2$ (see Figure 1). For more details about the generation of the operands, one can refer to [8].

When a set of m points is considered, a set of $k_{max} = 2m$ half-spaces is obtained. Defining a convex $\mathcal{H}$-polyhedron (where $\mathcal{H}$ stands for half-space) in $\mathbb{R}^6$:

$$\Gamma = \bigcap_{k=1}^{k_{max}} \bar{U}_k^+ \tag{2}$$

As presented in the former equation, a polyhedron with a non-empty interior can be uniquely defined through an intersection of a finite number of half-spaces.

Similarly, a polyhedron representing the allowable displacements of a couple of features potentially in contact inside its clearance can be characterized. For

this case, the toleranced feature is defined in the nominal case of permanent contact between the features and the tolerance zone is obtained by offsetting the toleranced feature according to the clearance value.

Then, the polyhedron may arise from a feature regarding its nominal definition (geometric constraint), two features of two distinct parts sharing a mating condition (contact constraint) or two features of two distinct parts defining a functional specification.

The relative position between any couple of surfaces of an assembly can be simulated operating on polyhedra. Under the consideration of rigid parts, the set of required operations can be determined according to the topological structure of the assembly.

In the case of parts mated in a serial configuration, defects accumulate. This can be modeled by computing the Minkowski sums of the polyhedra derived from the involved surfaces. When parts are mated with multiple contacts, defects counteract between them, so the intersection of the respective polyhedra is required. Once the final calculated polyhedron is obtained (that containing all the cumulative stack-up of variations) the compliance of the functional condition can be verified checking the inclusion of the calculated polyhedron inside the functional one [21].

Except for the case of complex surfaces, the set of constraints derived from a toleranced feature (Eq. (2)) defines an unbounded set in a 6-dimensional space of deviations, i.e. a polyhedron (a higher dimensional prism with a polyhedral cross-section, the details are presented in section 3). This is a consequence of the degrees of invariance of the toleranced features or the degrees of freedom of the joints which define unbounded displacements [8].

From the computational and algorithmic point of view, the manipulation of unbounded objects is more challenging. In response to this, Homri et al. [8] suggested to virtually limit the DOFs (and invariance) of the toleranced joints

and features by introducing additional facets $\bar{U}^+_{c_j}$, called caps.

$$\Gamma' = \left(\bigcap_{k=1}^{k_{max}} \bar{U}^+_k \right) \cap \left(\bigcap_{j=1}^{2d} \bar{U}^+_{c_j} \right) = \Gamma \cap \left(\bigcap_{j=1}^{2d} \bar{U}^+_{c_j} \right) \tag{3}$$

Even if this solution allows to turn the polyhedra into polytopes and to avoid the manipulation of unbounded objects, the addition of cap half-spaces to the operand sets affects the topology of a calculated polytope, making necessary to differentiate the facets that are generated by the cap half-spaces and the ones that derive from geometric and constant constraints. In [8, 22] the authors proposed some methods to deal with this difficulty: i) to create a circumscribed hyper-parallelepiped at an interval d ($d > 0$) from the minimal volume circumscribed, and ii) to trace the operand facets during the different operations.

The alteration of the operands topology by adding the cap facets does not influence the result from the tolerancing point of view, however, it does involve spending time calculating meaningless information. This problem increases after each sum due to the accumulation of the degrees of freedom along the toleranced chains; until becoming far too significant.

3. Proposed approach by prismatic polyhedra

The new method proposed in this paper to operate on sets of constraints is based on an alternative representation of polyhedra.

According to the Minkowski-Weyl theorem [23], a polyhedron can be decomposed into the direct sum of a polytope P (the bounded part) and a polyhedral cone C (the unbounded part):

$$\Gamma = P \oplus C \tag{4}$$

In geometric tolerancing, C is related to the degrees of invariance of the toleranced feature or the degrees of freedom of the joint. In [18] a method based on kinematic analysis with screws for doing this decomposition is presented. Each of these unbounded displacements characterizes a straight line Δ_j in $\mathbb{R}^6$ passing

through the origin:

$$C = \sum_{j=1}^{d} \Delta_j, \quad \text{with } d \leq 6 \tag{5}$$

In geometric tolerancing, the polyhedron can be seen as an 'extrusion' of the underlying polytope (derived from geometric or contact constraints) along its associated straight lines. Such a polyhedron is called prismatic (see Figure 2).

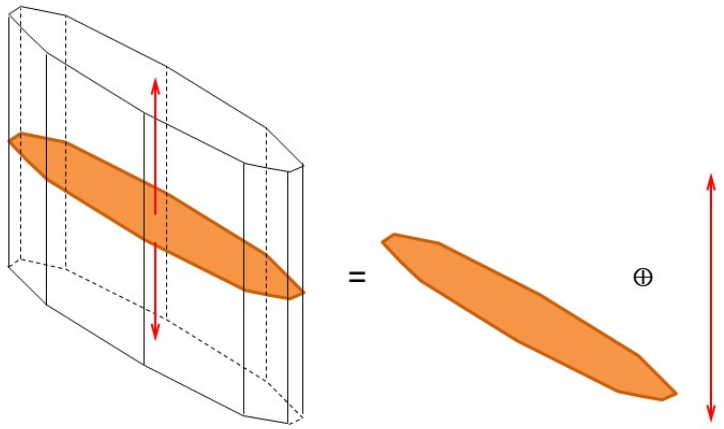


Figure 2: Decomposition of a polyhedron into the sum of a polytope and one straight line Δ

3.1. Contact modelling

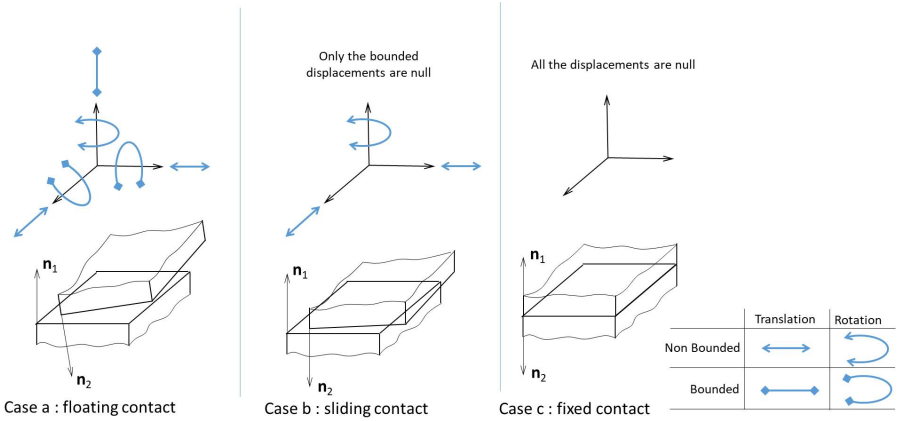


Figure 3: Contact modelling according to functional attributes

The very definition of prismatic polyhedra separate the unbounded part from the bounded one providing the ability to perform a full contact modelling of the mechanism. Mathematically speaking, it is done through all the operations (sum, intersection, inclusion) taking into account the prismatic polyhedra straight lines. This is how, for example, we can model with no restrictive assumption floating, sliding and fixed contacts introduced by [24].

Such a specification is impossible to introduce without restrictive assumptions if we only manipulate single polytopes as in [9], [11], [21] and [22]. Figure 3 illustrates the three main cases of contact attribute of a planar pair where d, the DOFs, is equal to 3. This approach can be applied to any kind of kinematic pair. A floating contact (case i) assumes that the pair of surfaces can be separated (non-null clearance) and the displacements induced by the DOFs are allowable. The sliding contact (case ii) assumes that the clearance is null and the displacements induced by the DOFs are allowable. The fixed contact (case iii) assumes that the clearance is null and the displacements induced by the DOFs are suppressed. In case (i) the bounded displacements can be defined by a full polytope of dimension 3, i.e. $(6-d)$ in the general case, and the non-bounded displacements are an affine space of dimension 3, i.e. d in the general case. The case (ii) derives from the case (i) in which the bounded displacements are reduced to a singleton of dimension 3, i.e. $(6-d)$ in the general case. Finally, the case (iii) derives from the case (ii) in which the non-bounded displacements are restricted to a singleton of dimension 3, i.e. d in the general case.

These topological properties are summarized in Table 1 according to Eq. (6).

$$\Gamma = P + \sum_{j=1}^{d} \Delta_j \ , \text{ with d=DOFs} \tag{6}$$

3.2. Summing prismatic polyhedra

Instead of operating directly on polyhedra in $\mathbb{R}^6$ or their corresponding polytopes made up of cap facets, [18] proposes the manipulation of simplified sets

Table 1: Topological structure of a contact polyhedron

	P	$\sum_{j=1}^{d} \Delta_j$
Case (i) floating contact	Full polytope of dimension (d–6)	Affine space of dimension d
Case (ii) sliding contact	Singleton of dimension $(d-6)$	Affine space of dimension d
Case (iii) fixed contact	Singleton of dimension $(d-6)$	Singleton of dimension d

of constraints by excluding the straight lines derived from the unbounded displacements. Their strategy suggests summing the underlying polytopes of the polyhedra. However, as these polytopes usually belong to spaces of a different dimension, we propose a more significant complexity reduction:

Theorem 3.1 (Sum of prismatic polyhedra)**.** *Let* Γ_1 *and* Γ_2 *be two prismatic polyhedra in* $\mathbb{R}^n$ *such that:*

$$\Gamma_1 = P_1 \oplus \sum_{i=1}^{k} \Delta_i = P_1 \oplus C_1, \quad P_1 \subset H_{P_1} = \bigcap_{i=1}^{k} H_i$$

$$\Gamma_2 = P_2 \oplus \sum_{i=k+1}^{l} \Delta_i = P_2 \oplus C_2, \quad P_2 \subset H_{P_2} = \bigcap_{i=k+1}^{l} H_i$$

$$\text{with } H_i = \Delta_i^{\perp} \ \forall i \in \{1, ..., l\}$$

$\Gamma_1 \oplus \Gamma_2$ *can be calculated as the sum of the projection of their underlying polytopes on the subspace* $H_{P_1} \cap H_{P_2}$ *plus their respective straight lines:*

$$\Gamma_1 \oplus \Gamma_2 = \pi_{H_{P_1} \cap H_{P_2}}(P_1) \oplus \pi_{H_{P_1} \cap H_{P_2}}(P_2) \oplus \sum_{i=1}^{l} \Delta_i$$

where π_H *is the orthogonal projection on* H*.*

Proof. We first want to prove the theorem for $\Gamma_1 = P_1 \oplus \Delta_1$ and $\Gamma_2 = P_2 \oplus \Delta_2$ with prismatic polyhedra which hold only one straight line. The only assumption we make is that Δ_1 and Δ_2 are not parallel (i.e. not equal as all the straight lines pass through the origin), this obviously implies that H_1 and H_2 are not parallel:

$$\Gamma_1 \oplus \Gamma_2 = P_1 \oplus P_2 \oplus \Delta_1 \oplus \Delta_2, (\Delta_1 \nparallel \Delta_2)$$

We need first the following lemma to carry on the demonstration.

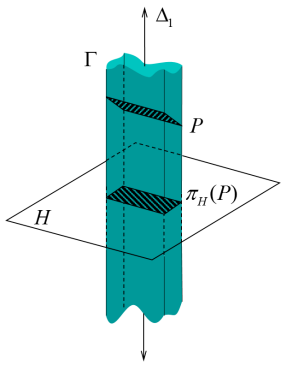


Figure 4: Illustration of $P \oplus \Delta = \pi_H(P) \oplus \Delta$.

Lemma 3.2. *Let P be a polytope and Δ a straight line*

$$P \oplus \Delta = \pi_H(P) \oplus \Delta, \quad \textit{with } \Delta \perp H$$

Do we have $P \oplus \Delta \subset \pi_H(P) \oplus \Delta$? Every point of P can be written as the sum of its projection on H plus a vector of Δ so $P \oplus \Delta \subset \pi_H(P) \oplus \Delta$. Reversely if $\boldsymbol{x} \in P$, after projection on the hyperplane H: $\pi_H(\boldsymbol{x}) = \boldsymbol{x} \ -(\delta_0 + \boldsymbol{\delta} \cdot \boldsymbol{x})\boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is a unit normal of H and δ_0 the constant of H. Consequently we can write $\pi_H(\boldsymbol{x})$ as the sum of a point of P and a vector of Δ, as $H \perp \Delta$ (see Figure 4) so $\pi_H(P) \oplus \Delta \subset P \oplus \Delta$.

$$P_1 \oplus \Delta_1 \oplus \Delta_2 = \pi_{H_1}(P_1) \oplus \Delta_1 \oplus \Delta_2$$

$\pi_{H_1}(P_1)$ is a polytope too so we can now project it on H_2:

$$P_1 \oplus \Delta_1 \oplus \Delta_2 = \pi_{H_2}\big(\pi_{H_1}(P_1)\big) \oplus \Delta_1 \oplus \Delta_2$$

Then we can run again the same projections:

$$P_1 \oplus \Delta_1 \oplus \Delta_2 = \pi_{H_2}\Big(\pi_{H_1}\big(...\pi_{H_2}\big(\pi_{H_1}(P_1)\big)\big)\Big) \oplus \Delta_1 \oplus \Delta_2$$

We need to determine what is the limit of all of the alternate projections $\pi_{H_2}\Big(\pi_{H_1}\big(...\pi_{H_2}(\pi_{H_1}(P_1))\big)\Big)$ with $H_1 \cap H_2 \neq \emptyset$ (it is the case as $\Delta_1 \nparallel \Delta_2$). This property has been studied in [25] and [26], the successive projections of a point $\boldsymbol{x}$ on two convex sets H_1 and H_2 converges towards a point $\boldsymbol{x}^* \in H_1 \cap H_2$

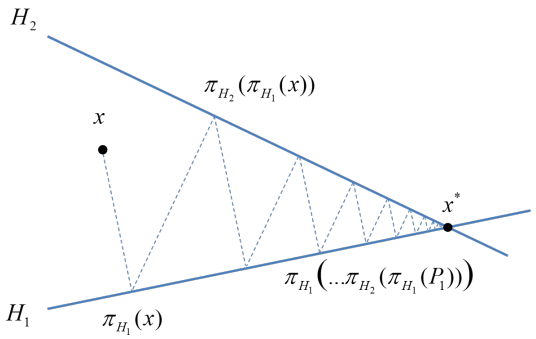


Figure 5: Successive projections in a 2D vector space.

whatever the initial point is (see Figure 5). We know that the limit of our suite belongs to $H_1 \cap H_2$ and that the elements of this suite are inside the space $V = x + Vect(H_1^{\perp}, H_2^{\perp}) = x + (H_1 \cap H_2)^{\perp}$. So $x* \in (V \cap H_1 \cap H_2) = \{x + (H_1 \cap H_2)^{\perp}\} \cap H_1 \cap H_2$, this latter set is reduced to one point so $x* = V \cap H_1 \cap H_2$. Given that $V \perp H_1 \cap H_2, x* = \Pi_{H_1 \cap H_2}(x)$. So it proves the following property:

$$P_1 \oplus \Delta_1 \oplus \Delta_2 = \pi_{H_1 \cap H_2}(P_1) \oplus \Delta_1 \oplus \Delta_2$$

Now we can introduce another straight line Δ_3:

$$\begin{aligned}
&P_1 \oplus \Delta_1 \oplus \Delta_2 \oplus \Delta_3 \\
&= \pi_{H_2 \cap H_3}(P_1) \oplus \Delta_1 \oplus \Delta_2 \oplus \Delta_3 \\
&= \pi_{H_1}\left(\pi_{H_2 \cap H_3}(P_1)\right) \oplus \Delta_1 \oplus \Delta_2 \oplus \Delta_3 \\
&= \pi_{H_1}\left(\pi_{H_2 \cap H_3}\left(\ldots \pi_{H_2 \cap H_3}(\pi_{H_1}(P_1))\right)\right) \oplus \Delta_1 \oplus \Delta_2 \oplus \Delta_3
\end{aligned}$$

Applying the same reasoning than before gives:

$$P_1 \oplus \Delta_1 \oplus \Delta_2 \oplus \Delta_3 = \pi_{H_1 \cap H_2 \cap H_3}(P_1) \oplus \Delta_1 \oplus \Delta_2 \oplus \Delta_3$$

By recurrence we obtain Theorem 3.1:

$$\begin{aligned}
\Gamma_1 \oplus \Gamma_2 &= P_1 \oplus P_2 \oplus \sum_{i=1}^{l} \Delta_i \\
&= \pi_{H_{P_1} \cap H_{P_2}}(P_1) \oplus \pi_{H_{P_1} \cap H_{P_2}}(P_2) \oplus \sum_{i=1}^{l} \Delta_i
\end{aligned}$$

□

Such a property is really interesting in the sense that the sum is now to be computed in a subspace of significantly reduced dimension.

In the context of geometric tolerancing, calculating $H_{P_1} \cap H_{P_2}$ can be done by means of a kinematic analysis manipulating screws. This subspace is characterized by the displacements defining the relative position of the involved toleranced features. In order to go back to the original space $\mathbb{R}^n$ to have the full expression of $\Gamma_1 + \Gamma_2$, the calculated polytope which lives in the subspace $H_{P_1} \cap H_{P_2}$ must be finally extruded. This operation can be done by summing it to $C_1 + C_2$ i.e. the straight lines of the operands.

From Lemma 3.2 we can deduce directly the following corollary.

Corollary 3.2.1. *Let Γ be a prismatic polyhedra in $\mathbb{R}^n$ such that $\Gamma = P \oplus C$ with $C^\perp = H_P$.*

$$\Gamma = \pi_{H_P}(P) \oplus C = \pi_{H_P}(\Gamma) \oplus C \tag{7}$$

3.3. Intersecting prismatic polyhedra

We compute the intersection of prismatic polyhedra in order to determine the relative displacements between two surfaces from two different parts which share multiple contacts.

With this operation we are interested in calculating the intersection of $C_1 \cap C_2 = \sum_i \Delta_{1,i} \cap \sum_j \Delta_{2,j} = (H_{P_1} \oplus H_{P_2})^\perp$ to know the resulting common straight lines. If we note respectively $\delta_{1,i}$ and $\delta_{2,j}$ the vectors defining $\Delta_{1,i}$ and $\Delta_{2,j}$, with α_i and β_j being real numbers, we want to solve $\sum_{i=1}^{k} \alpha_i \delta_{1,i} = \sum_{j=1}^{l} \beta_j \delta_{2,j}$ to find the elements belonging simultaneously to C_1 and C_2. So we introduce the matrix A whose columns are the vectors $\delta_{1,i}\ \forall i \in \{1, ..., k\}$ and the vectors $-\delta_{2,j}\ \forall j \in \{1, ..., l\}$ to compute its null-space $N(A) = \{x \in \mathbb{R}^{k+l} : A.x = 0\}$.

$$A.x = \left(\begin{array}{c|c|c|c|c|c} \delta_{1,1} & ... & \delta_{1,k} & -\delta_{2,1} & ... & -\delta_{2,l} \end{array} \right).x = 0$$

A common result in linear algebra tells that a basis in $N(A)$ can be obtained through a gaussian elimination of the matrix A up to the reduced row echelon form. Then the k first columns of $N(A)$ vector basis provide a basis of $\Delta = C_1 \cap C_2$. Let's note $\Psi = H_{P_1} \oplus H_{P_2}$, as a consequence $\Delta^\perp = \Psi$.

Proposition 3.3. *The intersection of two prismatic polyhedra of $\mathbb{R}^n$ can be calculated adding the intersection of their projections on their added subspace with their common straight lines:*

$$\begin{aligned}\Gamma_1 \cap \Gamma_2 &= \Big(\pi_{H_{P_1} \oplus H_{P_2}}(\Gamma_1) \cap \pi_{H_{P_1} \oplus H_{P_2}}(\Gamma_2)\Big) \oplus C_1 \cap C_2 \\ &= \Big(\pi_{\Psi}(\Gamma_1) \cap \pi_{\Psi}(\Gamma_2)\Big) \oplus \Delta\end{aligned}$$

Proof. $\Gamma = \Gamma_1 \cap \Gamma_2 = \Big(\pi_{H_{P_1}}(\Gamma_1) + C_1\Big) \cap \Big(\pi_{H_{P_2}}(\Gamma_2) + C_2\Big)$

We first want to prove that Γ contains exactly all the straight lines of $C_1 \cap C_2$. $\forall D \in \Delta = C_1 \cap C_2, D \in C_1 \Rightarrow D \in \Gamma_1, D \in C_2 \Rightarrow D \in \Gamma_2$ so $D \in \Gamma_1 \cap \Gamma_2$. Now we have to check if there is a straight line in Γ which does not belong to $C_1 \cap C_2$. Let's assume that $D \notin C_1 \cap C_2$ such as $D \in \Gamma_1 \cap \Gamma_2$. Without loss of generality we can make the assumption that D is not in C_1. This is not possible as D is not a bounded object which belongs to Γ_1 and the only unbounded part of Γ_1 is C_1. So $D \in C_1$ and we can use the same reasoning to prove that $D \in C_2$. $\Gamma_1 \cap \Gamma_2$ contains exactly the straight lines belonging to $\Delta = C_1 \cap C_2$. As the intersection of two polyhedra is a polyhedron, $\Gamma_1 \cap \Gamma_2$ is a prismatic polyhedron. We can write $\Gamma_1 \cap \Gamma_2 = \pi_{\Delta^\perp}(\Gamma_1 \cap \Gamma_2) + \Delta$.

Let's prove that $\Gamma_1 \cap \Gamma_2 = \pi_{\Delta^\perp}(\Gamma_1) \cap \pi_{\Delta^\perp}(\Gamma_2) + \Delta$. We already know that the projection of the intersection of two sets is included into the intersection of these two projected sets i.e. $\pi_{\Delta^\perp}(\Gamma_1 \cap \Gamma_2) \subset \pi_{\Delta^\perp}(\Gamma_1) \cap \pi_{\Delta^\perp}(\Gamma_2)$.

The final part of the demonstration needs to check that $\pi_{\Delta^\perp}(\Gamma_1) \cap \pi_{\Delta^\perp}(\Gamma_2) + \Delta \subset \pi_{\Delta^\perp}(\Gamma_1 \cap \Gamma_2) + \Delta = \Gamma_1 \cap \Gamma_2$. We can decompose an element $z \in \pi_{\Delta^\perp}(\Gamma_1) \cap \pi_{\Delta^\perp}(\Gamma_2) + \Delta$ into the sum of an element of $(C_1 \cap C_2)^\perp$ and an element of its orthogonal space $C_1 \cap C_2$. $z = x + y$, with $x \in \pi_{\Delta^\perp}(\Gamma_1) \cap \pi_{\Delta^\perp}(\Gamma_2)$ and $y \in C_1 \cap C_2$. As $\Delta = (C_1 \cap C_2) \subset C_1$ we have $C_1^\perp \subset \Delta^\perp$ so $\pi_{C_1^\perp} \circ \pi_{\Delta^\perp} = \pi_{C_1^\perp}$. Given that $x \in \pi_{\Delta^\perp}(\Gamma_1)$, $\pi_{\Delta^\perp}(x) \in \pi_{C_1^\perp} \circ \pi_{\Delta^\perp}(\Gamma_1) = \pi_{C_1^\perp}(\Gamma_1)$. As a consequence $x + y \in \pi_{C_1^\perp}(\Gamma_1) + C_1$ which means that $x + y \in \Gamma_1$. We can demonstrate that $x + y \in \Gamma_2$ following the same way.

□

3.4. Verifying the inclusion of the resulting polyhedron into the target

Once the resulting polyhedron Γ_R is calculated, we can verify the compliance of the functional condition by checking the inclusion of Γ_R inside the functional polyhedron Γ_{FC}. To do that it is necessary:

1. to verify that the space generated by the set of straight lines coming from the resulting polyhedron is included into the space generated by the set of straight lines coming from the functional polyhedron
2. to check and quantify the inclusion of the resulting polyhedron inside the functional one

The first point consists in verifying the kinematic compliance of both unbounded sets [27]. If the kinematic compliance is not achieved, it is impossible to accomplish the functional requirement without modifying the mechanism. In other words, there is at least one DOF which prevents the respect of the functional condition. If the kinematic compliance is achieved, it is possible to quantify the inclusion of the resulting polyhedron Γ_R inside the functional one Γ_{FC}. If such an inclusion is satisfied we compute the minimum distance between the two operands. If not, we compute the maximum distance between the functional one and the points of Γ_R located outside Γ_{FC}. In both cases, we decide to optimize the size of the functional polyhedron Γ_{FC} to circumscribe Γ_R.

The functional polyhedron Γ_{FC} can be expressed as a weighted Minkowski sum between two operands, Γ_1 and Γ_2, derived from the surfaces related to the functional condition according to Figure 1. Let's consider Γ'_1 and Γ'_2 the respective polyhedra deriving from the functional condition surfaces. So we have the definitions (8):

$$\begin{cases} \Gamma'_1 = \bigcap_{i=1}^{k} \left\{ \boldsymbol{x} \in \mathbb{R}^6 : \frac{t_{f_1}}{2} + a_{i_1} x_1 + ... + a_{i_6} x_6 \geq 0 \right\} \\ \Gamma'_2 = \bigcap_{j=1}^{l} \left\{ \boldsymbol{x} \in \mathbb{R}^6 : \frac{t_{f_2}}{2} + a_{j_1} x_1 + ... + a_{j_6} x_6 \geq 0 \right\} \end{cases} \tag{8}$$

t_{f_1} and t_{f_2} are the respective functional tolerances assuming that the tolerance zones are centered around the nominal surfaces. In practice, these functional tolerances are very often equal (i.e. combined zone concept from ISO) or one of them is null (i.e. datum). From (8), we can obtain (9), [28] [29].

Algorithm 1 Including Γ_R into Γ_{FC}: verification and fitting

Require: $\Gamma_R, k_1, \Gamma_1, k_2, \Gamma_2, t_f, \delta$

Ensure: $t_{f_{circ}}$ such that $\Gamma_R \in \Gamma_{FC}$

Build the functional polyhedron: $\Gamma_{FC} = t_f(k_1.\Gamma_1 \oplus k_2.\Gamma_2)$
Let $\{\Delta_i\}$ be the set of straight lines of Γ_{FC}
Let $\{\Delta_j\}$ be the set of straight lines of Γ_R
if $\left\{\sum_j \beta_j \Delta_j\right\} \subset \left\{\sum_i \alpha_i \Delta_i\right\}$ **then**
  // The kinematic compliance is verified
  Check the inclusion: $\Gamma_R \subset \Gamma_{FC}$
  if $\Gamma_R \subset \Gamma_{FC}$ **then**
    // The tolerance compliance is achieved
    Compute the distance (D^a_{min})
    (D^a_{min}) minimum distance between Γ_{FC} and Γ_R
    Consider a value $\delta > 0$ to scale up Γ_{FC}
    $\Gamma'_{FC} = (t_f + \delta)[k_1.\Gamma_1 \oplus k_2.\Gamma_2]$
    Compute the distance (D^b_{min})
    (D^b_{min}) minimum distance between Γ'_{FC} and Γ_R
    Compute $t_{f_{circ}} = t_f - \frac{D^a_{min}}{D^b_{min} - D^a_{min}}\delta$
  else if $\Gamma_R \not\subset \Gamma_{FC}$ **then**
    // The tolerance compliance is not achieved
    Compute the distance (D^a_{max})
    (D^a_{max}) maximum distance between Γ_{FC} and a point of Γ_R located outside Γ_{FC}
    Consider a value $\delta, t_f > \delta > 0$ to scale down Γ_{FC}
    $\Gamma'_{FC} = (t_f - \delta)[k_1.\Gamma_1 \oplus k_2.\Gamma_2]$
    Compute the distance (D^b_{max})
    (D^b_{max}) maximum distance between Γ'_{FC} and a point of Γ_R located outside Γ_{FC}
    Compute $t_{f_{circ}} = t_f + \frac{D^a_{max}}{D^b_{max} - D^a_{max}}\delta$
  end if
else
  // The inclusion cannot be achieved
end if

$$\Gamma_{FC} = \Gamma'_1 \oplus \Gamma'_2 = \tfrac{1}{2} t_{f_1} \Gamma_1 \oplus \tfrac{1}{2} t_{f_2} \Gamma_2$$
$$\begin{cases} \Gamma_1 = \bigcap_{i=1}^{k} \left\{ \boldsymbol{x} \in \mathbb{R}^6 : 1 + a_{i_1} x_1 + ... + a_{i_6} x_6 \geq 0 \right\} \\ \Gamma_2 = \bigcap_{j=1}^{l} \left\{ \boldsymbol{x} \in \mathbb{R}^6 : 1 + a_{j_1} x_1 + ... + a_{j_6} x_6 \geq 0 \right\} \end{cases} \tag{9}$$

Finally, we obtain the weighted Minkowski sum as follows:

$$\Gamma_{FC} = t_f [k_1.\Gamma_1 \oplus k_2.\Gamma_2] \text{ with } k_1 \geq 0, k_2 \geq 0, k_1 + k_2 \neq 0 \tag{10}$$

The multiplication of a polyhedron by a scalar is distributive over the Minkowski sum [30]. The last property implies that for real positive numbers λ and μ:

$$\begin{aligned} \lambda(\Gamma_A \oplus \Gamma_B) &= \lambda\Gamma_A \oplus \lambda\Gamma_B \\ (\lambda + \mu)\Gamma_A &= \lambda\Gamma_A \oplus \mu\Gamma_A \end{aligned} \tag{11}$$

Thanks to the former properties, the functional polyhedron can be modified by scaling the term t_f in Eq. (10). If the ratio (k_1/k_2) is constant, the topology of the resultant polyhedron is maintained, and all the functional polyhedra that can be obtained are homothetic to each other.

The algorithm 1 presents the strategy to verify the compliance of the resulting polyhedron Γ_R inside the functional polyhedron Γ_{FC}, and it allows to find an optimal tolerance $t_{f_{circ}}$ for the surfaces related in the functional condition, starting from an initial tolerance t_f. In the algorithm, the tolerance is a function of the distance (D) between the two operands, which means that if the distance is too big, and the resulting polyhedron is included, the tolerance is going to increase proportionally.

If the two polyhedra are kinematically compliant, we consider $\Gamma_{FC_{circ}}$ the circumscribed polyhedron to Γ_R such as (12).

$$\Gamma_{FC_{circ}} = t_{f_{circ}} [k_1.\Gamma_1 \oplus k_2.\Gamma_2],\ \Gamma_R \subset \Gamma_{FC_{circ}},\ \Gamma_R \cap \Gamma_{FC_{circ}} \neq \emptyset. \tag{12}$$

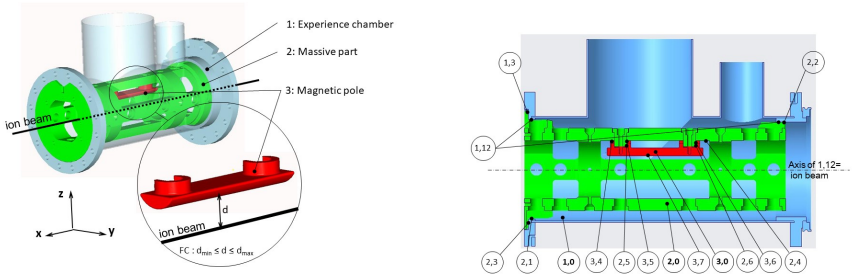
$t_{f_{circ1}}$ and $t_{f_{circ2}}$ are the optimal functional tolerances whatever the starting case (inclusion or not inclusion):

$$\begin{cases} \text{if } \Gamma_R \subset \Gamma_{FC}, t_{f_{circ1}} = 2k_1.t_{f_{circ}} \leq t_{f_1}, t_{f_{circ2}} = 2k_2.t_{f_{circ}} \leq t_{f_2} \\ \text{if } \Gamma_R \not\subset \Gamma_{FC}, t_{f_{circ1}} = 2k_1.t_{f_{circ}} \geq t_{f_1}, t_{f_{circ2}} = 2k_2.t_{f_{circ}} \geq t_{f_2} \end{cases}$$

The deviations $|t_{f_{circi}} - t_{f_i}|$ are proportional to the inclusion or the non inclusion of Γ_R inside Γ_{FC}. In other words, these deviations quantify the compliance of the mechanism with respect to the functional condition FC.

4. Case study: application to a spectrometer

To compare the strategies presented in this paper, we'll do the complete tolerance analysis process for a high-resolution spectrometer, starting from the reduction and finishing with the inclusion verification and optimization. This spectrometer is used in nuclear physics to check the number of atoms in an ion beam experimentally. The resolution of this spectrometer is strongly correlated with the location between the cylindrical surface of the magnetic pole 3 (surface $(3, 7)$) and the ion beam, see Figure 6a. In the following, we will assume that the ion beam is the axis of the cylindrical surface $(1, 12)$. The surface $(1, 12)$ is a unified feature made up of two coaxial cylindrical surfaces from the experience chamber 1 as shown in Figure 6b.



(a) CAD model - spectrometer

(b) Parts and surfaces enumeration of the spectrometer

Figure 6: Spectrometer: CAD model and enumeration of parts and surfaces.

Since the proper functioning of the spectrometer depends on the location of the surfaces $(3, 7)$ and $(1, 12)$, from now on we are going to call them the handle surfaces. Seeing that the objective of the simulation is to control the relative position of the handle surfaces considering manufacturing and contact deviation

on the mating parts, the functional requirement is built through the Minkowski sum of the operands that define the handle surfaces, as shown in Eq. 10, with $t_f = 0.4$ and $k_1 = k_2 = 1/2$. According to the enumeration of the parts and the surfaces presented in Figure 6b, the topological model of the assembly is presented in Figure 7. In this graph, nodes designated as (α, β) represent the nominal model of the part α when $\beta = 0$, and the substitute surfaces when $\beta \neq 0$. Each edge of the graph represents geometric deviations, in the case of edges within one part, or deviations due to contacts, in the case of edges connecting two nodes from different parts. These deviations can be represented by geometric and contact polyhedra respectively [8].

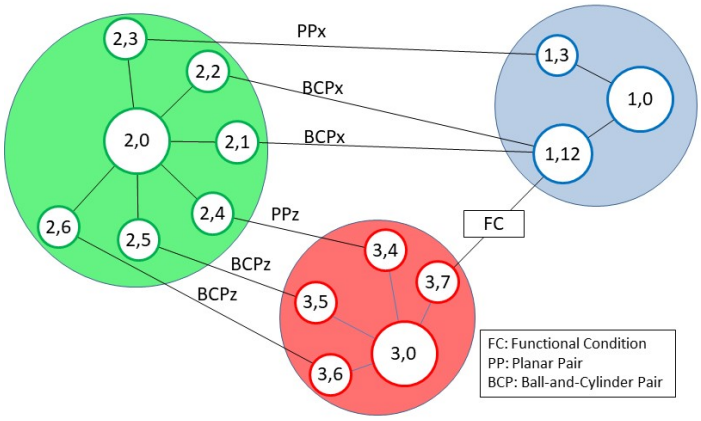


Figure 7: Contact graph of the spectrometer

The set of operations required to simulate the relative position of the handle surfaces can be determined by doing the graph reduction as follows:

$$\Gamma_c = \Gamma_{3,7/3,0} \oplus \Gamma_{3,0/2,0} \oplus \Gamma_{2,0/1,12} \tag{13}$$

with:

$$\begin{aligned}
\Gamma_{3,0/2,0} &= \Gamma_{3,0/2,0-a} \cap \Gamma_{3,0/2,0-b} \cap \Gamma_{3,0/2,0-c} \\
\Gamma_{3,0/2,0-a} &= \Gamma_{3,0/3,4} \oplus \Gamma_{3,4/2,4} \oplus \Gamma_{2,4/2,0} \\
\Gamma_{3,0/2,0-b} &= \Gamma_{3,0/3,5} \oplus \Gamma_{3,5/2,5} \oplus \Gamma_{2,5/2,0} \\
\Gamma_{3,0/2,0-c} &= \Gamma_{3,0/3,6} \oplus \Gamma_{3,6/2,6} \oplus \Gamma_{2,6/2,0}
\end{aligned} \tag{14}$$

and with:

$$\begin{aligned}
\Gamma_{2,0/1,12} &= \Gamma_{2,0/1,12-a} \cap \Gamma_{2,0/1,12-b} \\
\Gamma_{2,0/1,12-a} &= (\Gamma_{2,0/2,1} \oplus \Gamma_{2,1/1,12}) \cap (\Gamma_{2,0/2,2} \oplus \Gamma_{2,2/1,12}) \\
\Gamma_{2,0/1,12-b} &= \Gamma_{2,0/2,3} \oplus \Gamma_{2,3/1,3} \oplus \Gamma_{1,3/1,0} \oplus \Gamma_{1,0/1,12}
\end{aligned} \tag{15}$$

All the operands involved in the former relations were created with the open-source software PolitoCAT [31] and calculated at the point $M\ (35, 0, 20)$.

The tolerance and dimensional limits are taken into account based on the application of the spectrometer and the manufacturing processes that are usually used on the parts (see Figures 8–10 and Tables 2– 4). With the information presented, we have the tolerance zone for the calculation of the geometric operands and the clearance for the contact operands. The clearance is calculated taking into account the most undesirable condition, Least Material Condition (LMC). It is considered the maximum size for the holes and the minimum size for the shafts.

When applying the cap-based method, the unbounded displacements of the toleranced features are treated through the use of cap facets and the phenomenon of propagation of cap facets appears all along the simulation making it to take almost 7 hours. As shown in Table 5, 98.7% of the calculated facets are capped, and therefore they have no meaning in the related tolerancing problem.

Figure 11 represents graphically the problem of caps propagation taking as an example the computation of the operand $\Gamma'_{1,3/1,12}$ which corresponds to the relative position of the plane surface $(1, 3)$ with respect to the cylindrical surface $(1, 12)$. The operands are projected in the subspace spanned by $[r_y, r_z, t_z]$ to generate 3D representations. It is worth mentioning that these figures are just

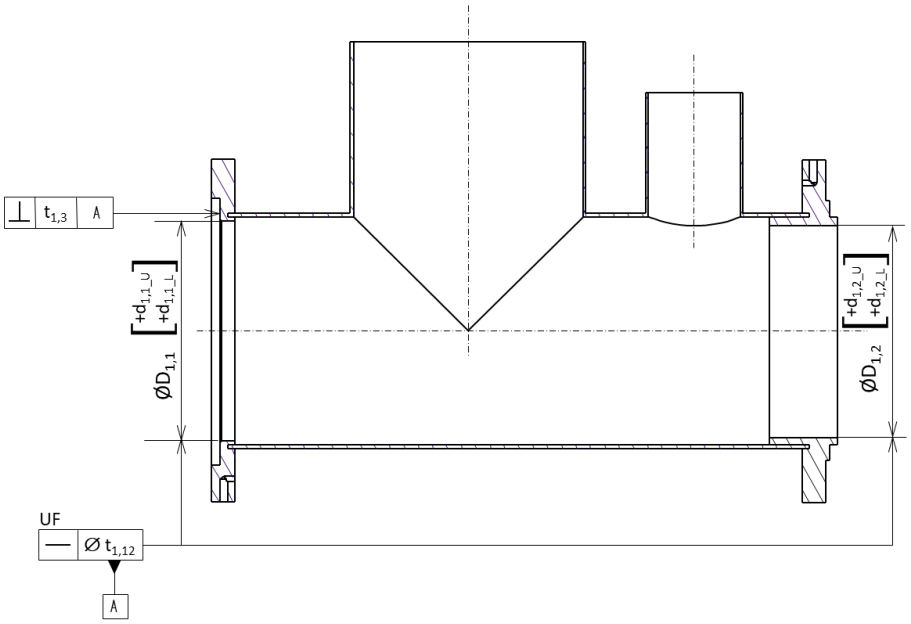


Figure 8: Experience Chamber Drawing

$t_{1,12}$	0.020
$\varnothing D_{1,1}$	145
$d_{1,1\_U}$	+0.040
$d_{1,1\_L}$	0
$\varnothing D_{1,2}$	138
$d_{1,2\_U}$	+0.040
$d_{1,2\_L}$	0
$t_{1,3}$	0.012

Table 2: Experience chamber dimensions and tolerances

partial representations since their original polytopes belong to spaces with a dimension larger than three.

In Figure 11, it is possible to see the pair of cap facets (in red) introduced to bound t_z in the operand $\pi(\Gamma'_{1,3/1,0})$; after the computation, many cap facets and consequently unnecessary vertices appear in the calculated polytope $\pi(\Gamma'_{1,3/1,12})$.

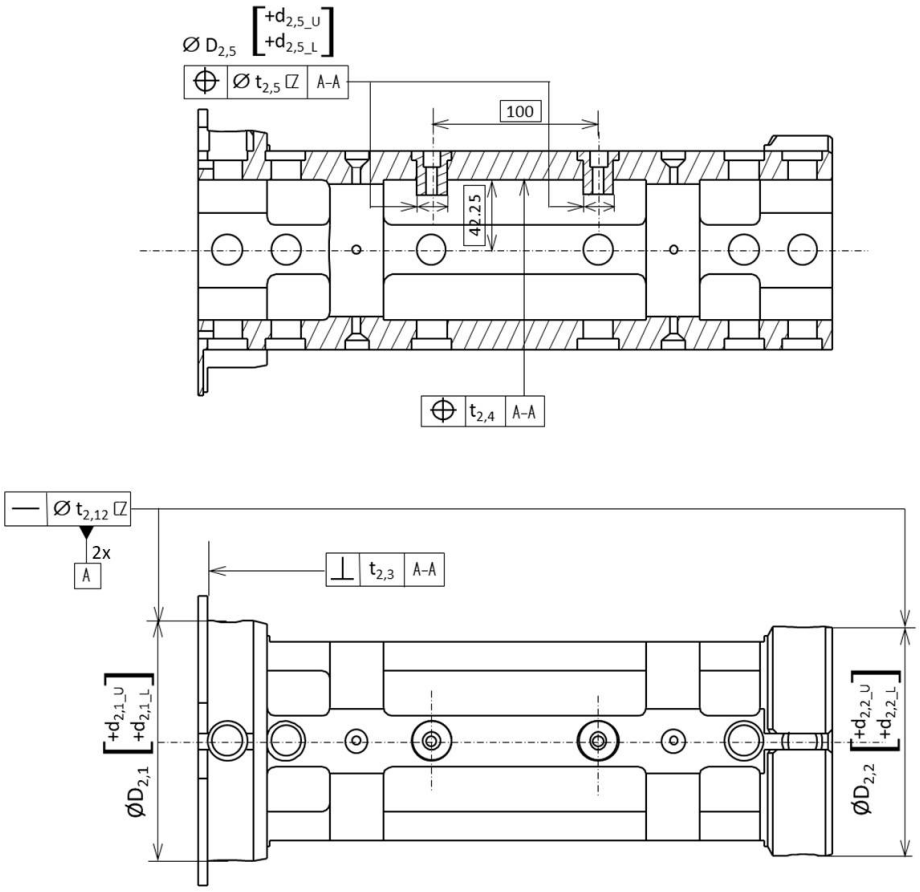


Figure 9: Massive part drawing

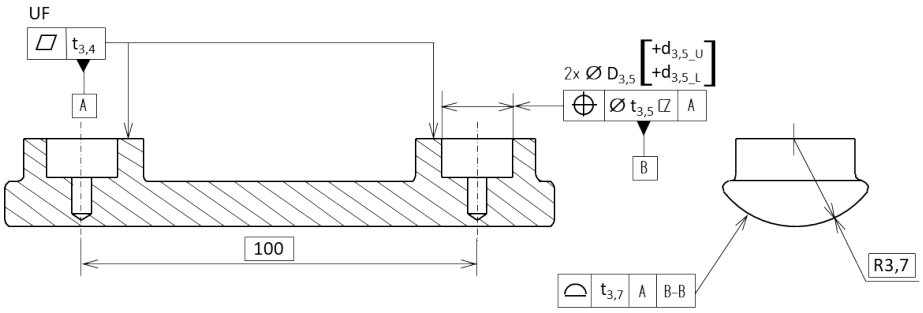


Figure 10: Magnetic pole drawing

$t_{2,12}$ $(t_{2,1} = t_{2,2})$	0.020
$\varnothing D_{2,1}$	145
$d_{2,1\_U}$	$+0.040$
$d_{2,1\_L}$	-0.039
$\varnothing D_{2,2}$	138
$d_{2,2\_U}$	$+0.040$
$d_{2,2\_L}$	-0.039
$t_{2,3}$	$+0.02$
$t_{2,4}$	$+0.025$
$\varnothing D_{2,5} = \varnothing D_{2,6}$	18
$d_{2,5\_U} = d_{2,6\_U}$	$+0.029$
$d_{2,5\_L} = d_{2,6\_L}$	$+0.018$

Table 3: Massive part dimensions and tolerances

$t_{3,4}$	0
$\varnothing D_{3,5} = \varnothing D_{3,6}$	18
$d_{1,1\_U}$	0.059
$d_{1,1\_L}$	0.032
$t_{3,5} = t_{3,6}$	0.025
$R_{3,7}$	23
$t_{3,7}$	0.01

Table 4: Magnetic pole dimensions and tolerances

This phenomenon is associated with the propagation of the unconstrained displacement t_z along the kinematic chain.

When applying the strategy based on polyhedra, the computational time is reduced 99.9% with respect to the cap-based method, see Table 6. This reduction is achieved by the fact that no cap-facets are introduced and instead the DOFs are treated separately employing straight lines avoiding the propagation of unconstrained displacements along the kinematic chain and allowing to

Table 5: Simulation following the cap-based method

Dimension of the calculation space	6
Number of sums	11
Number of intersections	4
Number of cap facets calculated	80903
Number of facets of the result	1024
Computational time	6.73h

Computations performed with the library politopix [31] with an Intel(R) Core(TM) i5-8265U CPU.

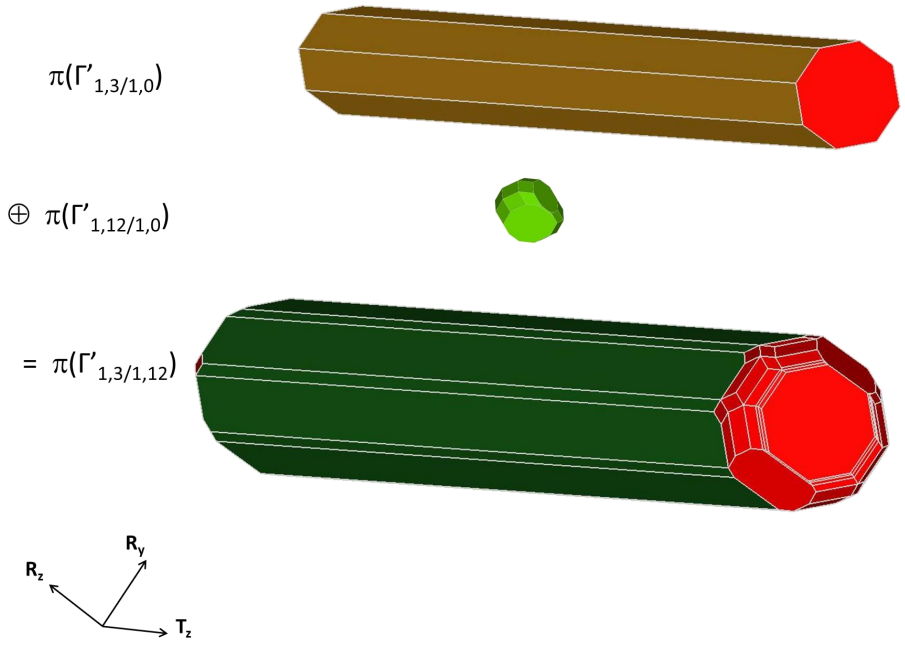


Figure 11: 3D representation of the computation of $\Gamma'_{1,3/1,12}$. Cap facets are shown in red.

calculate only the facets related to the tolerance constraints.

In Figure 13, we show the result of the tolerancing analysis projected in the subspace $[r_y, r_z, t_z]$. The chosen subspace contains only the bounded displacements of the resulting operand, meaning that no straight-lines or cap facets are

Table 6: Simulation following the prismatic polyhedra method

Dimension of the calculation space	3
Number of sums	8
Number of intersections	4
Number of facets of the result	1001
Computational time	12.96 s

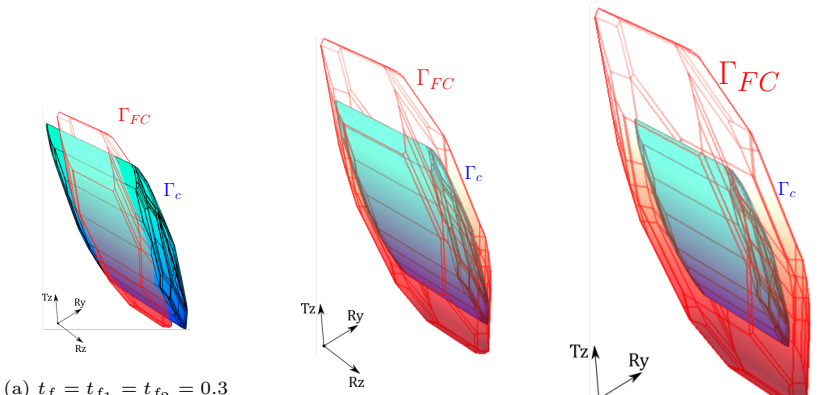
Computations performed with the library politopix [31] with an Intel(R) Core(TM) i5-8265U CPU.

shown in the projection. The Figure 13a represents the result obtained with capped polytopes while Figure 13b shows the result with polyhedra operands, as it can be seen, both results are equal, and it can be proved by comparing the volume of the two results, capped polytope(V_{ctop}) and the polyhedron(V_{ppld}), with the volume of their intersection(V_{int}):

$$V_{ctop} = V_{ppld} = V_{int}$$

Since the kinematic compliance is satisfied, the resulting or the functional polyhedron can be modified to achieve a better fitting. In the Figure 12 we can see the three cases that we can have:i) the resulting polyhedron is included inside the functional polyhedron, but the fitting can be improved (Figure 12a), ii) the resulting polyhedron is not included inside the functional polyhedron, but the inclusion can be achieved (Figure 12c), and iii) the inclusion is achieved, and the minimum distance between the nodes of the functional and the resulting polyhedron is close to 0 (Figure 12b). Whatever the case, the tolerances t_f can be modified (decreased or increased) to obtain the optimal case. For the tolerance and dimensional limits used during the simulation, the minimum functional tolerance that can be satisfied by the system is $t_{f_{circ}} = 0.394 < t_f = 0.4$, meaning that the functional tolerances of the target can be decreased to 0.006 and still guarantee the inclusion of the resulting polyhedron (eq. (16)).

$$\Gamma_{FC} = t_{f_{circ}}(k_1.\Gamma_{3,7/3,0} \oplus k_2.\Gamma_{1,0/1,12}), \text{ where: } k_1 = k_2 = 1/2 \qquad (16)$$

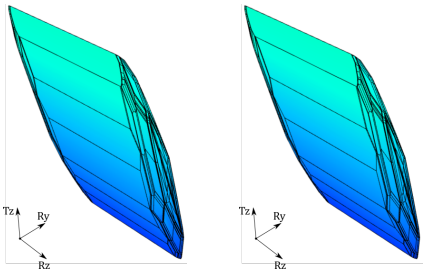


(a) $t_f = t_{f_1} = t_{f_2} = 0.3$

(b) $t_{f_{circ}} = t_{f_{circ1}} = t_{f_{circ2}} = 0.394$

(c) $t_f = t_{f_1} = t_{f_2} = 0.5$

Figure 12: Verifying the inclusion of the resulting polyhedron (Γ_c) inside the functional one (Γ_{FC}) and improving the fitting: a) The resulting polyhedron is not included into the target, the value of t_f must be increased to achieve $t_{f_{circ}}$; b) The resulting polyhedron is included into the target and $t_f = t_{f_{circ}}$; c) The resulting polyhedron is included into the target but t_f can be decreased until reaching $t_{f_{circ}}$.



(a) Result obtained by means of the capped operands

(b) Result obtained using Polyhedron operands

Figure 13: 3D representation of Γ'_c.

The discretization of the features while creating the operands plays an important role in terms of precision and computational time. Figure 14 shows how the value of $t_{f_{circ}}$ varies as the amount of nodes increases. After 10 nodes, the value of $t_{f_{circ}}$ stabilizes in a range between 0.397 and 0.393.

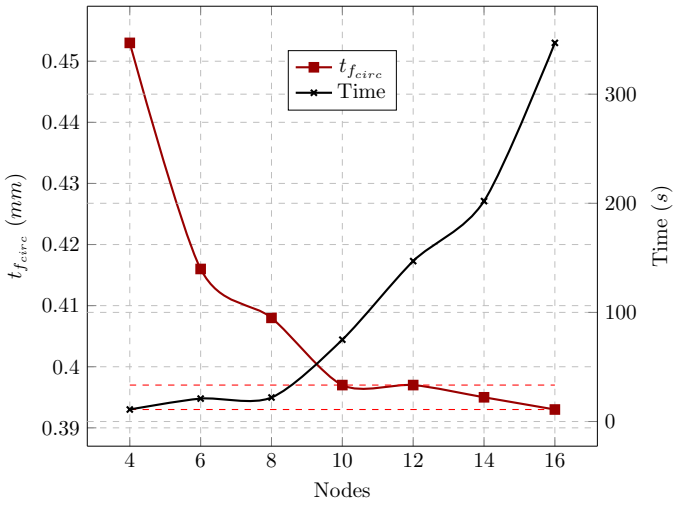


Figure 14: Convergence of $t_{f_{circ}}$ and variation on the time of simulation with an increase in the number discretization nodes of the features.

5. Conclusions and future work

We have presented a new strategy to handle sets of geometric constraints through prismatic polyhedra. Each set of constraints derives naturally from an ISO GPS specification or a contact specification. The approach presented is feature based and all the dependencies between the 3 translations and 3 rotations are taken into account simultaneously, implying that only one simulation is needed to verify the satisfaction of a functional requirement.

We compared it with a previous method that adds cap half-spaces to virtually constraint the unbounded displacements and generates bounded sets in $\mathbb{R}^6$. The new strategy is based on the decomposition of geometric and contact polyhedra into the sum of a polytope (the bounded part of the polyhedron or wrench space in screws theory) and polyhedral cones (the unbounded part of the polyhedron or twist space in screws theory). While the intersection between operands continues to be done in the same way, the prismatic polyhedra method suggests using the decomposition property to sum the projection of the underlying polytopes in the common subspace. There we perform a very efficient reduction model as polyhedra contain exclusively facets of mechanical

interest: not only the polyhedra have the minimum number of facets but also the subspace dimension is as small as possible. The operands decomposition and the identification of the subspace are proposed to be done through a kinematic analysis using screws theory.

A case study was presented to illustrate the methods and to compare the time they required to complete the simulation. The example consists of a spectrometer made up of several over-constrained joints. The result of the simulation following the cap-based method showed that bounding the DOFs with capped facets increases the complexity of the model due to the kinematic propagation of DOFs, at the end more than 98.7% of the calculated facets where capped, meaning that 99.9% of the time was used to calculate meaningless information. The other approach avoids this situation, allowing to decrease the computational time.

While using the polyhedral algorithm, verifying the inclusion of the final result inside the target can be done in just two steps: i) check the kinematic compliance of the mechanical system and ii) check the functional tolerance compliance. This differentiation between kinematic and tolerance compliance allows the designer to know, not only if the functional requirement is met but also if it could be satisfied and how much it could be improved.

Since the method presented is feature-based, the result is highly impacted by the discretization of the features of the mechanism; thanks to the reduction in time complexity of the method, we are able now of doing a convergence analysis that will tell us which is the required amount of nodes to ensure that the results of the analysis are not affected by imprecisions due to the number of nodes used. For the case study presented, Figure 14 shows that the result converges to $0.395 \pm 0.002\ mm$ and using more than 10 nodes will affect the calculation time but it will not have a significant impact on the result. It is worth mentioning that the convergence analysis was made for the study case and its not possible to generalize the results, meaning that another system may need more or less discretization nodes to give a good result. Additionally, the location of the points has an impact on the result, further work is needed to

be able to identify how to adapt their distribution in order to get a result with a satisfying precision without having a large amount of nodes. In that case, it will be necessary to take into account not only the geometry of the surfaces but also the functional condition.

The reduction in both computation time and polyhedron size - hence in memory footprint - discussed along this paper opens exciting directions for future work related to tolerance synthesis by optimizing the individual tolerances of the components according to manufacturing and quality costs and taking into account the dependencies between the tolerances of all the mechanism. We plan to modify the tolerances of the contacts and the surfaces of the mechanism through a stochastic process until a set of tolerances that satisfy the requirements of functionality, cost, and quality is found. In the future, we also plan to identify the most contributing tolerance zones or clearances by tracing the polytope vertices during the process of intersections and sums; this is due to the fact that polytopes offer the possibility to correlate the topology of the final result to its basic operands deriving directly from the set of constraints.

Until now several assumptions are made while defining the model (no form defect in surfaces, no local strain due to the contact and no deformable parts). Future work is required to enrich the model as much as possible by taking into account form defects, flexible parts and dynamic interaction between parts. This work has been started by Gouyou et al. [32], who studied the tolerance analysis considering flexible assemblies in a flange.

References

[1] P. Ziegler and S. Wartzack. Sensitivity analysis of features in tolerancing based on constraint function level sets. *Reliability Engineering & System Safety*, 134:324 – 333, 2015.

[2] B. Anselmetti. *CLIC: A Method for Geometrical Specification of Products*, pages 207–239. John Wiley & Sons, Inc., 2013.

[3] L. Lindkvist and R. Söderberg. Computer-aided tolerance chain and stability analysis. *Journal of Engineering Design*, 14(1):17–39, 2003.

[4] A. Desrochers, W. Ghie, and L. Laperriere. Application of a unified jacobian-torsor model for tolerance analysis. *Journal of Computing and Information Science in Engineering(Transactions of the ASME)*, 3(1):2–14, 2003.

[5] A. Desrochers and A. Clément. A dimensioning and tolerancing assistance model for cad/cam systems. *The International Journal of Advanced Manufacturing Technology*, 9(6):352–361, 1994.

[6] A. Clément, A. Rivière, P. Serré, and C. Valade. *The TTRSs : 13 Constraints for Dimensioning and Tolerancing*, pages 122–131. Springer US, Boston, MA, 1998.

[7] P. Clozel and P. Rance. Mecamaster: a tool for assembly simulation from early design, industrial approach. *Geometric Tolerancing of Products*, pages 241–273, 2010.

[8] L. Homri, D. Teissandier, and A. Ballu. Tolerance analysis by polytopes: Taking into account degrees of freedom with cap half-spaces. *Computer-Aided Design*, 62:112 – 130, 2015.

[9] M. Giordano, D. Duret, S. Tichadou, and R. Arrieux. Clearance space in volumic dimensioning. *CIRP Annals - Manufacturing Technology*, 41(1):565 – 568, 1992.

[10] D. Teissandier, V. Delos, and Y. Couétard. Operations on polytopes: application to tolerance analysis. In *Global Consistency of Tolerances*, pages 425–433, Enschede (Netherlands), 1999. Kluwer academic.

[11] J.K. Davidson, A. Mujezinovic, and J.J. Shah. A new mathematical model for geometric tolerances as applied to round faces. *Journal of mechanical design*, 124(4):609–622, 2002.

[12] A. Dumas, J.Y. Dantan, and Gayton N. Impact of a behavior model linearization strategy on the tolerance analysis of over-constrained mechanisms. *Computer-Aided Design*, 62:152 – 163, 2015.

[13] J.Y. Dantan and A.J. Qureshi. Worst-case and statistical tolerance analysis based on quantified constraint satisfaction problems and monte carlo simulation. *Computer-Aided Design*, 41(1):1 – 12, 2009.

[14] P. Beaucaire, N. Gayton, E. Duc, and J.Y. Dantan. Statistical tolerance analysis of over-constrained mechanisms with gaps using system reliability methods. *Computer-Aided Design*, 45(12):1547 – 1555, 2013.

[15] G. Ameta, S. Serge, and M. Giordano. Comparison of spatial math models for tolerance analysis: tolerance-maps, deviation domain, and ttrs. *Journal of Computing and Information Science in Engineering*, 11(2):021004, 2011.

[16] M. Mansuy, M. Giordano, and J.K. Davidson. Comparison of two similar mathematical models for tolerance analysis: T-map and deviation domain. *Journal of Mechanical Design*, 135(10):101008, 2013.

[17] J.J. Shah, G. Ameta, Z. Shen, and J.K. Davidson. Navigating the tolerance analysis maze. *Computer-Aided Design and Applications*, 4(5):705–718, 2007.

[18] S. Arroyave-Tobón, D. Teissandier, and V. Delos. Applying screw theory for summing sets of constraints in geometric tolerancing. *Mechanism and Machine Theory*, 112:255 – 271, 2017.

[19] V. Delos, S. Arroyave-Tobón, and D. Teissandier. Introducing a Projection-Based Method to Compare Three Approaches Computing the Accumulation of Geometric Variations. volume 1A: 38th Computers and Information in Engineering Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 08 2018.

[20] P. Bourdet, L. Mathieu, C. Lartigue, and A. Ballu. The concept of the small displacement torsor in metrology. *Series on advances in Mathematics for Applied Sciences*, 40:110–122, 1996.

[21] S. Arroyave-Tobón, D. Teissandier, and V. Delos. Tolerance analysis with polytopes in HV-description. In *Proceedings of ASME IDETC-CIE*, Charlotte, NC, USA, August 2016.

[22] V. Delos, S. Arroyave-Tobón, and D. Teissandier. Model reduction in geometric tolerancing by polytopes. *Computer-Aided Design*, March 2018.

[23] G.M. Ziegler. *Lectures on polytopes*, volume 152. Springer Science & Business Media, 1995.

[24] J.Y. Dantan, L. Mathieu, A. Ballu, and P. Martin. Tolerance synthesis: quantifier notion and virtual boundary. *Computer-Aided Design*, 37(2):231 – 240, 2005.

[25] W. Cheney and A.A. Goldstein. Proximity maps for convex sets. *Proceedings of the American Mathematical Society*, 10(3):448–450, 1959.

[26] S. Boyd and J. Dattorro. Alternating projections. *EE392o, Stanford University*, 2003.

[27] D. Teissandier, V. Delos, and S.C. García. Simulating the Respect of a Functional Condition in a Mechanical System with Mobilities. *Procedia CIRP*, 92:106–111, January 2020.

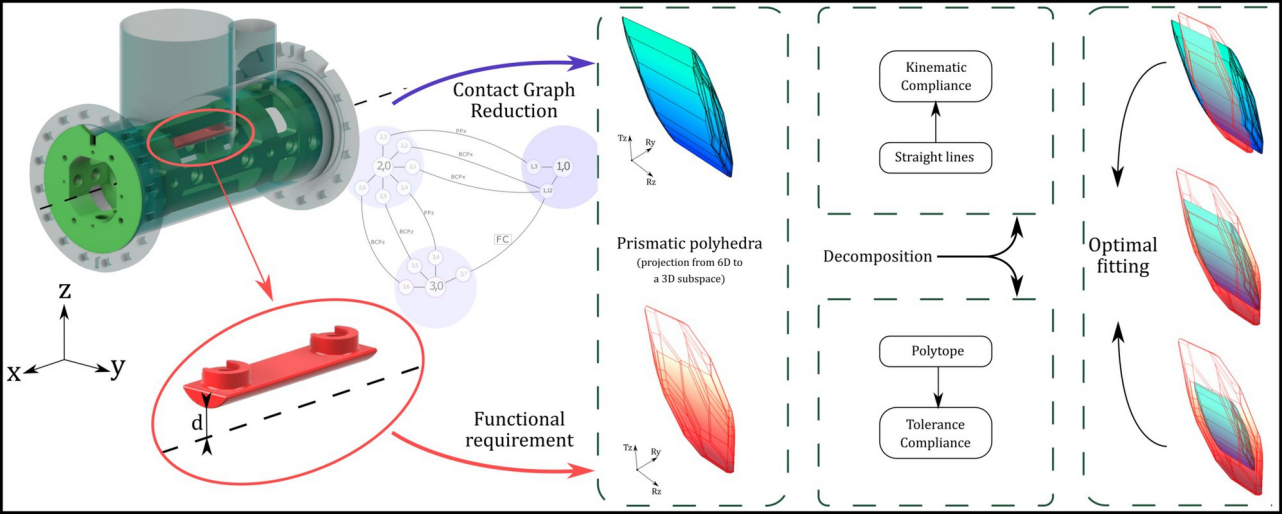
[28] ISO 1101, Geometrical product specifications (GPS)— Geometrical tolerancing — Tolerances of form, orientation, location and run-out. 2017.

[29] ISO 5459 geometrical product specifications (GPS) — Geometrical tolerancing — Datums and datum systems. 2011.

[30] R. Schneider. *Convex Bodies: The Brunn–Minkowski Theory —— Minkowski addition*, volume 10.1017/CBO9780511526282. 1993.

[31] V. Delos and D. Teissandier. PolitoCAT and politopix. http://i2m.u-bordeaux.fr/politopix.html, 2017.

[32] D. Gouyou, Y. Ledoux, D. Teissandier, and V. Delos. Tolerance analysis of overconstrained and flexible assemblies by polytopes and finite element computations: application to a flange. *Research in Engineering Design*, 29(1):55–66, April 2017.


Contact Graph Reduction
Functional requirement
Prismatic polyhedra
(projection from 6D to
a 3D subspace)
Decomposition
Kinematic Compliance
Straight lines
Polytope
Tolerance Compliance
Optimal fitting
FC
z
x
y
d
Tz
Ry
Rz