Polyhedral models reduction in geometric tolerance analysis

Santiago Arroyave-Tobón

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGÉNIEUR
SPÉCIALITÉ: MÉCANIQUE ET INGÉNIERIE

Par Santiago ARROYAVE-TOBÓN

POLYHEDRAL MODELS REDUCTION IN GEOMETRIC TOLERANCE ANALYSIS

Sous la direction de : Denis TEISSANDIER
Co-encadrant : Vincent DELOS

Soutenue le 10 novembre 2017

Membres du jury :

M. LINARES, Jean-Marc	Professeur des Universités, Aix-Marseille Université	Président
M. DANTAN, Jean-Yves	Professeur des Universités, ENSAM Metz	Rapporteur
M. ANWER, Nabil	Professeur des Universités, Université Paris Sud	Rapporteur
M. TEISSANDIER, Denis	Professeur des Universités, Université de Bordeaux	Examinateur
M. DELOS, Vincent	Ingénieur Recherche, CNRS	Examinateur
M. SENGER, Gérald	Ingénieur R&D, Safran Helicopter Engines	Invité

Réduction de modèles polyédriques pour l'analyse de tolérances géométriques

Résumé : L'analyse de tolérances par des ensembles de contraintes repose sur la détermination de l'accumulation de variations géométriques par des sommes et intersections d'ensembles opérandes 6d. Les degrés de liberté des liaisons et les degrés d'invariance des surfaces génèrent des opérandes non-bornés (polyèdres), posant des problèmes de simulation. En 2014, L. Homri a proposé une méthode pour résoudre ce problème, consistant à ajouter des limites artificielles (contraintes bouchon) sur les déplacements non-bornés. Même si cette méthode permet la manipulation d'objets bornés (polytopes), les contraintes bouchon augmentent la complexité des simulations. En réponse à cette difficulté, une méthode dérivée est proposée dans cette thèse. Cette méthode consiste à tracer et simplifier les contraintes bouchon au travers des opérations. Puis une seconde stratégie basée sur la décomposition d'un polyèdre en une somme d'un polytope et de lignes droites (associées aux déplacements non-bornés). Cette stratégie consiste à simuler d'une part les sommes de droites, et d'autre part, à déterminer la somme de polytopes dans un sous-espace de dimension inférieur à 6. Ces trois stratégies sont comparées au travers d'une application industrielle. Cela montre que la traçabilité des contraintes bouchons est un aspect fondamental pour contrôler leur propagation et pour réduire le temps de calcul des simulations. Toutefois, cette méthode exige encore de déterminer les limites des déplacements non-bornés. La deuxième méthode, adaptant systématiquement la dimension de l'espace de calcul, elle permet de diminuer davantage le temps de calcul. Ce travail permet d'envisager la mise en œuvre de cette méthode selon des formulations statistiques avec la prise en compte des défauts de forme des surfaces.

Mots clés : Analyse de Tolérances, Géométrie Algorithmique, Polyèdre, Somme de Minkowski, Degrés de Liberté, Screws.

Polyhedral models reduction in geometric tolerance analysis

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 degrees of freedom (DOF) of joints generate unbounded sets (i.e. polyhedra) which are difficult to deal with. L. Homri presented in 2014 a solution based on the setting of fictitious limits (called cap constraints) to each DOF to obtain bounded 6D sets (i.e. polytopes). These additional constraints, however, increase the complexity of the models, and therefore, of the computations. In response to this situation, we defined a derived strategy to control the effects of the propagation of the fictitious limits by tracing and simplifying the generated, new cap constraints. We proposed a second strategy based on the decomposition of polyhedra into the sum of a polytope and a set of straight lines. The strategy consists in isolating the straight lines (associated to the DOF) and summing the polytopes in the smallest sub-space. After solving an industrial case, we concluded that tracing caps constraints during the operations allows reducing the models complexity and, consequently, the computational time; however, it still involves working in 6d even in cases where this is not necessary. In contrast, the strategy based on the operands decomposition is more efficient due to the dimension reduction. This study allowed us to conclude that the management of mechanisms' mobility is a crucial aspect in tolerance simulations. The gain on efficiency resulting from the developed strategies opens up the possibility for doing statistical treatment of tolerances and tolerance synthesis.

Keywords : Tolerance Analysis, Computational Geometry, Polyhedron, Minkowski Sum, Degrees of Freedom, Screws.

I2M

Institut de Mécanique et d'Ingénierie, CNRS UMR 5295, Bordeaux

Acknowledgements

Aunque sea solo los agradecimientos, algo tenía que quedar en español en mi tesis.

Un enorme agradecimiento para Denis y Vincent por su apoyo, por su confianza y sobre todo por su amistad. Fue un gran placer y un gran honor trabajar con ustedes!

Agradezco a los miembros del jurado por haber aceptado la invitación a evaluar mi trabajo. Agradezco a Jean-Yves Dantan, profesor de *l'ENSAM de Metz*, y a Nabil Anwer, profesor de la *Université Paris Sud*, por la evaluación de esta memoria y por sus comentarios. Agradezco a Jean-Marc Linares, profesor de la *IUT d'Aix-Marseille*, por su ayuda, por los intercambios sobre mi trabajo y por haber presidido el jurado. Un agradecimiento también para Gerald Sanger, ingeniero I+D de la compañía *Safran Helicopter Engines*, por aceptar la invitación a conformar el jurado.

Agradezco a Jean-Piere Nadeau, profesor emerito de *l'ENSAM de Bordeaux*, e igualmente a Ricardo Mejía y a Gilberto Osorio, profesores de la Universidad EAFIT, por haber permitido mi pasantía en Burdeos, que finalmente fue el punto de partida para mi doctorado.

Al combo los caravanos: Ricardo, Ana, Ulises, Jorge, Rachel, Doriane, Lina, David, Xingyu; y a mis parceros Aitor y Tonino: gracias!

Agradezco a los profes y personal del laboratorio I2M por su apoyo. Béatrice Desoudin, Jean-Pierre Lariviere, Yann Ledoux, Alex Ballu, Michel Mesnard: muchas gracias!

A mi padres que siempre estuvieron a mi lado (aunque fuera a la distancia) y a mi hermanito (que incluso participó directamente en la elaboración de esta memoria corrigiendo el inglés): muchas gracias. A ustedes tres dedico esta tesis.

Contents

List of Figures

List of Tables

Résumé étendu

Problématique générale

Toute pièce fabriquée est soumise à des variations géométriques inhérentes aux procédés de fabrication. La prise en compte de ces variations est un aspect crucial du cycle de vie du produit, notamment en conception. L'analyse des tolérances géométriques consiste à simuler le comportement d'un système mécanique en fonction de ces variations. L'objectif est de vérifier la conformité du système au regard des exigences fonctionnelles caractérisant le fonctionnement attendu du système. Cette simulation impose de prendre en compte les spécifications géométriques des pièces constitutives et les spécifications entre les pièces potentiellement en contact.

Nombreux sont les travaux publiés sur la représentation et la modélisation des écarts et variations géométriques, et de leur cumul au niveau des assemblages. Le chapitre 1 de ce mémoire, *«Review of geometric tolerancing approaches»*, propose une classification de ces travaux en selon : leur modèle physique, leur modèle géométrique, leur modèle de défauts, leur modèle de comportement, des outils d'analyse (probabiliste ou pire des cas) et du principe de résolution (analyse ou synthèse). Cette classification permet de statuer sur le positionnement de ce travail de recherche par rapport aux autres travaux dans le domaine du tolérancement.

Méthode de tolérancement par des polytopes (méthode de référence)

Parmi les approches d'analyse de tolérances, celles reposant sur la manipulation d'ensembles de contraintes géométriques présentent l'avantage d'adresser indifféremment la modélisation de la propagation de variations géométriques dans des assemblages isostatiques et hyperstatiques. Les ensembles de contraintes proviennent de la restriction de déplacements induits par des spécifications géométriques, par des spécifications de contact ou bien par des exigences fonctionnelles d'un système mécanique. Ces contraintes sont générées en discrétisant les géométries nominales manipulées en Conception Assistée par Ordinateur (CAO). Chaque ensemble est conformé en un polyèdre résultant d'une intersection de demi-espaces dans un espace affine de dimension 6. Cette approche permet de déterminer la position relative entre deux surfaces quelconques d'un système mécanique par des opérations sur des polytopes : somme de Minkowski et intersection. Le respect d'une exigence fonctionnelle est caractérisé par l'inclusion d'un polytope calculé dans un polytope fonctionnel.

Les degrés de liberté des liaisons et les degrés d'invariance des surfaces génèrent des opérandes non-bornés (polyèdres), posant des problèmes de simulation. En 2014, L. Homri a proposé une méthode pour résoudre ce problème, consistant à limiter les degrés d'invariance des surfaces et les degrés de liberté des contacts par des limites fictives (contraintes bouchon).

Même si cette méthode permet la manipulation uniquement d'objets bornés (polytopes) pendant les simulations, les contraintes bouchon augmentent la complexité de la topologie des opérandes. Ceci a un fort impact sur le calcul des sommes de Minkowski. Cette opération, algorithmiquement complexe et très gourmande en temps de calcul, est très sensible à la complexité des polytopes opérandes.

Dans le chapitre 2, *«Geometric tolerancing with 6D polytopes»*, la méthode de tolérancement par des polytopes est détaillée, ainsi que la stratégie de simulation reposant sur l'adjonction de contraintes bouchon. L'algorithme de troncature, qui permet d'exploiter la HV-description des polytopes, est présenté. Une étude de cas portant sur un assemblage hyperstatique (système de freinage) a été utilisée pour illustrer la démarche de référence.

Cet exemple a permis de mettre en évidence le problème de l'explosion combinatoire issue de la propagation de facettes bouchons. Le polytope résultant de la simulation, représentant le cumul de tous les défauts de la chaîne de cotes, est constitué de 99,99% de facettes bouchons parmi l'ensemble des facettes. Ces facettes représentent les limites fictives sur des déplacements non bornés et, par conséquent, n'ont pas d'intéret du point de vue du tolérancement et accroissent la difficulté de l'interprétation des résultats de simulation.

En réponse à cette problématique, cette thèse a comme objectif de développer des stratégies pour maîtriser cette explosion combinatoire en réduisant la complexité et temps de calcul des simulations.

Méthode de réduction des contraintes bouchons

Une méthode dérivée des travaux de L. Homri est proposée dans le chapitre 3, *«Controlling the effects of DOF propagation»* , afin de contrôler la propagation des déplacements théoriquement illimités dont les limites sont fictives. Cette méthode se base sur une définition formelle d'une contrainte bouchon et sur des mécanismes pour tracer ces contraintes à travers les sommes et les intersections. Parmi ces opérations, c'est pendant la somme que de nouvelles contraintes sont générées. Un théorème est démontré pour décrire ce phénomène de propagation: :

Théorème de propagation des bouchons : *soient* Γ_1, Γ_2 *deux polyèdres de* $\mathbb{R}^n$ *et* Γ'_1, Γ'_2 *leurs polytopes bornés respectifs. Soit* $F_{\Gamma'}$ *une facette de* $\Gamma'_1 \oplus \Gamma'_2$ *et* $F_{\Gamma'_1} + F_{\Gamma'_2}$ *sa décomposition en faces de* Γ'_1 *et* Γ'_2*.* $F_{\Gamma'}$ *a une facette correspondente dans* F_Γ *qui appartient à* $\Gamma_1 \oplus \Gamma_2$*, si et soulement si les représentations combinatorires* $\mathcal{CR}(F_{\Gamma'_1})$ *et* $\mathcal{CR}(F_{\Gamma'_2})$ *ne contiennent pas des hyperplans bouchon.*

Ce nouveau formalisme permet, après chaque opération, d'effectuer une simplification du polytope résultant afin de réduire systématiquement au minimum le nombre de contraintes bouchon, tout en assurant que les contraintes non-bouchons sont conservées.

L'efficacité de la nouvelle approche est démontrée sur le cas d'étude du système de freinage à travers une réduction considérable du temps de calcul (99,37%). Cela montre que la traçabilité des contraintes bouchons est un aspect fondamental pour contrôler leur propagation et pour réduire le temps de calcul des simulations.

Toutefois, cette méthode exige encore le calcul des opérations dans R6, même si dans la plupart de cas cela n'est pas nécessaire (à cause de degrés de liberté ou des degrés d'invariance, la position relative entre deux surfaces peut être décrite avec un nombre de paramètres inférieur à

6). En conséquence, faire les opérations avec de polytopes dans R6 implique le calcul de limites des déplacements théoriquement illimités.

Méthode de décomposition cinématique des opérandes

Une seconde méthode est introduite dans le chapitre 4, *«Kinematic decomposition of geometric constraints»*. Basée dans la définition de polyèdre prismatique, cette méthode repose sur la décomposition des polyèdres opérandes en une somme d'un polytope (sous-ensemble borné) et de lignes droites (sous-ensembles non-bornés). Dans le cadre du tolérancement, le sous-ensemble non-borné caractérise les degrés de liberté de l'élément tolérancé ou bien de la liaison et le sous-ensemble non-borné caractérise les contraintes géométriques. Cette décomposition peut être réalisée par une analyse cinématique. Étant donné les similitudes de leurs définitions mathématiques avec les contraintes géométriques, la méthode des "screws" est mise en œuvre. Ce principe de décomposition est présenté pour les éléments tolérancés et les liaisons les plus couramment utilisées en tolérancement géométrique. Au lieu de sommer des polyèdres de dimension 6, cette stratégie consiste à simuler d'une part les sommes de déplacements non bornés par des sommes de droites, et d'autre part, à déterminer la somme des déplacements bornés par des sommes de polytopes de dimension inférieure à 6. Un théorème a été démontré pour formaliser cette nouvelle stratégie de somme :

Au lieu de sommer des polyèdres de dimension 6, cette stratégie consiste à simuler d'une part les sommes de déplacements non bornés par des sommes de droites, et d'autre part, à déterminer la somme des déplacements bornés par des sommes de polytopes de dimension inférieure à 6. Un théorème a été démontré pour formaliser cette nouvelle stratégie de somme :

Théorème de somme de polyèdres prismatiques : *soient Γ_1 et Γ_2 deux polyèdres prismatiques de $\mathbb{R}^n$ décomposables en une somme d'un polytope P_1, P_2 plus une somme de droites Δ_i :*

$$\begin{gathered}
\Gamma_1 = P_1 \oplus \sum_{i=1}^{k} \Delta_i, \quad P_1 \subset H_{P_1} = \bigcap_{i=1}^{k} H_i \\
\Gamma_2 = P_2 \oplus \sum_{i=k+1}^{l} \Delta_i, \quad P_2 \subset H_{P_2} = \bigcap_{i=k+1}^{l} H_i \\
\text{avec } \Delta_i \perp H_i \ \forall i \in \{1, ..., l\}
\end{gathered}$$

La somme $\Gamma_1 \oplus \Gamma_2$ la somme peut être calculée comment la somme des projections des polytopes sous-jacents dans le sous-espace commun aux opérandes $H_{P_1} \cap H_{P_2}$ plus leurs respectives droites :

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

(π_H : projection orthogonale sur l'espace H)

Les droites Δ_i représentent les déplacements illimités (degrés d'invariance ou degrés de liberté) et leur somme définissent le sous-espace des mobilités (*twist-space*). Selon la théorie des mécanismes reposant sur l'hypothèse de liaisons parfaites, ce sous-espace est orthogonal au sous-espace des efforts H_{P_1}, H_{P_2} (*wrench-space*). Le sous-espace commun $H_{P_1} \cap H_{P_2}$, correspond aux directions de déplacements le long desquelles des efforts peuvent être transmis par les deux liaisons. Finalement, c'est selon ces directions de mouvement que les variations géométriques peuvent être maîtrisées.

Dans ce contexte, à partir d'analyses cinématiques des éléments tolérancés, il est possible d'identifier le sous-espace affine de $\mathbb{R}^6$ dans lequel toutes les contraintes manipulées proviennent uniquement de déplacements bornés. Cela présente l'avantage de calculer uniquement les facettes significatives selon le problème de tolérancement associé (en évitant le calcul de facettes sur d'axes de déplacements non bornés). De plus, cela permet de réduire la complexité des opérations en réduisant la dimension de l'espace de calcul (en réalisant des opérations dans un espace de dimension inférieure à 6).

En appliquant cette méthode au cas d'étude du système de freinage, une diminution significative du temps de calcul par rapport à la méthode de référence (99,997% de réduction) et à la méthode de réduction des contraintes bouchons (99,539% de réduction) a été démontrée. Ceci est du à l'adaptation systématique de la dimension de l'espace de calcul qui permet de manipuler uniquement les facettes non-bouchon.

Conclusions et perspectives

Ce travail a montré que la gestion des mobilités des systèmes mécaniques est un aspect fondamental pour l'analyse de tolérances par des ensembles de contraintes.

Les stratégies développées dans cette thèse ont permis d'améliorer l'efficacité de l'approche de tolérancement par des polytopes en maîtrisant l'explosion combinatoire de facettes pendant les simulations. Ceci permet d'envisager la mise en œuvre de cette méthode selon des formulations statistiques où les défauts des pièces sont simulés conformément aux lois de distribution de la production avec la prise en compte des défauts de forme des surfaces.

Introduction

Context

Geometric uncertainties influence product assembly, functioning and aesthetics. In response to this, geometric product specification (GPS) has emerged as an international and unequivocal means of communication by which the designer, the manufacturer and the metrologist can exchange functional information about a product (Petit, 2004). These specifications, formalized under ISO-1101 (2012), ISO-5459 (2011), ISO-8015 (2011), ISO-14405-1 (2011); ISO-14405-2 (2011) and ASME-Y14.5 (2009) standards, provide a way to define admissible manufacturing deviations according to the design requirements (Mounaud et al., 2011).

In general, the choice of a tolerancing scheme is not made lightly, and in addition, it impacts all the stages of the product life cycle (Dumas, 2014); hence the highly active research in tolerance management. The main problem, which is still unresolved, is how to decompose the assembly design requirements into tolerances on individual parts, given the available manufacturing capabilities (Fleming, 1987; Lindkvist and Söderberg, 2003; Petit, 2004).

This problem is usually addressed in three steps (Dantan, 2000). First, the geometric deviations are modelled. Second, design requirements are mathematically described. Finally, the mathematical constraints of the assembly are transferred to constraints in the deviations of the constituent parts. This process is illustrated in Figure 1.

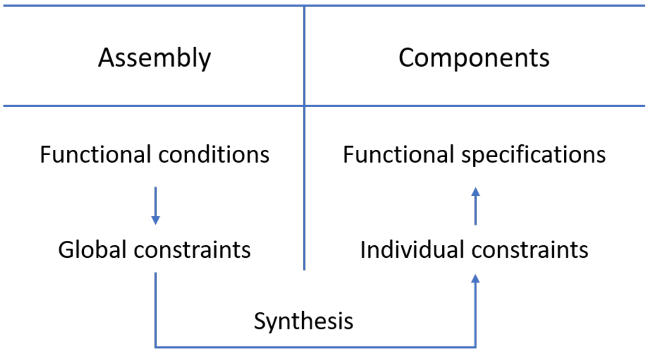


Figure 1: Geometric tolerancing process (adapted from (Dantan, 2000)).

Ideally, this transfer should be performed in a top-down fashion (tolerance synthesis). However, due to its complexity, the problem is usually solved in the opposite direction: the designer first identifies the influential parts along the tolerance chain, chooses tolerances and then verifies that the requirements are satisfied (tolerance analysis) (Anselmetti et al., 2010).

Different methods of tolerance simulation have been developed. Most of them consider the effects of 3D geometric variations in rotation or translation along a given direction (Lindkvist and Söderberg, 2003; Desrochers et al., 2003; Clozel and Rance, 2010; Anselmetti, 2013), limiting their application only to isostatic mechanisms. In addition, this strategy implies that more than one simulation has to be carried out, considering different displacements directions, in order to validate the fulfilment of a design requirement (Dantan, 2000; Homri et al., 2015). In contrast, methods based on operations of geometric constraints can cope with these issues. In an abstract deviation space, these methods represent the movement constraints imposed by the tolerance zone of each toleranced feature and joint (Giordano et al., 1992; Teissandier et al., 1999; Davidson et al., 2002; Beaucaire et al., 2013). The obtained set of constraints represents all possible deviations of a feature or joint within its tolerance zone. By combining these sets, deviation propagation at the assembly level can be simulated.

Depending on the way the sets of constraints are defined and represented, different operations are required. They can be defined either, by means of their frontiers, or by means of representative samples of internal points. In the first case, Minkowski sums and intersection of polytopes are able to simulate deviation propagation. In the second case, linear optimization combined with reliability calculations can be used.

This thesis, in continuity with previous work carried out at the I2M laboratory, is focused on the management of sets of constraints by means of their frontiers.

Contribution of this thesis

Although the method based on sets of constraints is sufficiently robust to treat most mechanical design cases (including over-constrained assemblies), the complexity of the Minkowski sums makes it time-consuming and involves elaborate computing processes.

We found that such complexity, in the context of geometric tolerancing, is correlated to the way in which the degrees of freedom (DOFs) of 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 general, however, the sets belong to spaces of different dimensions requiring special treatment.

One solution to this problem is presented by Homri (2014). The author proposes to compute the sums in a 6-dimensional space by introducing some additional constraints, called caps, to the displacements related to the DOFs. We found that this solution entails an increase in model complexity caused by the propagation of the DOFs along the tolerance chains. This complexity worsens after each sum until it becomes far too significant and consumes most of the computational resources.

In response to this challenge, this thesis proposes different strategies to handle sets of geometric constraints with which to face the problems resulting from unconstrained displacements during tolerance propagation simulations:

- The first strategy, which follows the approach of Homri (2014), proposes to tag the cap constraints when the operand sets are created. This can be easily done as the type of surface from which the constraints are derived is known. The strategy is based on a characterization of the way in which the cap constraints spread during the computations. This allows the caps to be traced during the different operations throughout the tolerance simulations. After each operation, the calculated set can be simplified, re-establishing a minimum set of cap constraints to control the increase in complexity.

- The second strategy proposes to decompose each set of geometric constraints into a bounded subset (representing the limits imposed by the tolerance zones) and an unbounded subset (representing the DOFs). When summing two operand sets, only the bounded subsets can be considered, thus 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 are bounded. This sub-space is characterized by the displacements that define the relative position of the two features from which the operand sets derive. The fact of calculating the sum in this sub-space means that the complexity of the operands, and consequently the computational time, can be significantly reduced. We propose to carry out this decomposition and calculate this sub-space by means of kinematic analysis using screw systems.

After testing both strategies and comparing them by solving an industrial case, we concluded that the traceability of constraints during the operations allows reducing the models complexity and consequently of the computational time. This strategy is straightforward to implement over a Minkowski sum algorithm, but it still involves working in 6D, even in cases where this is not necessary. On the other hand, the strategy based on the sum of the operand projection is more efficient because the dimensions of the calculation space are reduced.

The whole list of scientific communications derived from this thesis is presented in Annex A.

Document review

This document is divided into five main parts:

- In Chapter 1, the existing geometric tolerancing techniques are reviewed. They are categorized, first according to the way they model the deviations at the part level and their propagation at the assembly level, and second according to the strategy adopted to perform computations.
- In Chapter 2, the method for geometric tolerancing based on 6D polytopes is summarized. It describes how restrictions derived from the tolerance zone of toleranced feature or from the gap of a mechanical joint can be represented by 6D polyhedra. These polyhedra are then bounded into polytopes by introducing cap constraints. In the last part of this chapter, the case study used throughout this thesis is presented and solved. Some aspects that have potential for improving the current method are highlighted and discussed.
- In Chapter 3, the mathematical definition of a cap constraint is presented. The chapter also describes how the cap constraints spread during sums and intersections. A theorem and an algorithm formalize the strategy for tolerance analysis with cap control propagation. The strengths and weaknesses of this strategy are discussed around the same case study presented in Chapter 2.

- Chapter 4 details how to decompose geometric sets of constraints using kinematic analyses. This decomposition is presented in a general way, with the most common cases in geometric tolerancing being illustrated. This decomposition leads to more efficient ways to simulate interactions between sets of geometric constraints. A method for summing decomposed sets of constraints, supported by a theorem, is presented and tested with the case study.

- Finally, in the last chapter, a general discussion is presented with prospects for further research.

Notations

The field of real numbers is denoted by $\mathbb{R}$. Vector spaces defined on $\mathbb{R}$ are denoted $\mathbb{R}^n$, where n is an integer representing the dimension of the space. Vectors are denoted in boldface type, such as $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$. The zero vector is denoted by $\mathbf{0}$. Scalars are represented in by Greek letters, such as α, β, γ. Integers are denoted with latin letters: a, b, c. Other notations used throughout this document are:

H	hyperplane of $\mathbb{R}^n$.
$\bar{H}^+$	positive closed half-space associated to H.
P	polytope of $\mathbb{R}^n$.
$\mathcal{H}_P$	set of half-spaces defining the polytope P.
$\mathcal{V}_P$	set of vertices defining the polytope P.
$\Gamma_{a,b/c,d}$	polyhedron of $\mathbb{R}^n$ describing the relative position of the surface b of the part a with respect to the surface d of the part c.
$\Gamma'_{a,b/c,d}$	capped polytope associated to polyhedron $\Gamma_{a,b/c,d}$.
$P_1 \oplus P_2$	Minkowski sum of polytopes P_1 and P_1.
$P_1 \tilde{+} P_2$	Minkowski sum of polytopes P_1 and P_1 followed with a cap removal.
$C_D(\boldsymbol{v})$	dual cone of vertex $\boldsymbol{v}$.
$\mathcal{N}(P)$	normal fan of polytope P.
Δ	straight line in $\mathbb{R}^n$.
$\mathcal{W}_{a,b/c,d}$	wrench describing a given restricted movement between the surface b of the part a with respect to the surface d of the part c.
$\mathcal{T}_{a,b/c,d}$	twist describing a given mobility between the surface b of the part a with respect to the surface d of the part c.
$\hat{\boldsymbol{W}}$	wrench-matrix.
$\hat{\boldsymbol{T}}$	twist-matrix.
π_H	orthogonal projection in the hyperplane H.

Chapter 1

Review of geometric tolerancing approaches

In this chapter, we review the existing approaches for simulating geometric variations in mechanical systems. The differences between them lie in the assumptions they make to reproduce variations at the part level and their propagation at the assembly level. Different phenomena interact at different stages of the products life cycle to produce greater geometric deviations (see for example Figure 1.1). The complexity of considering all the real phenomena and their interaction renders assumptions imperative. We review and categorize these assumptions throughout the different stages comprised in the definition of a geometric tolerancing approach. This taxonomy, summarized in Figure 1.2, is developed in the following sections.

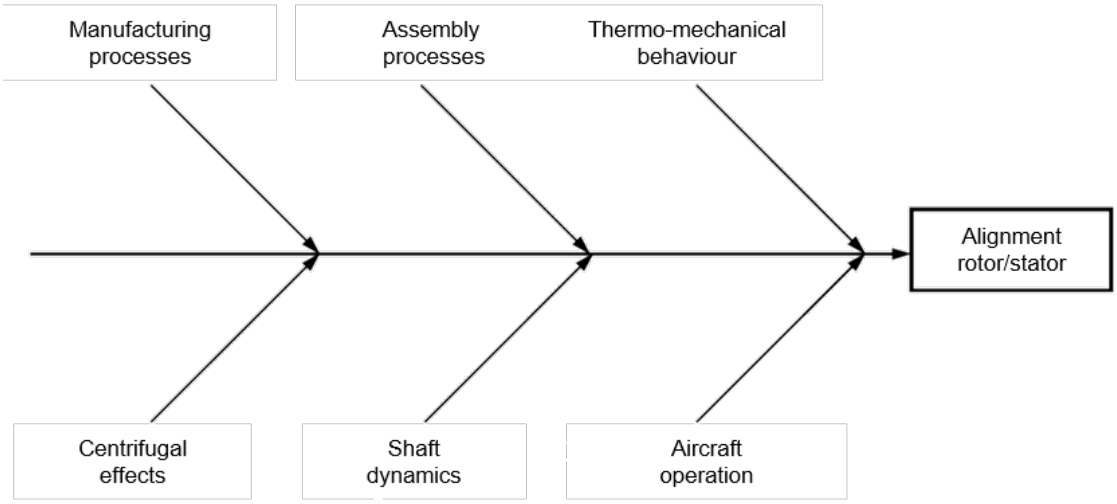


Figure 1.1: Phenomena causing geometric deviations in a turboshaft engine (Pierre, 2011).

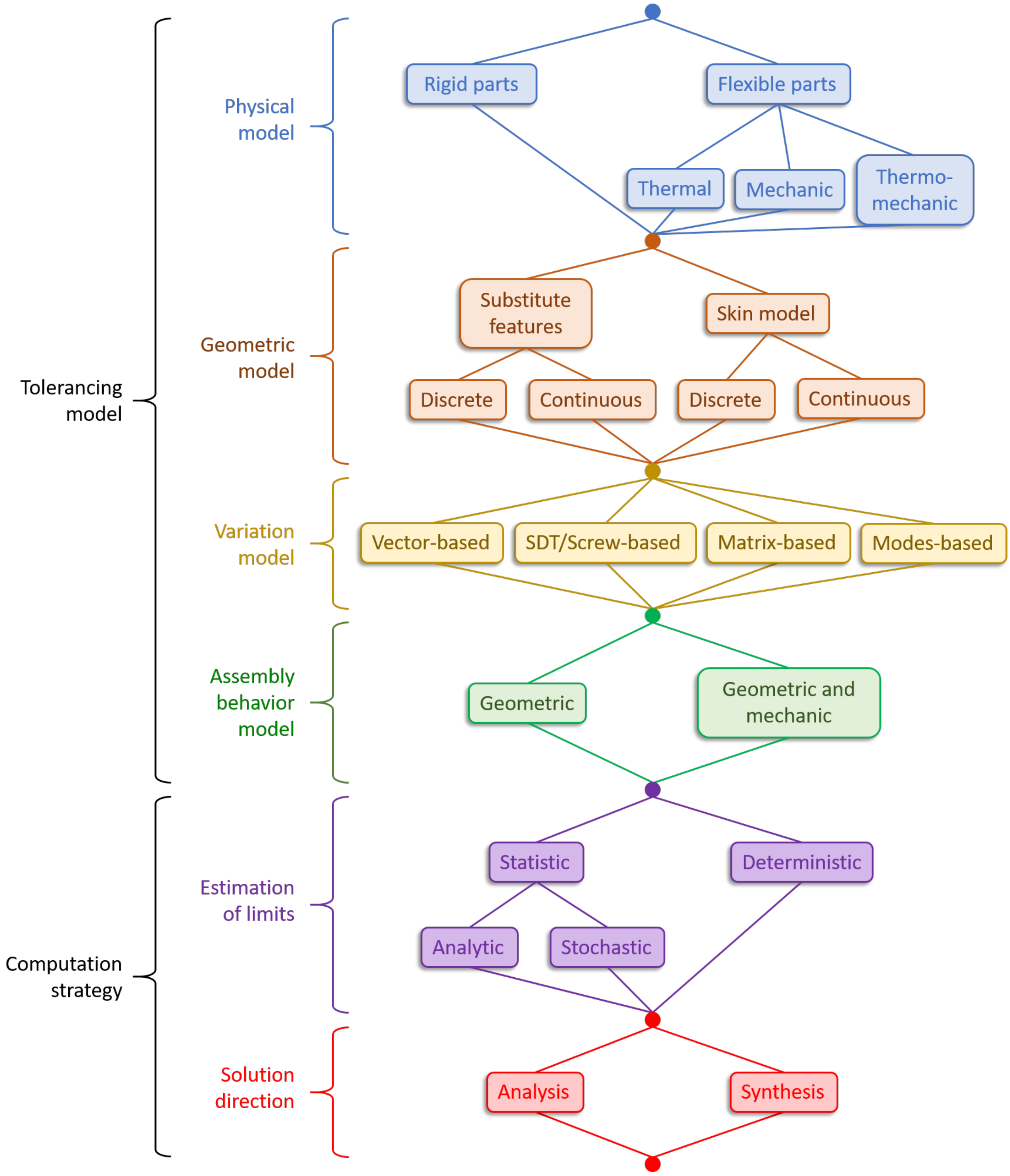


Figure 1.2: Map of geometric tolerancing approaches.

1.1 Tolerancing model

A tolerancing model aims to reproduce not only the geometric variations at the part level, but also their interaction at the assembly level. Reproducing these variations at part level involves, first, the definition of a physical model reproducing the phenomena causing the variations, next, the choice of model for describing the geometry of manufactured parts, and finally, a mathematical model to represent their variations.

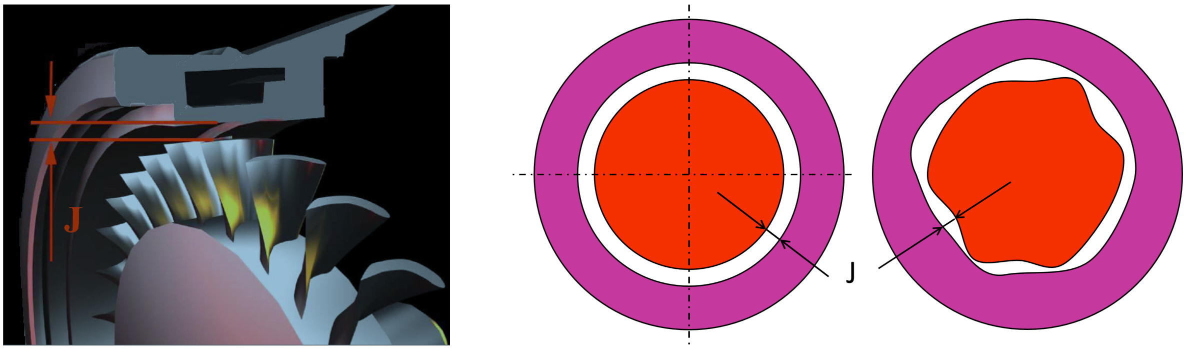
1.1.1 Physical model

Whatever the process, geometric deviations occur during manufacturing and assembly. These deviations generate uncertainties in surface orientation, position and form. Similarly, when products are in operation, other geometric deviations occur, generated by phenomena such as friction, gravity, motion-induced forces, heat, etc. According to the product type and its operating conditions, some of these phenomena may be more relevant than others and cannot be ignored.

The difficulty of modelling all of these variations realistically lies in the consideration of flexible parts. Thus we classify existing models into those that consider rigid bodies and those that consider deformations.

Mechanical deformations are a common concern in the automotive and aircraft industry. Sheet metal parts can suffer significant deviations during the assembly process, and they cannot therefore be modelled as rigid bodies (Wärmefjord et al., 2016). Beyond the automotive and aircraft industry, the elastic properties of parts can be considered as a possibility for recovering parts originally considered as non-compliant (Petit, 2004).

Thermal-induced deviations cannot be neglected in certain cases, for example on turboshaft engines. Pierre (2011) and Rique Garaizar et al. (2016) modelled thermo-mechanical deformations in turbine blade tips (see Figure 1.3). Benichou and Anselmetti (2011) considered the effect of thermal expansions in shafts, looking at uniform coaxiality deviation. Lorin et al. (2013) consider aesthetic impacts of thermal expansions in car parts.



(a) 3D model.

(b) 2D nominal and skin model.

Figure 1.3: Functional requirement in a turboshaft engine (Pierre, 2011).

Finding a general and efficient method to consider the deformation of parts in geometric tolerancing is a challenging task. Either the developed methods are based on specific models, valid for a given kind of part or they include finite element analysis, which tend to be time-consuming (Rouetbi et al., 2017; Stricher, 2013). Thus, most geometric tolerancing approaches assume rigid bodies.

1.1.2 Geometric model

Designers usually manipulate only the nominal geometry of products through CAD systems. However, the need to consider geometric deviations has led to alternative geometric representation models.

Skin model shapes are models based on non-ideal geometries representing given instances of manufactured parts, including form defects (see Figure 1.4). Different strategies to generate skin model shapes have been proposed in (Favreliere, 2009; Zhang et al., 2011; Schleich and Wartzack, 2014; Yan and Ballu, 2017; Homri et al., 2017; Zhu et al., 2017). The advantage of this kind of representation is that it can simulate geometric deviations that are expected, predicted or already observed in real manufacturing processes (Anwer et al., 2013).

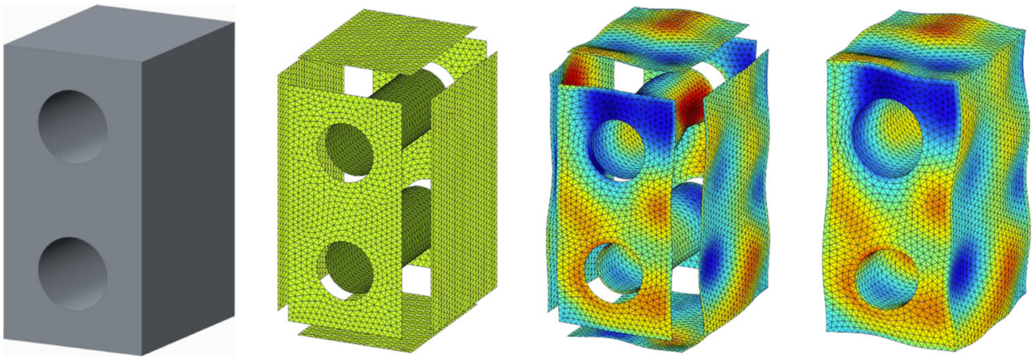


Figure 1.4: Skin model shape generation (Yan and Ballu, 2017).

However, the level of detail of the models, and consequently of the simulations, restricts their use to simple assemblies, as shown in Figure 1.5. Furthermore, designers tend to neglect form defects, assuming that contact clearances are of a higher-order (Adragna et al., 2010; Dumas, 2014); this cannot be assumed at the nano-scale (Samper, 2007).

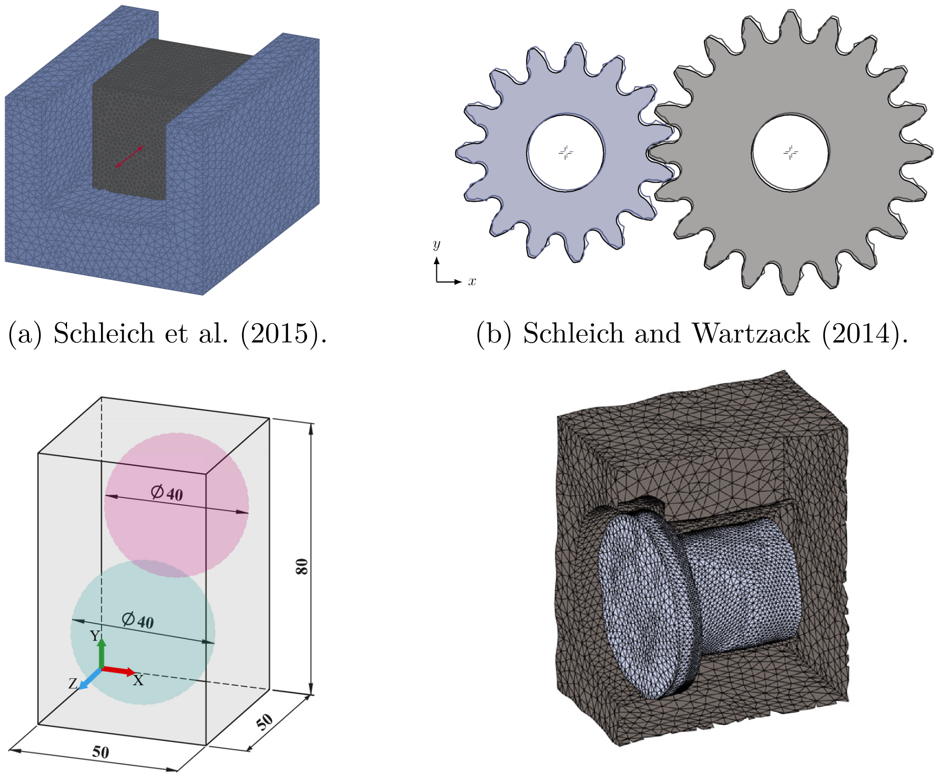
Tolerance simulations are often based on features of perfect form, called substitute features. These are associated to the real feature following a given criterion (minimization of the sum of the squared distances, minimization of the maximal distance) to minimize the form defects. Figure 1.6 illustrates an example of this association process.

In the most approximate case, a real surface is represented as a point (1D tolerancing). In 2D tolerancing, real surfaces are represented by line segments. In 3D tolerancing, perfect form surfaces are used to simulate position and orientation deviations. Seven surface classes are used: spherical, planar, cylindrical, helical, rotational, prismatic and complex. Each of these classes is derived from the displacement subgroup that leaves it globally invariant (Hervé, 1994).

1.1.3 Variation model

A mathematical model is associated to a geometric model to represent its variations. Different kinds of models have been proposed for both skin model-based and substitute feature-based representations.

Variations in skin model shapes are typically simulated by linear combinations of form error



(a) Schleich et al. (2015).

(b) Schleich and Wartzack (2014).

(c) Corrado et al. (2017).

(d) Schleich and Wartzack (2015).

Figure 1.5: Simulations with skin model shapes.

modes associated to the nominal geometry (Samper, 2007; Homri et al., 2017). Figure 1.7 illustrates examples of some natural modes of a plane with circular boundary.

Following this logic, neglecting form defects in geometric tolerancing corresponds to considering only the first six natural modes. In these cases, the deviations (in position and orientation) of the substitute feature are characterized by means of vectors, transformation matrices or torsors/screws. The difference between these models lies in the assumed dimension of the substitute feature and the consideration or not of rotations as linear (i.e. small displacements).

When the substitute feature is a point and only one movement direction is considered, the variation is associated to a 1D interval. Despite its approximation level, this model is still used by the draftsmen's community because of its simplicity (Shah et al., 2007). This model can also be used in 2D (Chase et al., 1995) and 3D (Gao et al., 1998) by associating a vector to the point (vectorial tolerancing).

In 3D tolerancing, the position and orientation of features can be characterized by means of homogeneous transforms (Sodhi and Turner, 1994). This approach involves manipulating non-linear relations, which can make further manipulations difficult. Serré et al. (2011) suggest to use conformal geometric algebra to solve such relations. Another way to proceed is to linearize these relations assuming that rotations are small with regard to the dimensions of the features (Bourdet et al., 1996). An example of this linearization is presented in Figure 1.8. Among the linearized models, we can find one based on Jacobian matrices (Laperrière and ElMaraghy, 2000; Desrochers et al., 2003), one based on small displacement torsors (SDT) (Anselmetti, 2006; Clozel and Rance, 2010) or screws (Desrochers and Delbart, 1998; Adams and Whitney,

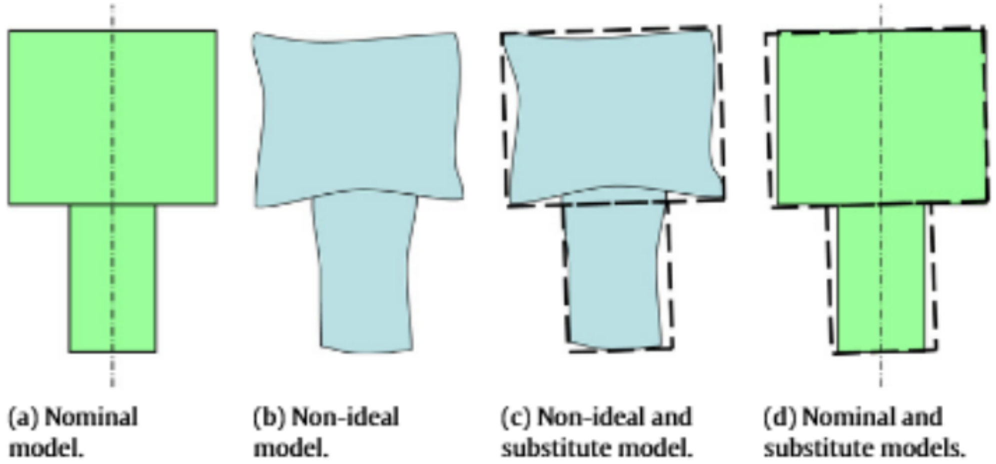

(a) Nominal model.
(b) Non-ideal model.
(c) Non-ideal and substitute model.
(d) Nominal and substitute models.

Figure 1.6: Substitute surface association (Dantan and Qureshi, 2009).

1999; Kumaraswamy et al., 2013).

Models such as domains, T-Maps and polytopes represent all possible positions of a feature within their tolerance zone or two surfaces potentially in contact. These models consider the deviation limits of features and contacts in an abstract deviation space.

A deviation domain is created by translating the displacement limits imposed by the tolerance zones (or contact clearance) into algebraic constraints expressed in terms of small displacements. The same reasoning is applied to toleranced joints to create clearance domains. A graphic representation of a domain derived from a planar surface with a circular contour is shown in Figure 1.9a. Further details about this model can be found in (Giordano et al., 1992; Samper et al., 2006; Giordano et al., 2007; Mansuy et al., 2013b).

T-Maps are obtained from a basic simplex which is described with areal coordinates in the space defined by the Euclidean motions. The extreme positions of the related feature within its tolerance zone are mapped to a specific point of the T-Map. A graphic representation of a T-Map derived from a planar surface with a circular contour is given in Figure 1.9b. This method has been described in various studies (Davidson et al., 2002; Davidson and Shah, 2002; Ameta et al., 2004; Jian et al., 2007).

The polytope-based method (which is detailed further in Chapter 2) is also based on SDTs to express the constraints imposed by the tolerance zones. The particularity of this method is that non-linear features' boundaries are discretized to obtain linear constraints and manipulate linear objects only. A graphic representation of a polytope derived from a planar surface with a circular contour is presented in Figure 1.9c. More details about this technique can be found in (Teissandier et al., 1999; Homri et al., 2015; Arroyave-Tobón et al., 2017c)

The main difference between the models described above lies in the way they define the constraints. Although T-Maps and domain models are initially able to handle quadratic constraints, they finally linearize the set boundaries because of the complexity of summing convex non-linear constraints. The effects of this linearization are discussed in (Dumas et al., 2015). A comparison of T-Maps and domain models can be found in (Ameta et al., 2011; Mansuy et al., 2013a) and comparisons with parametric approaches are described in (Shah et al., 2007; Pierre and Anselmetti, 2014). Other reviews are presented in (Prisco and Giorleo, 2002; Chen et al.,

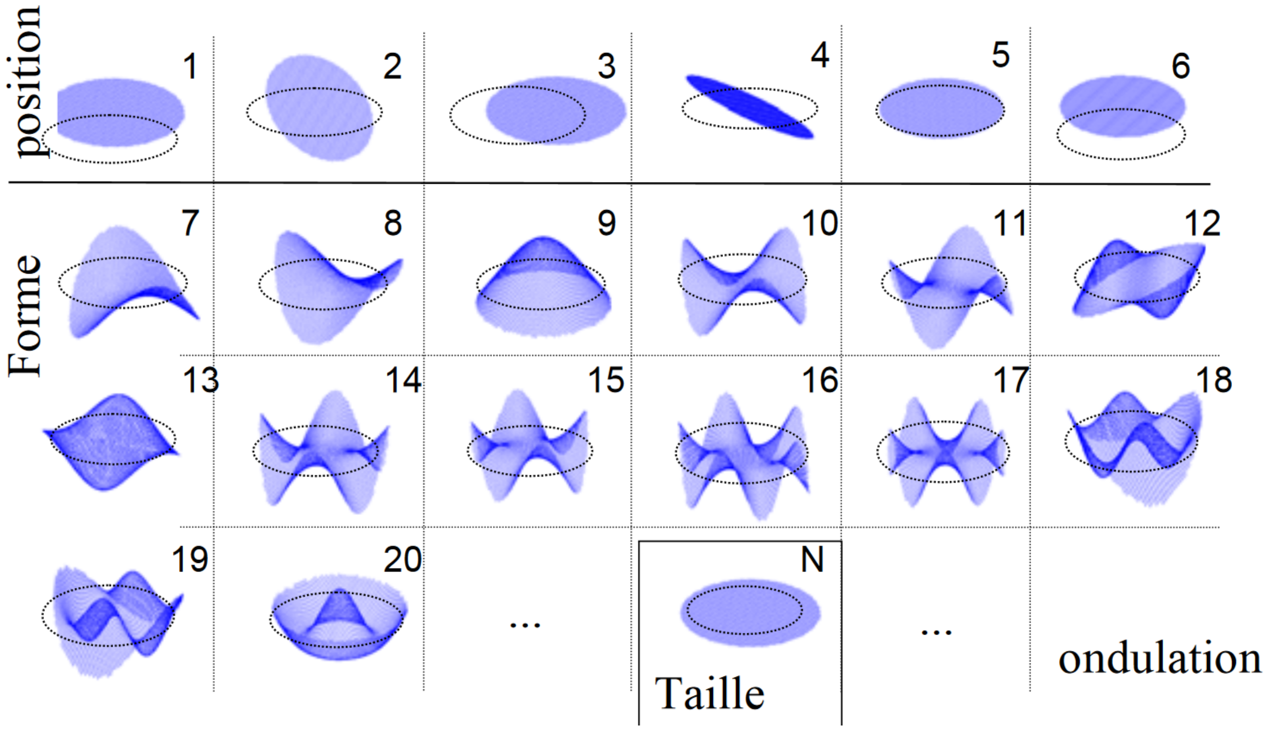


Figure 1.7: Natural modes of a disc (Samper, 2007).

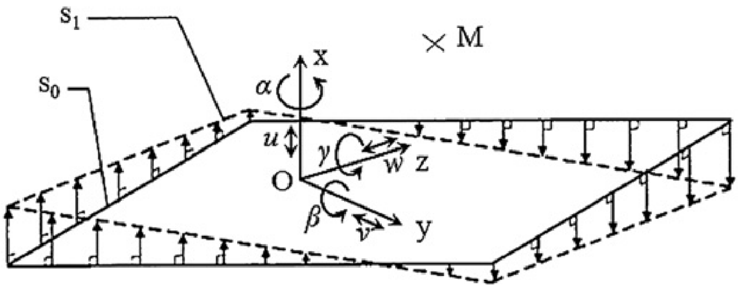


Figure 1.8: Rotation linearization (Chen et al., 2014).

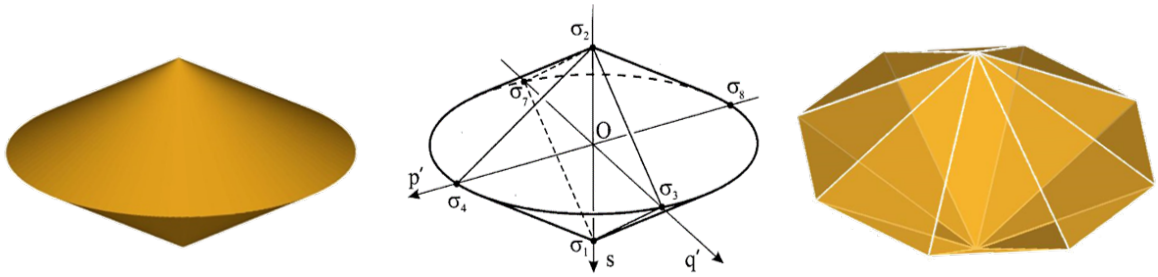
2014).

1.1.4 Assembly behaviour model

At the assembly level, component uncertainties combine to produce greater uncertainties. In mathematical terms, there is an assembly response Y representing the accumulative stack-up of deviations which depends of independent deviations on n components $\{X_1, X_2, ...X_n\}$ (Nigam and Turner, 1995; Dantan and Qureshi, 2009):

$$Y = f(X_1, X_2, ...X_n) \tag{1.1}$$

Function f corresponds to the assembly behaviour model and simulates the effects of deviation interaction. This interaction depends on how individual parts are mated (geometric



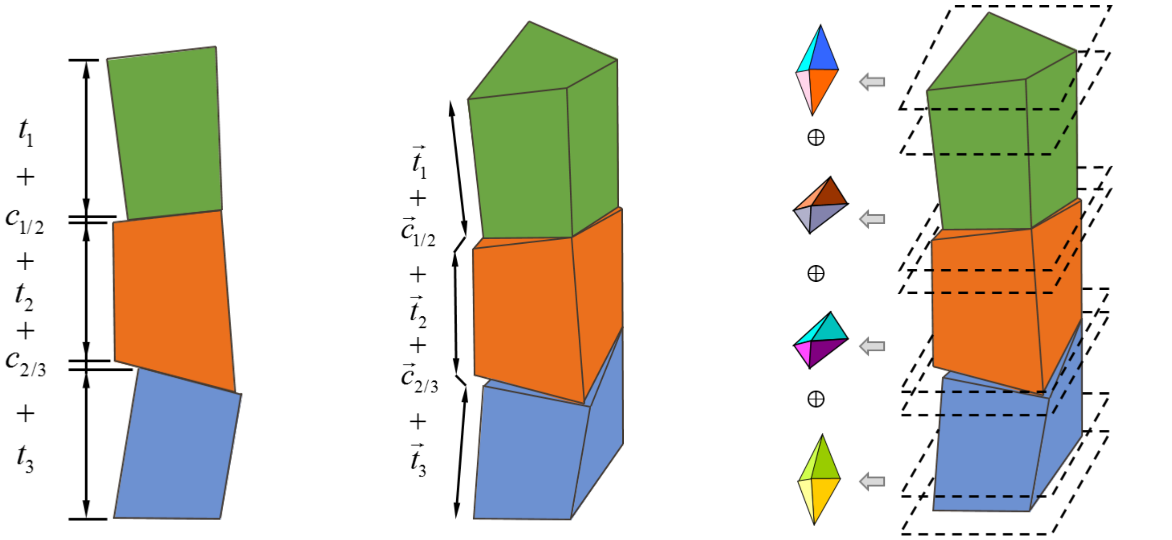
(a) Domain. (b) T-Map. (c) Polytope.

Figure 1.9: Comparison of domains, T-Maps and polytopes.

behaviour laws) and strength-related phenomena (mechanical behaviour laws).

Considering the interaction of geometric and mechanical behaviour laws simultaneously is challenging. This is why only a few studies have addressed this problem: (Mazur et al., 2011; Samper and Giordano, 2003; Xie et al., 2007; Dupac and Beale, 2010; Stricher, 2013; Lindau et al., 2016). Most tolerancing models therefore consider only geometric behaviour laws.

Geometric behaviour laws depend on the topological structure of the assembly (Bourdet and Ballot, 1996). While serial joints entail a variation accumulation, parallel ones generate a variation counteraction. When variables $\{X_1, X_2, ...X_n\}$ represent 1-dimensional deviations (considering only dimensional and positional variations), Eq. 1.1 is reduced to an arithmetic operation (Shah et al., 2007). Figure 1.10a illustrates this case.



(a) Points variation in 1D. (b) Points variation in 3D. (c) Surfaces variations in 3D.

Figure 1.10: Models for representing geometric variations and their accumulation.

The construction of the assembly response function depends on how variations are modelled. When using Unified Jacobian-Torsors, f is a relation of matrices (Ghie et al., 2010; Zeng et al., 2017b). The CLIC method (Anselmetti, 2006) proposes to determine a set of analysis lines along a tolerance chain. These lines represent the most disadvantageous movement directions for a given design requirement. By using pre-established relations for the different types of joints, the assembly response can be characterized. The TTRS (Technologically and Topologically

Related Surfaces) method suggests a successive surface association to obtain f (Desrochers and Clément, 1994; Clément et al., 1998). These associations are based on intersections and/or unions of displacement subgroups. The same results can be obtained by a direct manipulation of SDTs or screws and solving the derived system of linear equations (Ballot et al., 2003; Teissandier, 2012; Bourdet et al., 2013).

Using these strategies, explicit linear relations can be obtained (as shown in Figure 1.10b) (Ledoux and Teissandier, 2013); which facilitate their integration into commercial Computer-Aided Tolerancing tools (Clozel and Rance, 2010). However, the main drawback of these models is that they consider each movement direction to be independent from the others (Dantan, 2000; Homri et al., 2015). This implies that more than one simulation can be required to validate the fulfilment of a design requirement. Furthermore, over-constrained mechanisms cannot be treated with these models.

These limitations can be overcome when all possible feature deviations are taken into account simultaneously. By means of models such as domains, T-Maps or polytopes, the dependencies between rotations and translations are considered, requiring only one simulation to ensure the compliance of a design requirement, even in the case of over-constrained assemblies. In this instance, f is a relation of 6D vector sets. For parallel contacts, the intersection of the respective vector sets is required (Teissandier et al., 1999; Mansuy, 2012; Homri et al., 2015). Defect accumulations in serial contacts are simulated by means of Minkowski sums of the respective vector sets (Fleming, 1988; Srinivasan, 1993) (see Figure 1.10c). Once the final calculated polytope is obtained (the polytope containing the entire cumulative stack-up of variations), the compliance of the functional condition can be verified, checking that the calculated polytope is included inside the functional polytope.

1.2 Computation strategy

Different strategies have been proposed in the literature to solve tolerancing problems. These strategies define the way deviations limits are treated (deterministically or statistically) and the solution sense of the problem (analysis or synthesis).

1.2.1 Estimation of limits

To compute the worst-case deviation propagation scenario in an assembly, the extreme values for each tolerance and contact are assumed. By doing this, it is ensured that 100% of the products are going to be assembled and/or work as expected, which becomes interesting for prototypes or small production series. However, it implies tighter tolerances, and consequently, higher manufacturing costs due to additional machining operations or more expensive machining equipment (Jeang, 1994).

The statistical approach considers that it is more economical to reject a small percentage of production than to use a more accurate manufacturing process (Fleming, 1987). Using this idea, tolerance values are relaxed to some extent, which is interesting for high production rates. The challenge is then to find the right compromise between the cost of the rejected production and the cost of the increase in the tolerance values.

The objective of a statistical simulation is to estimate the probability distribution of Y considering the probability distributions of the contributors. When real manufacturing data are available, more accurate tolerance propagation simulations can be performed (Dantan and Qureshi, 2009). This can be done either by analytic or stochastic methods. When analytic methods are used, it is assumed that the contributors $X_1, X_2, ...X_n$ are all normally distributed. Their nominal values are then set at the mean and from the standard deviation of the input variables the probability distribution of the output can be estimated (Morse, 2004; Shen et al., 2005; Ghie et al., 2010):

$$\sigma_Y = \sqrt{\left(\frac{\partial X_1}{\partial f}\right)^2 \sigma_{X_1}^2 + \left(\frac{\partial X_2}{\partial f}\right)^2 \sigma_{X_2}^2 + ...\left(\frac{\partial X_n}{\partial f}\right)^2 \sigma_{X_n}^2} \tag{1.2}$$

When f is not available in analytic form, the partial derivatives cannot be computed. Stochastic methods, such as the Monte Carlo simulation, are then required to generate a population of input parameters and estimate σ_Y (Chase et al., 1995; Yang et al., 2013; Qureshi et al., 2012). A representative number of feature instances are generated, varying their position and orientation according to a given probability distribution (see for example the sample illustrated in Figure 1.11). In this way, intersections and Minkowski sums are avoided. The propagation is computed by means of linear optimization methods combined with reliability calculation algorithms (Dantan and Qureshi, 2009; Beaucaire et al., 2013; Mansuy et al., 2013b). One of the advantages of this strategy is that sensitivity indices can be estimated in relation to the tolerance values, which produces very interesting results for tolerance decision-making (Ziegler and Wartzack, 2015).

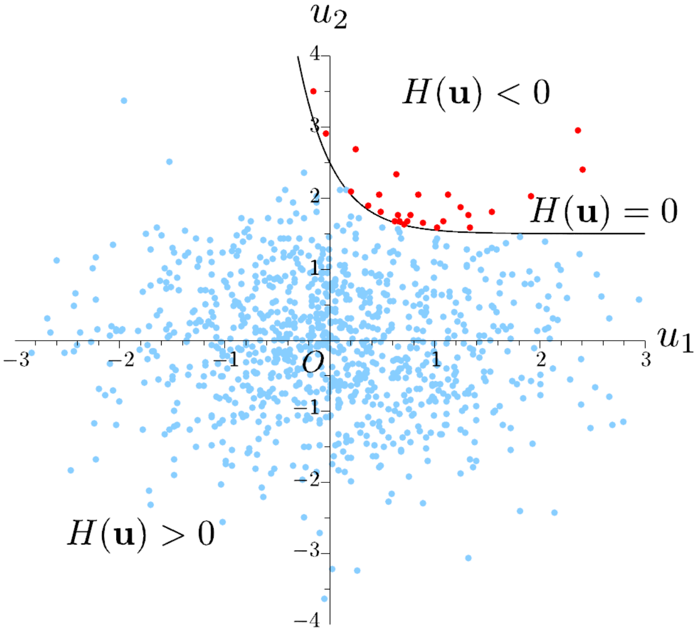


Figure 1.11: Monte Carlo simulation (Dumas, 2014).

1.2.2 Solution direction

In the design process, the functional requirements of products have to be transferred to constraints in the deviations of the constituent parts. This way to solve the problem is referred to as tolerance synthesis. It involves, first defining the required functional specifications (specification synthesis), and second their magnitude (tolerance allocation), so that the functional requirements can be satisfied. The problem of specification synthesis has been addressed by, among others, Anselmetti (2006) and Mejbri et al. (2005). Tolerance allocation is an underconstrained problem, and therefore it is usually addressed from an optimization perspective (Etienne, 2007). The aim is to balance tolerance values against cost aspects while considering manufacturing constraints (Wandebäck et al., 2009). The cost function, which must be minimized, usually takes the form (Chase and Parkinson, 1991; Mansuy et al., 2011):

$$C = A + \frac{b_1}{t_1} + \frac{b_2}{t_2} + ... + \frac{b_n}{t_n} \tag{1.3}$$

where A represents fixed costs (machine setup, material, etc.) and b_i the cost of producing a single component dimension to a specified tolerance t_i.

Different optimization techniques have been used for tolerance allocation considering different types of constraints. Ming and Mak (2001) and also Geetha et al. (2013) used genetic algorithms, Zeng et al. (2017a) and Wu et al. (2009) proposed a statistical strategy based on Monte Carlo simulations; Muthu et al. (2009) used particle swarm optimization.

Due to its complexity, geometric tolerancing problems are usually solved in the opposite direction: the designer first identifies the influential parts along the tolerance chain, chooses tolerances and then verifies that requirements are satisfied (Anselmetti et al., 2010).

1.3 Positioning this thesis in the geometric tolerancing map

The work proposed here is based on the following considerations:

- parts are considered as rigid bodies,
- form defects are not considered,
- contact surfaces are considered frictionless,
- small rotations are considered as linear.

Based on the taxonomy illustrated at the beginning of this chapter, Figure 1.12 positions this thesis in the geometric tolerancing map.

Although this thesis was developed for a given computation strategy (worst-case in tolerance analysis), it can also be applied to the statistical treatment of tolerances as well as tolerance synthesis.

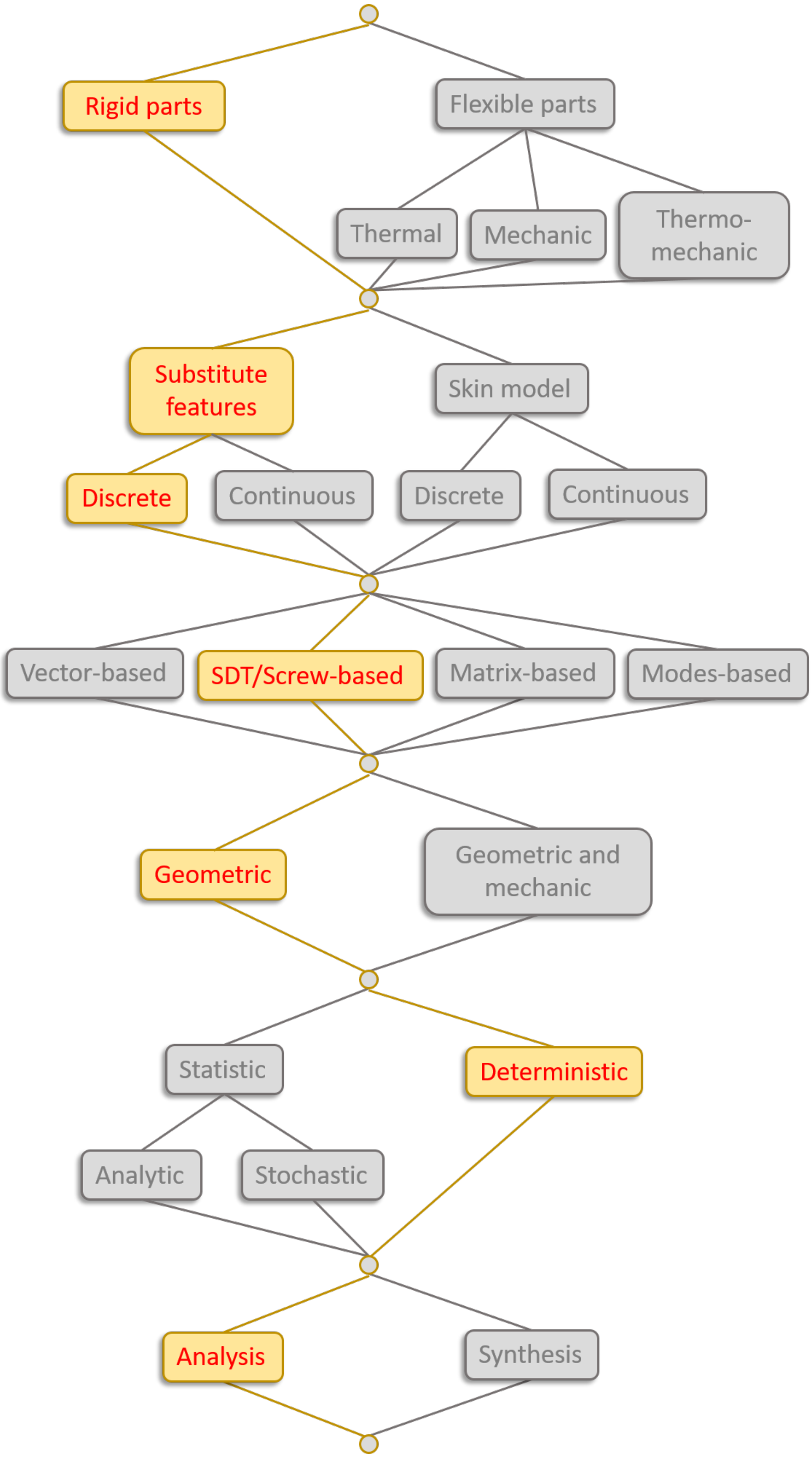


Figure 1.12: Position of this thesis in the geometric tolerancing map.

1.4 Conclusions

The need to consider geometric uncertainties in product quality has brought about several geometric tolerancing approaches. We found that these approaches usually consist of a tolerancing model and a solution strategy.

On the one hand, the tolerancing model serves to reproduce geometric variations of parts, derived from different kinds of phenomena and their influence in the assembly context. Assumptions significantly reduce the difficulty of considering all real situations that influence geometric variations in products. Existing models usually focus on reproducing some phenomena, while neglecting others. This is also because there can be different assembly and operation conditions, making some phenomena more relevant than others. For example, in the automotive and aircraft industry, geometric variations due to part deformation are usually considered, whereas, for the nanotechnology industry, form deviations are an important issue. Thus, the creation of a generic, realistic and efficient model for geometric tolerancing remains an unresolved problem.

On the other hand, the solution strategy determines, first, how deviations limits are considered; and second, the direction for solving the problem. Deviations limits can be considered according to a probabilistic distribution or a worst-case scenario. The probabilistic approach is useful when real manufacturing data are available for simulating a probability distribution of geometric variations. The worst-case approach becomes more interesting for prototypes or small production series, but because of its conservative spirit (tighter tolerances), it requires higher manufacturing costs. Lastly, the direction for solving the problem can be top-down or bottom-up. The top-down fashion, which is the ideal way to do it, involves optimization problems. However, because the complexity it implies, tolerance allocation is typically done in a bottom-up fashion: after choosing a given tolerancing scheme the designer verifies if the design requirements are satisfied.

Chapter 2

Geometric tolerancing with 6D polytopes

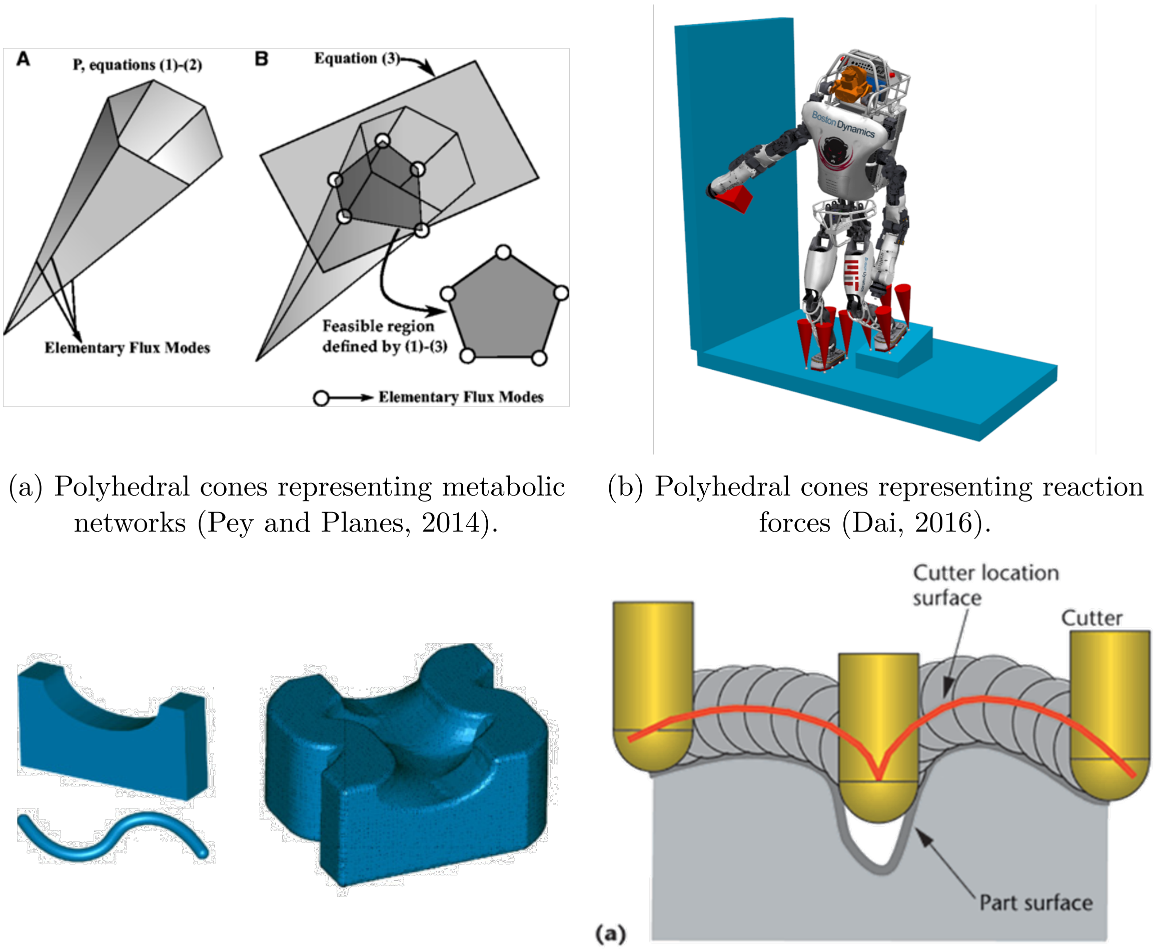
In this chapter, we describe how geometric deviations can be modelled with polytopes. In assemblies, the accumulation of these deviations is computed by summing and intersecting polytopes. This is a summary of the work of Teissandier (2012) and Homri (2014) and establishes the basis for this thesis. We also review the method proposed by Homri (2014) to treat unbounded displacements related to degrees of freedom (of joints) or invariance (of geometric features). In these cases, the derived sets of constraints are unbounded objects, i.e. polyhedra, which are challenging to deal with from an algorithmic point of view. A case study is developed to present the approach and aspects for improvement.

2.1 Combinatorial geometry

The combinations and arrangements of geometric objects is known as combinatorial geometry. It is concerned not only with theoretical aspects but also applied ones (Weibel, 2007).

The applied branch of the combinatorial geometry is called computational geometry. It deals with algorithms computing geometrical objects and solving geometrical problems. Since many general problems in sciences can be solved by geometrical models, the field of applications is very large.

In biology, Pey and Planes (2014) uses polyhedral cones to represent metabolic networks (see Figure 2.1a). In robotics, Firmani et al. (2008) and Dai (2016) represent sets of reaction forces by means of polyhedral cones (see Figure 2.1b). In CAD, Peternell and Steiner (2007) employ polytopes combinatorics (particularly Minkowski sums) in solid modelling (see Figure 2.1c). In manufacturing, Inui and Ohta (2007) use Minkowski sums for computing tool paths (see Figure 2.1d). Grandguillaume et al. (2017) represent kinematic limits of machine-tools with polytopes for choosing tooling orientation.



(a) Polyhedral cones representing metabolic networks (Pey and Planes, 2014).

(b) Polyhedral cones representing reaction forces (Dai, 2016).

(c) Polytopes in solid modelling (Peternell and Steiner, 2007).

(d) Polytopes representing tool paths (Inui and Ohta, 2007).

Figure 2.1: Different application of polytopes.

On the other hand, theory attempts to understand the combinatorial properties of geometrical objects. Some relevant references in the subject, which were an important support for this thesis, are (Fukuda and Rosta, 1994; Ziegler, 1995; Fukuda, 2004; Weibel, 2007). From these references, we extracted the following concepts.

Definition 2.1.1 (Polytope) *A polytope of $\mathbb{R}^n$ is a subset $P \subseteq \mathbb{R}^n$ that can be presented as a $\mathcal{V}$-polytope or, equivalently, as an $\mathcal{H}$-polytope.*

Definition 2.1.2 ($\mathcal{H}$-polytope) *A $\mathcal{H}$-polytope of $\mathbb{R}^n$ is the bounded intersection of a finite number of closed half-spaces of $\mathbb{R}^n$.*

Definition 2.1.3 ($\mathcal{V}$-polytope) *A $\mathcal{V}$-polytope of $\mathbb{R}^n$ is the convex hull of a finite number of points of $\mathbb{R}^n$.*

Minkowski-Weyl theorem states that the former two definitions are equivalent (see Figure 2.2).

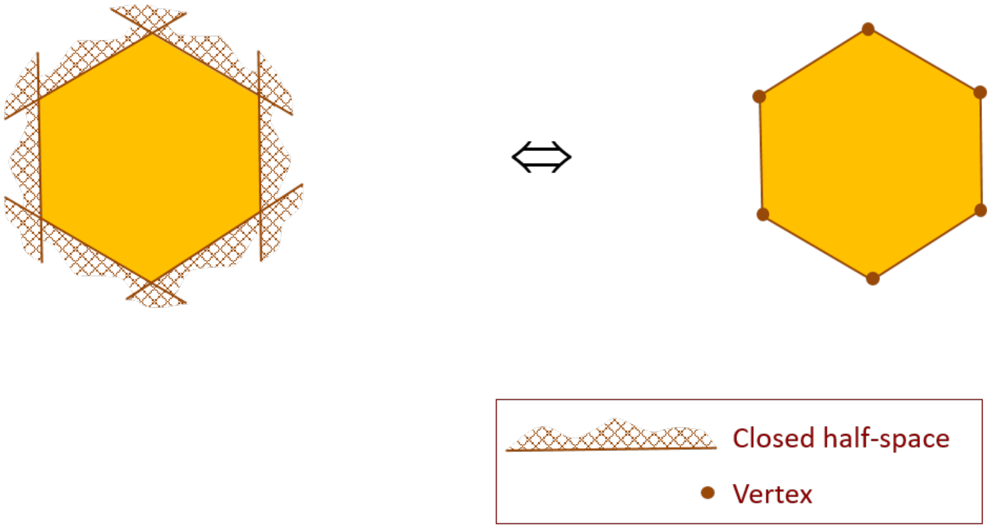


Figure 2.2: Two equivalent definitions of a polytope: half-spaces versus vertices.

Definition 2.1.4 (Polyhedron) *A polyhedron is a unbounded polytope.*

2.2 Modelling sets of constraints with 6D polyhedra

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 $\boldsymbol{N}_i$, this restriction is transferred to each of them. These geometric constraints can be modelled as algebraic constraints:

$$S_1 \subseteq TZ \Leftrightarrow \forall \boldsymbol{N}_i \in S_0 : d^{sup} \geq \boldsymbol{t}_{\boldsymbol{N}_i} \cdot \boldsymbol{n}_i \geq d^{inf} \tag{2.1}$$

where S_1 is the substitute surface related to the nominal feature S_0, TZ is the tolerance zone defined offsetting S_0 from d^{inf} to d^{sup} and $\boldsymbol{t}_{\boldsymbol{N}_i}$ is the translation displacement of S_1 in relation to S_0 at point $\boldsymbol{N}_i$ (see Figure 2.3).

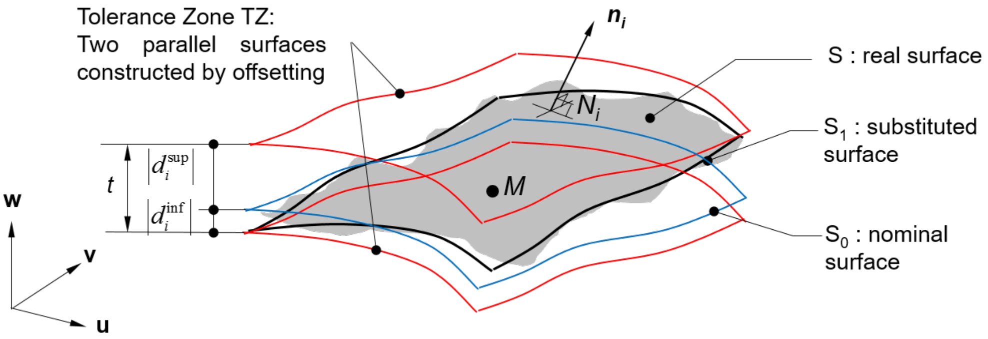


Figure 2.3: Toleranced feature and tolerance zone (Homri et al., 2015).

Expressing the constraints in (2.1) at any point $\boldsymbol{M}$ (which is assumed to be rigidly linked with the toleranced feature) of the Euclidean space and linearizing the rotations under the assumption of small displacements (Bourdet et al., 1996) we have:

$$d^{sup} \geq (\boldsymbol{t}_M + \boldsymbol{N}_i\boldsymbol{M} \times \boldsymbol{r}) \cdot \boldsymbol{n}_i \geq d^{inf} \tag{2.2}$$

where $\boldsymbol{r}$ is the rotation vector of S_1 in relation to S_0.

Considering a vectorial base $[\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}]$, we define the vectors $\boldsymbol{r} = [r_u, r_v, r_w]$, $\boldsymbol{t_M} = [t_u, t_v, t_w]$, $\boldsymbol{N}_i\boldsymbol{M} = [d_{iu}, d_{iv}, d_{iw}]$, $\boldsymbol{n}_i = [n_{iu}, n_{iv}, n_{iw}]$. When expanding the vectorial and scalar products in inequality (2.2) and setting $r_u = x_1$, $r_v = x_2$, $r_w = x_3$, $t_u = x_4$, $t_v = x_5$ and $t_w = x_6$, we obtain:

$$\begin{aligned} d^{sup} \geq &(n_{iv}d_{iw} - n_{iw}d_{iv})x_1 + (n_{iw}d_{iu} - n_{iu}d_{iw})x_2 + \\ &(n_{iu}d_{iv} - n_{iv}d_{iu})x_3 + n_{iu}x_4 + n_{iv}x_5 + n_{iw}x_6 \geq d^{inf} \end{aligned} \tag{2.3}$$

These constraints represent closed half-spaces of $\mathbb{R}^6$. For each point $\boldsymbol{N}_i \in S_0$, two parallel half-spaces are obtained. Then, if a finite set of m points $\boldsymbol{N}_i$ is considered, a set of $2m$ constraints is obtained (Homri et al., 2015). According to definition 2.1.2, the intersection of these constraints defines a convex $\mathcal{H}$-polyhedron in $\mathbb{R}^6$ (Teissandier et al., 1999):

$$\Gamma = \bigcap_{k=1}^{k_{max}} \bar{H}_k^+ \tag{2.4}$$

where

$$\bar{H}_k^+ = \left\{ \boldsymbol{x} \in \mathbb{R}^6 : b_k + a_{k_1}x_1 + ... + a_{k_6}x_6 \geq 0 \right\} \tag{2.5}$$

Let us take, for example, the toleranced feature depicted at the left-hand side in Figure 2.4. For illustrative purposes, let us consider it as a 2D model: only displacements in the plane $[\boldsymbol{x}, \boldsymbol{y}]$ are taken into account. The tolerance zone implies the restriction on the translation of the points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$, $\boldsymbol{t}_{q_1}$ and $\boldsymbol{t}_{q_2}$ respectively:

$$c/2 \geq \boldsymbol{t_{q_1}} \cdot \boldsymbol{y} \geq -c/2$$

$$c/2 \geq \boldsymbol{t_{q_2}} \cdot \boldsymbol{y} \geq -c/2$$

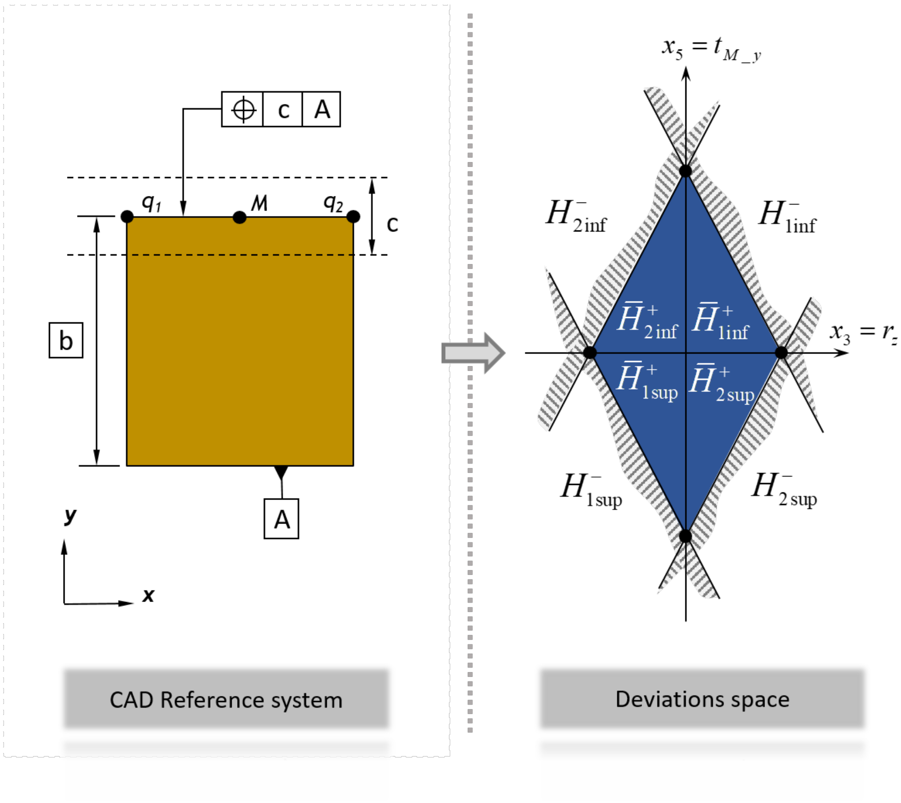


Figure 2.4: A polytope representing geometric deviations in 2D.

When expressing the former inequalities at point $\boldsymbol{M}$, we have:

$$c/2 \geq (\boldsymbol{t_M} + \boldsymbol{q_1 M} \times \boldsymbol{r}) \cdot \boldsymbol{y} \geq -c/2$$
$$c/2 \geq (\boldsymbol{t_M} + \boldsymbol{q_2 M} \times \boldsymbol{r}) \cdot \boldsymbol{y} \geq -c/2$$

assuming $\boldsymbol{q_1 M} = [d, 0, 0]$ and $\boldsymbol{q_2 M} = [-d, 0, 0]$, we obtain four half-spaces:

$$\begin{aligned}
\bar{H}^+_{1inf} &: c/2 + t_{M\_y} - r_z d \geq 0\\
\bar{H}^+_{1sup} &: c/2 - t_{M\_y} + r_z d \geq 0\\
\bar{H}^+_{2inf} &: c/2 + t_{M\_y} + r_z d \geq 0\\
\bar{H}^+_{2sup} &: c/2 - t_{M\_y} - r_z d \geq 0
\end{aligned}$$

In the space spanned by $[r_z, t_{M\_y}]$, the intersection of these half-spaces defines a bounded polyhedron, i.e. a polytope, illustrated at the right-hand side of Figure 2.4. When considering, for example, the 3D space spanned by $[r_z, t_{M\_x}, t_{M\_y}]$, the intersection of the half-spaces generates an unbounded object as the tolerance zone does not impose limits on t_{M_x} (see Figure 2.5). The same occurs with the other unbounded displacements in $\mathbb{R}^6$.

In a similar way, a polyhedron representing the allowable displacements of a couple of features potentially in contact can be defined. In this case, the toleranced feature is defined in the nominal case of permanent contact between the features (see Figure 2.6), and the tolerance zone is obtained offsetting the toleranced feature according to the clearance value (Teissandier

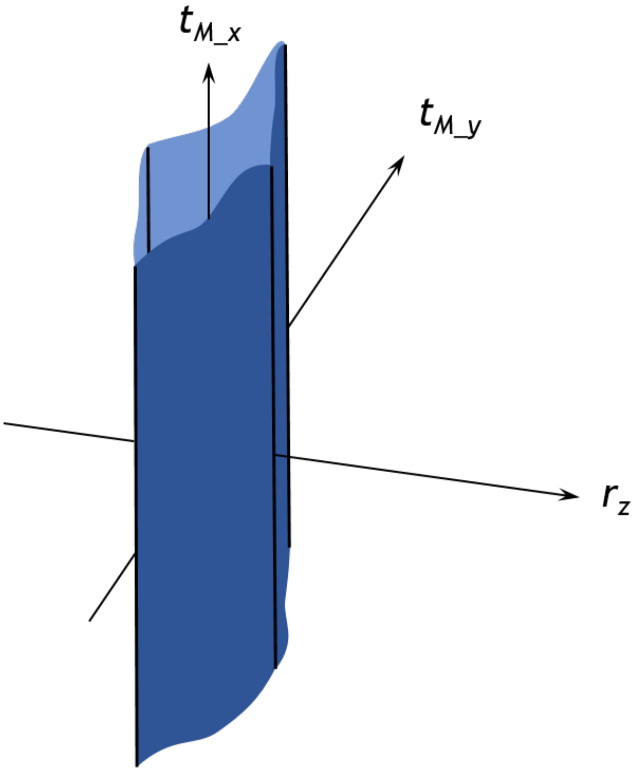


Figure 2.5: A polyhedron representing geometric deviations in 3D.

et al., 1999; Homri, 2014).

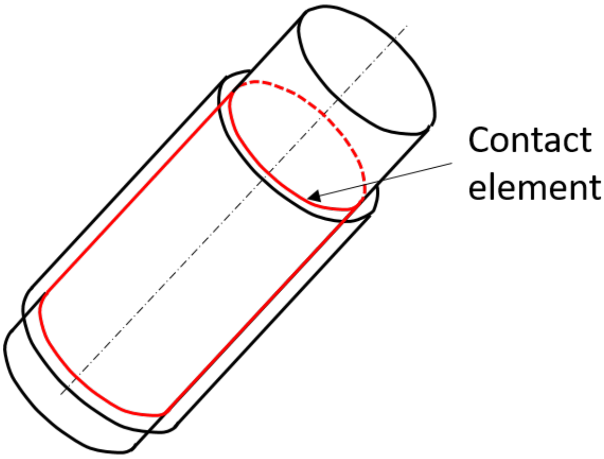


Figure 2.6: Cylindrical pair (Homri, 2014).

In general, the unbounded displacements characterized by the degrees of freedom of contact features or the degrees of invariance of geometric features generate unbounded sets of constraints, i.e. polyhedra. Just in the case of a complex surface, the derived set of constraints defines a polytope (a bounded polyhedron) in $\mathbb{R}^6$.

In the case of unilateral pairs (ball-and-cylinder pair, cylinder-and-plane pair and planar pair), asymmetric sets of constraints are obtained. This because the lower clearance bound is $d^{inf} = 0$ (ensuring no interpenetration) and the upper one d^{sup} is not defined, i.e. it is an infinite value. For further details in this aspect, the reader can refer to (Homri, 2014).

2.3 Modelling stack-up of deviations by operations with polytopes

When considering rigid parts, the defects propagation in a mechanical system depends on how the constituting parts mated. The cumulative stack-up of deviations between any couple of surfaces of an assembly can be simulated operating with geometric and contact constraints. To do this, all the constraints must be expressed under the same reference system and at the same point. The set of required operations can be determined according to the topological structure of the assembly (serial or parallel).

As presented in Eq. 2.5, the native input data used for defining polyhedra in geometric tolerancing is the $\mathcal{H}$-representation (set of closed half-spaces). As we expose next, the $\mathcal{V}$-representation (set of vertices) is also required for computing some operations. So we handle polytopes in their $\mathcal{HV}$-representation.

2.3.1 Modelling parallel architectures - intersections

The interaction of geometric deviations when parts are mated in parallel, i.e. sharing multiple contacts, can be modelled as the intersection of the derived sets of constraints. In the typical example of a clamp, the misalignment between the connected parts can be calculated intersecting the polyhedra derived from the pin-hole joints, as depicted in Figure 2.7.

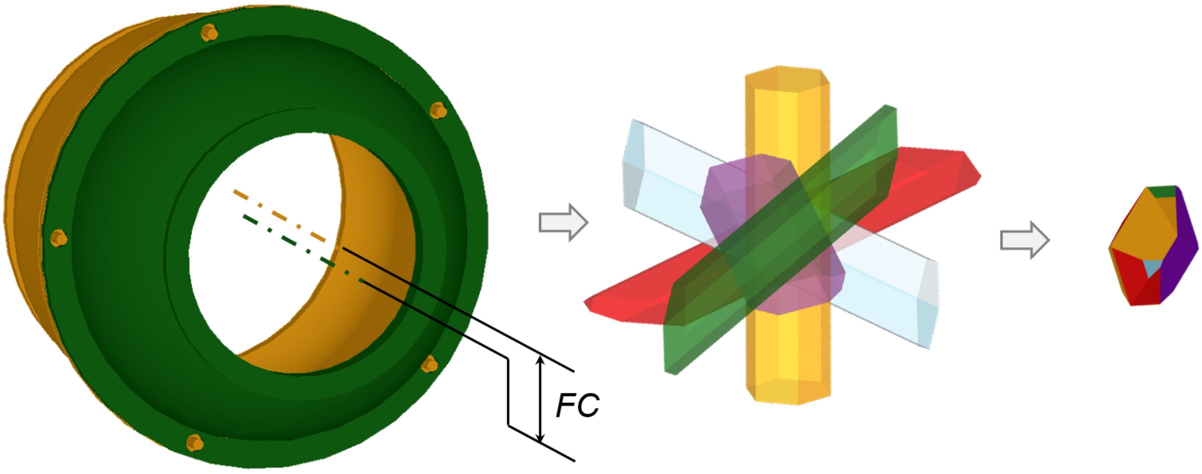


Figure 2.7: Modelling stack-up of deviations as the intersection of polyhedra (Gouyou et al., 2016).

Algorithmically, the computation of the intersection of polyhedra is not complicated. It demands joining together the constraints of both operands and removing the redundant ones. Then, an intersection of two polyhedra Γ_1 and Γ_2 can be computed with the $\mathcal{H}$-representation of the operands (Teissandier, 2012; Arroyave-Tobón et al., 2017c):

$$\Gamma_1 \cap \Gamma_2 = \left(\bigcap_{k_1=1}^{k_{max1}} \bar{H}_{k_1}^{+} \right) \cap \left(\bigcap_{k_2=1}^{k_{max2}} \bar{H}_{k_2}^{+} \right) \tag{2.6}$$

2.3.2 Modelling serial architectures - Minkowski sums

Fleming (1988) and Srinivasan (1993) established the correlation between cumulative defect limits on parts in contact and the Minkowski sum of sets of constraints. In other words, if several parts are mated in a serial configuration, the stack-up of their geometric deviations can be calculated summing the geometric and contact polyhedra involved in the toleranced chain (see Figure 2.8).

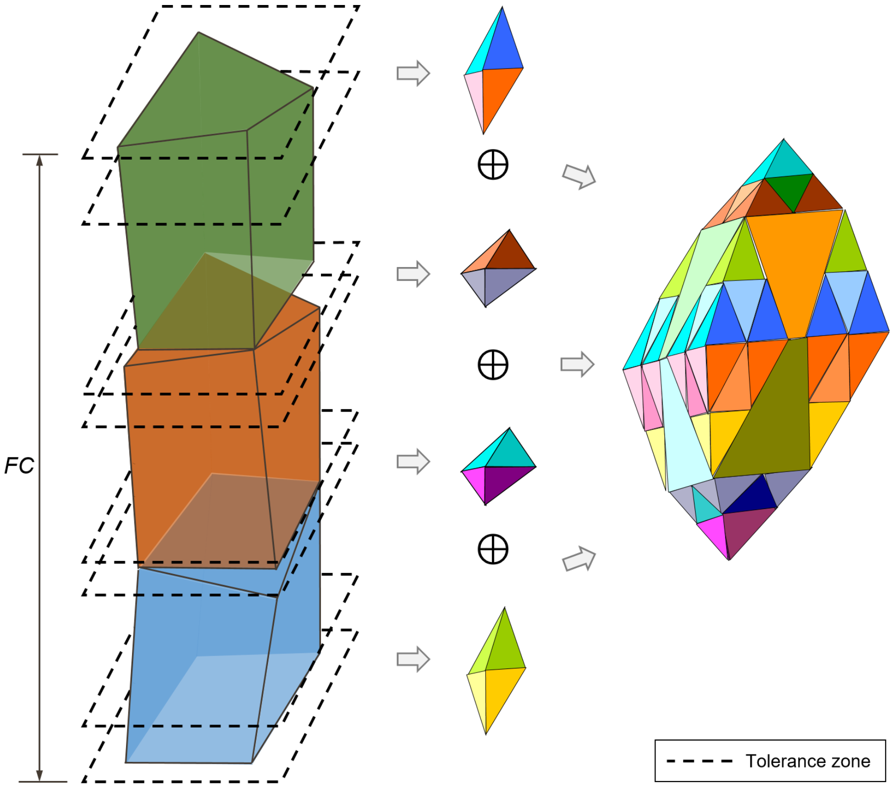


Figure 2.8: Modelling stack-up of deviations as the sum of polytopes.

Definition 2.3.1 (Minkowski sum) *Let P_1 and P_2 be two polytopes. Their Minkowski sum is defined as:*

$$P_1 \oplus P_2 = \left\{ \boldsymbol{a} + \boldsymbol{b}, \boldsymbol{a} \in P_1, \boldsymbol{b} \in P_2 \right\}$$

Because their unbounded nature, Minkowski sum of polyhedra is challenging in computational geometry. This is why few works has been published in this subject. A graphical example of the sum of polyhedra is presented in Figure 2.9.

Fukuda (2004) presented an algorithm to compute Minkowski sums of polytopes, mentioning the possibility of applying the same procedure for the case of polyhedra with at least one vertex (pointed polyhedra) by treating infinite rays as points at 'infinity'. However, due to the degrees of freedom (or invariance), the polyhedra manipulated in tolerancing usually do not have vertices. In fact, each degree of freedom (or invariance) implies a sweeping operation of a polytope along a straight line, placing the vertices at infinity.

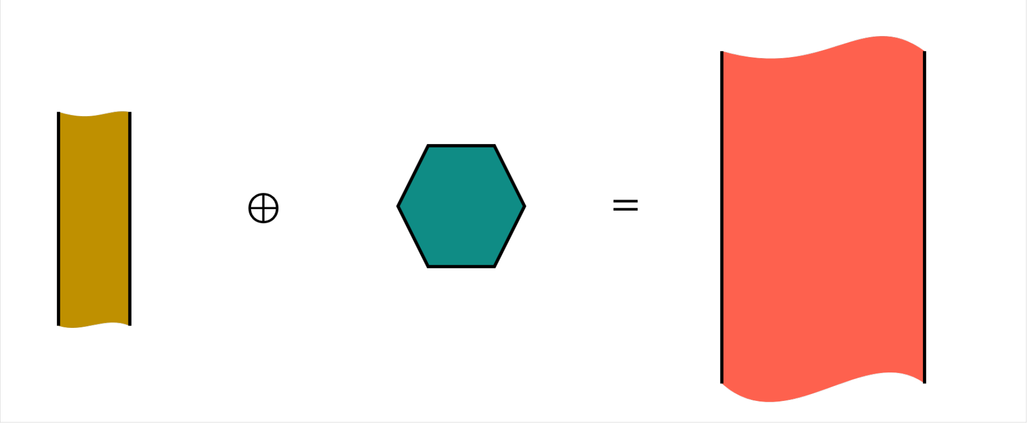


Figure 2.9: Sum of 2D polyhedra.

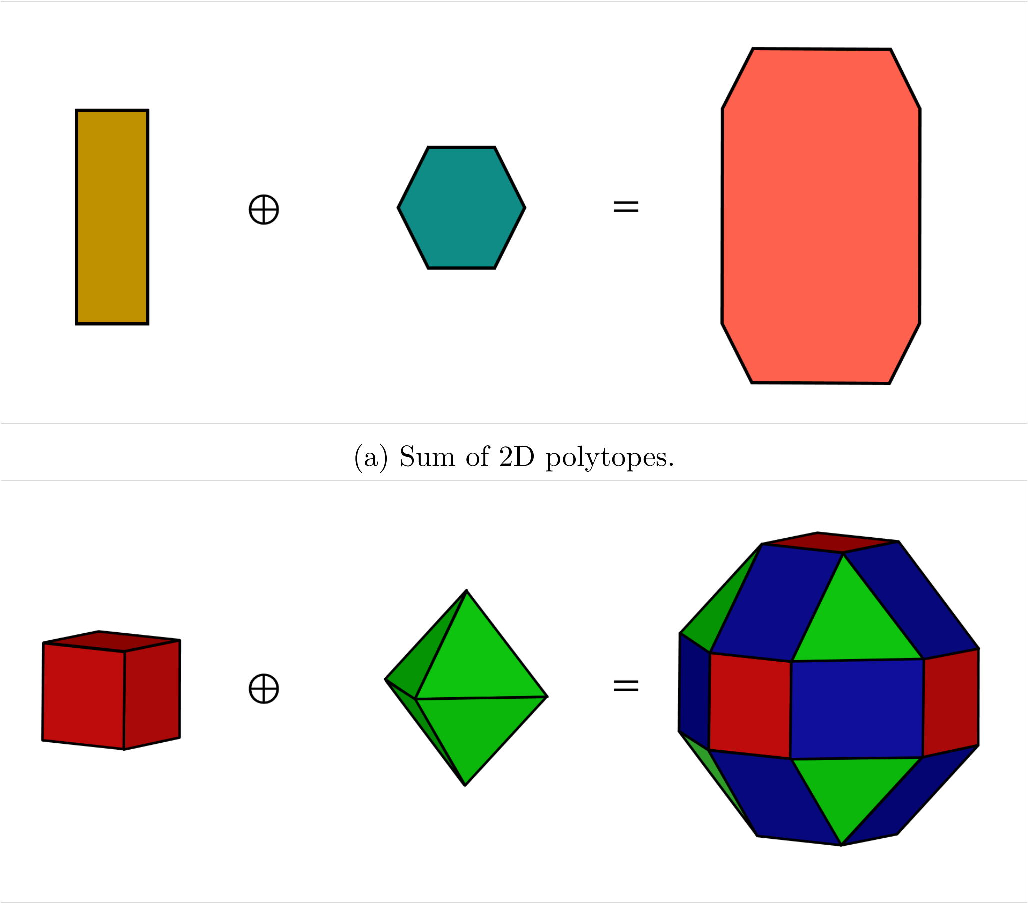
Figures 2.10a and 2.10b present examples of Minkowski sums of 2D and 3D polytopes respectively. Probably the most common technique for computing sums of polytopes consists in adding all the vertices of the operands (Wu et al., 2003; Peternell and Steiner, 2007). It implies, afterwards, the computation of the convex hull of the calculated cloud of points, which can be expensive in an affine space of dimension 6. In addition, this method has to deal with the identification of the points which are not vertices but which are located on the boundary of the calculated polytope.

Although some improvements in the calculation of the sum of the vertices of the operands are presented in (Weibel, 2007; Mansuy et al., 2011; Delos and Teissandier, 2015c), these algorithms do not provide the $\mathcal{H}$-description of the calculated polytope (required in tolerance analysis to compute subsequent intersections). Other methods have been proposed in the literature (Fogel and Halperin, 2007; Lien, 2010; Li and McMains, 2014) but they are only applicable in a 3-dimensional space and can not be generalized to higher dimensions.

Within the context of geometric tolerancing, Mansuy et al. (2011) propose a method for calculating separately the sum of the most disadvantageous vertices with respect to a functional polytope. Even if this method avoids the computation of Minkowski sums, the set of computed vertices is only representative of a given functional condition. In addition, the authors only consider the case of serial tolerance chains made up of features of the same invariance class and in a particular relative position (i.e. a set of parallel planes).

Teissandier and Delos (2011) proposed a method for summing $\mathcal{HV}$-polytopes in a 3D space. The generalization in $\mathbb{R}^n$ is formalized in (Delos and Teissandier, 2015b). With this algorithm it is possible to compute Minkowski sums taking advantage of the duality property of polytopes. This property, proved by Ziegler (1995), states that the normal fan of a Minkowski sum of two polytopes $P_1 \oplus P_2$ is the common refinement of the normal fans of its summands:

$$\mathcal{N}(P_1 \oplus P_2) = \mathcal{N}(P_1) \wedge \mathcal{N}(P_2) \tag{2.7}$$



(a) Sum of 2D polytopes.

(b) Sum of 3D polytopes.

Figure 2.10: Sum of polytopes. Operands assumed to be centred on the origin.

Definition 2.3.2 (Normal fan) *The normal fan $\mathcal{N}(P)$ a polytope P of $\mathbb{R}^n$ is defined as the set of all the dual cones of P (see Figure 2.11). $\mathcal{N}(P)$ partitions $\mathbb{R}^n$.*

Definition 2.3.3 (Dual cone) *A dual cone $C_D(\boldsymbol{v})$ of a vertex $\boldsymbol{v}$ is defined as the positive linear combination of the set of outer normals of its corresponding facets.*

Then, to calculate the common refinement of the normal fans of two polytopes P_1 and P_2, each pair of polyhedral cones $C_D(\boldsymbol{v_{1i}}) \in \mathcal{N}(P_1)$ and $C_D(\boldsymbol{v_{2j}}) \in \mathcal{N}(P_2)$ must be intersected. Examples of the Minkowski sum of polytopes following this strategy is presented in Figures 2.12a and 2.12b.

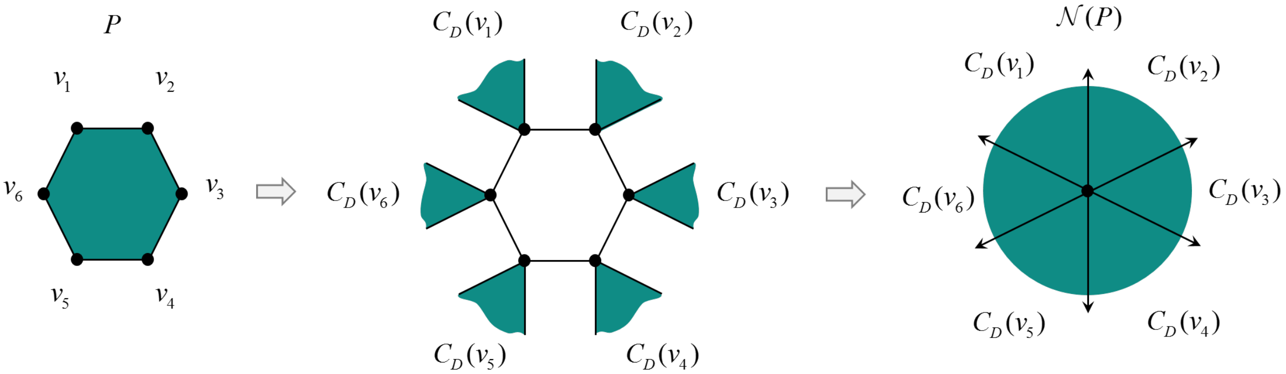


Figure 2.11: Normal fan $\mathcal{N}(P)$ of a polytope P.

2.3.3 Checking requirements satisfaction: inclusion test

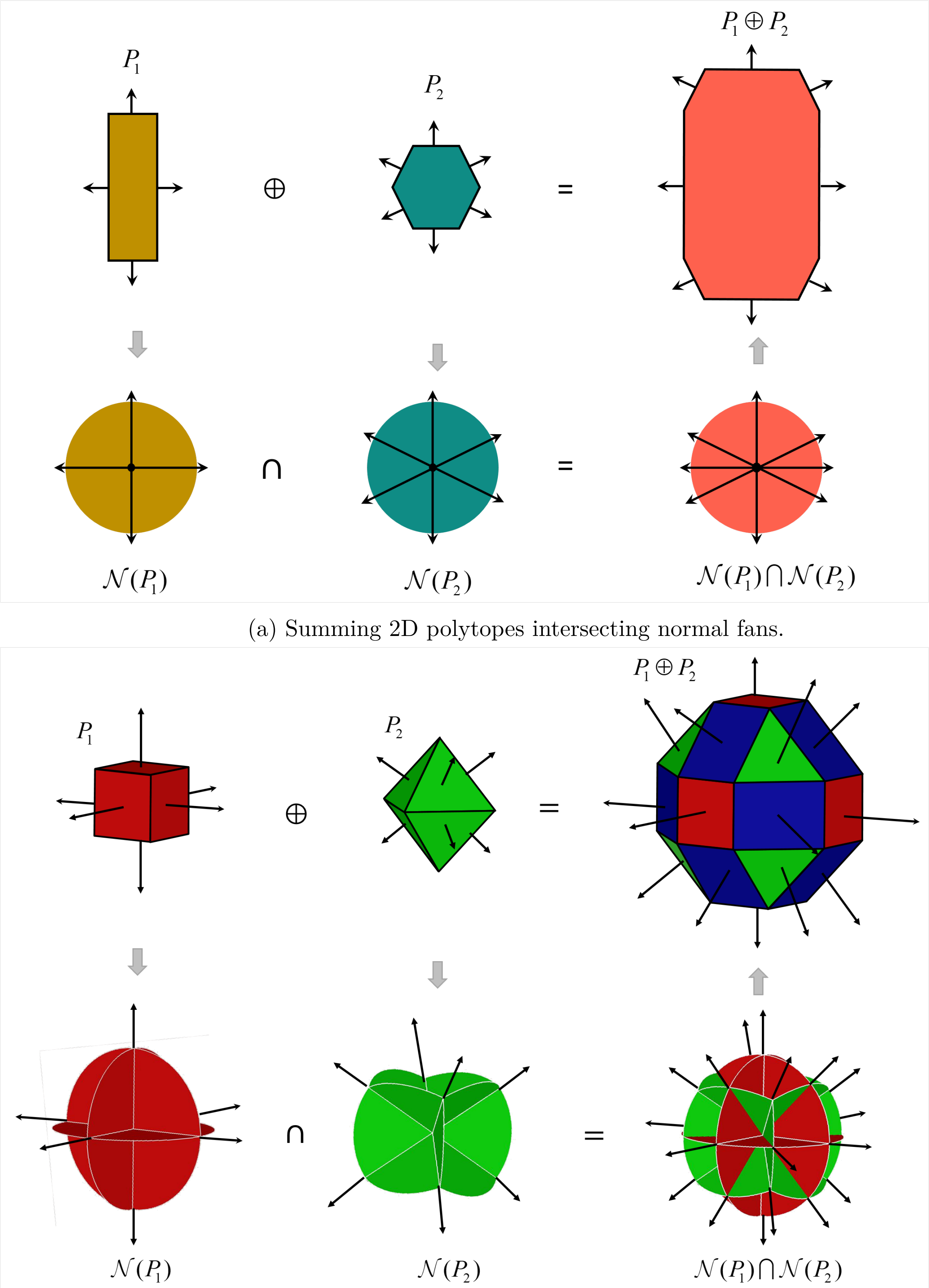
The main objective in tolerance analysis is to conclude whether the functional requirements of a product are satisfied or not according to the propagation of the defects of its components. Specifically, with a strategy based on polytopes it can be verified if the calculated polytope, containing the cumulative stack-up of variations along the dimension chain, fits inside the functional polytope: $P_R \subseteq ?\ P_F$. The evaluation of this inclusion can easily be done if the $\mathcal{HV}$-description of the operands is available. Then, it is required to evaluate if all the vertices of P_R satisfy all half-spaces of P_F:

$$P_R \subseteq P_F \Leftrightarrow \forall \boldsymbol{c} \in \mathcal{V}_R, \forall \{\boldsymbol{x} \in \mathbb{R}^n, b + \boldsymbol{a}\boldsymbol{x} \geq 0\} \in \mathcal{H}_F : b + \boldsymbol{a}\boldsymbol{c} \geq 0 \tag{2.8}$$

An graphical example of an inclusion test is presented in Figure 2.13.

2.4 Truncation algorithm

As exposed before, the approach for geometric tolerancing based on polytopes demands the $\mathcal{H}$-description for computing intersections and the $\mathcal{V}$-description for computing sums. Thus, for the general case, the double description ($\mathcal{HV}$-description) of the operands must be available. Additionally, the fact of deal with the $\mathcal{HV}$-description allows to identify the set of half-spaces



(a) Summing 2D polytopes intersecting normal fans.

(b) Summing 3D polytopes intersecting normal fans.

Figure 2.12: Sum of polytopes intersecting normal fans. Operands assumed to be centred on the origin.

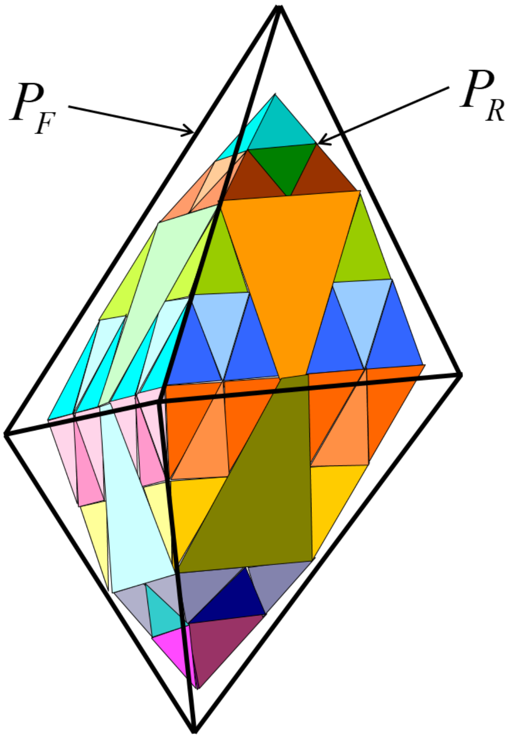


Figure 2.13: Polytopes inclusion test (Teissandier, 2012).

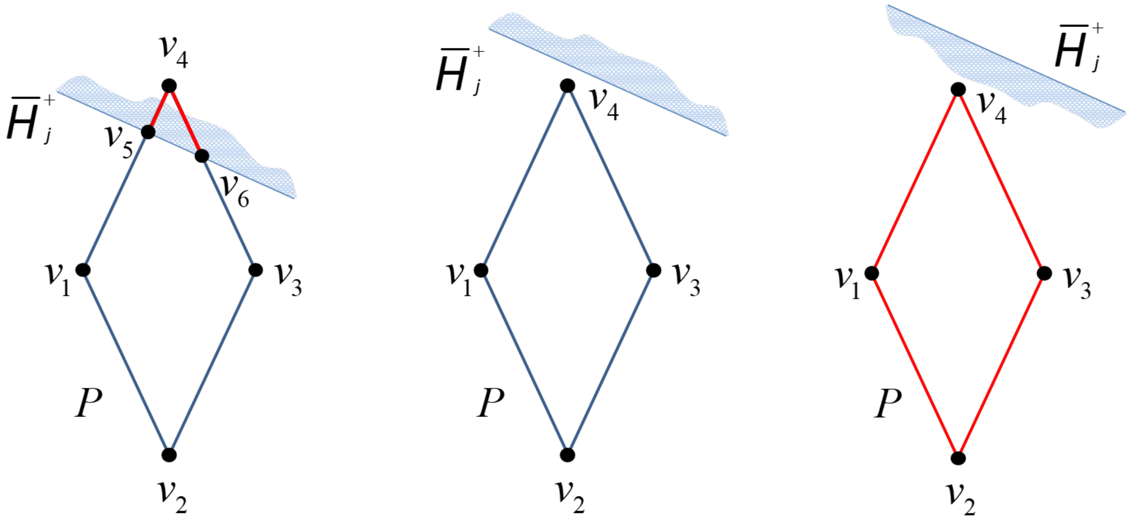
generating each vertex and the set of vertices belonging to each half-space. These relations can also be traced along the different operations of a simulation to identify the vertices and facets of a calculated polytope with respect to the vertices and facets of the operands. It turns out that it is essential to optimize the fitting of a calculated polytope inside a functional polytope.

In response to this, Delos and Teissandier (2015a) proposed using a double description algorithm for performing vertex and facet enumeration of polytopes in any dimension. The idea of the algorithm is to start with big hypercube K, which represents the whole space $\mathbb{R}^n$, and systematically chop it with each half-space $\bar{H}_i^+$ defining a polytope P. The definition of the size of this hypercube is not a problem in tolerancing analysis as we work in the space of small displacements. Evaluating if $\bar{H}_i^+$ intersects (formally separates) K, three cases can occur:

- $\bar{H}_i^+$ separates K, i.e. $K \not\subset H_i^-$ and $K \not\subset \bar{H}_i^+$ (see Figure 2.14a). In this case the new vertices have to be found. A new vertex $\boldsymbol{v_{hk}}$ is generated when an edge e_h of K has one vertex $\boldsymbol{v_{ha}} \in \bar{H}_i^+$ and the other one $\boldsymbol{v_{hb}} \in H_i^-$.

- $\bar{H}_i^+$ does not separates K and $K \subset \bar{H}_i^+$ (see Figure 2.14b). In this case, $\bar{H}_i^+$ is a redundant half-space that must be removed from the list.

- $\bar{H}_i^+$ does not separates K and $K \subset H_i^-$ (see Figure 2.14c). In this case, the result is an empty set.



(a) Separation half-space. (b) Redundant half-space. (c) Empty set.

Figure 2.14: Definition of separation half-space (Arroyave-Tobón et al., 2017c).

Algorithm 1 Truncation

```
Require: An hypercube $K$ and the list of half-spaces of $P$, $\mathcal{H}_P$
Ensure: The list of vertices of $P$, $\mathcal{V}_P$
for each half-space $\bar{H}_i^+ \in \mathcal{H}_P$ do
    if $K \subset H_i^-$ then                                // empty set
        $\mathcal{V}_P = \emptyset$
        break
    else if $K \subset \bar{H}_i^+$ then                     // redundant half-space
        remove $\bar{H}_i^+$ from $\mathcal{H}_P$
    else                                                     // separation half-space
        for each edge $e_h = (\boldsymbol{v_{ha}}, \boldsymbol{v_{hb}}) \in K$ do
            if $\boldsymbol{v_{ha}} \in H_i^-$ and $\boldsymbol{v_{hb}} \in H_i^-$ then
                remove $\boldsymbol{v_{ha}}$ and $\boldsymbol{v_{hb}}$ from $\mathcal{V}_P$
            else if $\boldsymbol{v_{ha}} \in \bar{H}_i^+$ and $\boldsymbol{v_{hb}} \in H_i^-$ then
                compute intersection $\boldsymbol{v_{hk}} = e_h \cap H_i$
                add $\boldsymbol{v_{hk}}$ to $\mathcal{V}_P$
                remove $\boldsymbol{v_{hb}}$ from $\mathcal{V}_P$
            end if
        end for
    end if
end for
```

This procedure, formalized in Algorithm 1, is illustrated in Figure 2.15. All the steps of the truncation of a polytope with a set of constraints derived from a planar surface with a rectangular boundary are depicted. The boundary of the surface is discretized in 4 points and then two constraints per point are obtained $\mathcal{H}_P = \{\bar{H}_1^+, ..., \bar{H}_8^+\}$. The constraints are expressed at the centroid of the surface. The input is the hypercube K and the output is the list of vertices of P, $\mathcal{V}_P$. Both, $\mathcal{H}_P$ and $\mathcal{V}_P$ represent the $\mathcal{HV}$-description of P.

In Figure 2.16, the truncated polytope P is shown. As shown, each vertex is related to its corresponding extreme positions of the related feature. For this case, if the value of the tolerance zone is augmented, the polytope will be scaled according to the ratio; and if the length and width of the toleranced surface are increased, the polytope will become more slender due to the decrease in r_x and r_y.

2.4.1 Computing intersections and sums

The former algorithm can also be used to compute the intersection of two polytopes. In this case, the initial hypercube K is replaced by one of the two operands. The $\mathcal{H}$-description of the other one is then used to carry out the truncation. If it exists, the polytope satisfying the $\mathcal{H}$-description of both operands is calculated. As the intersection of polytopes is a commutative operation, the choice of the operand to be truncated does not affect the result.

As explained in the previous section, the method for summing $\mathcal{HV}$-polytopes is based on the intersection of polyhedral cones. Therefore, the truncation algorithm can also be used for this purpose. In short, the truncation algorithm is the base of the approach developed at the I2M laboratory for tolerance analysis with polytopes (as it is depicted in Figure 2.17).

2.4.2 PolitoCAT and politopix software tools

The tolerancing process following the above described approach is detailed in Figure 2.18. This process can be carried out by means of the software tools PolitoCAT and politopix, available at `http://i2m.u-bordeaux.fr/politopix/` under the LGPL license.

By means of its graphical interface, the software allows the user to import CAD models in formats as **.step*, **.iges*, **.stl*, among others. Directly from the CAD model, operand polytopes can be created. Truncations, intersections and sums of polytopes can be computed with politopix. Finally, the designer can import the results in the graphical interface to analyze them. This process is illustrated in Figure 2.19.

2.5 Polytopes and cap half-spaces

Due to the aforementioned difficulties treating polyhedra, Homri et al. (2015) proposed bounding the original set of constraints with virtual boundaries called cap half-spaces to avoid the manipulation of vertices placed at infinity. An example of the Minkowski sum of a polyhedron Γ_1 and a polytope P_2 following this technique is presented in Figure 2.20. In this case, Γ_1 is

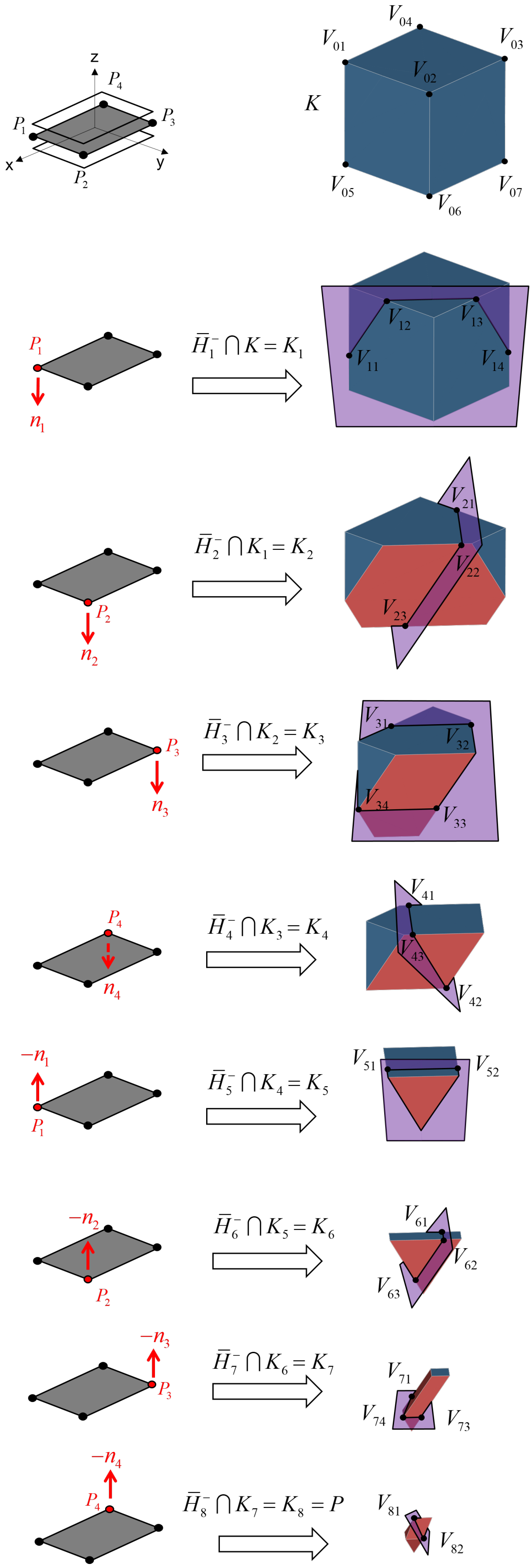


Figure 2.15: Example of the truncation process (Arroyave-Tobón et al., 2017c).

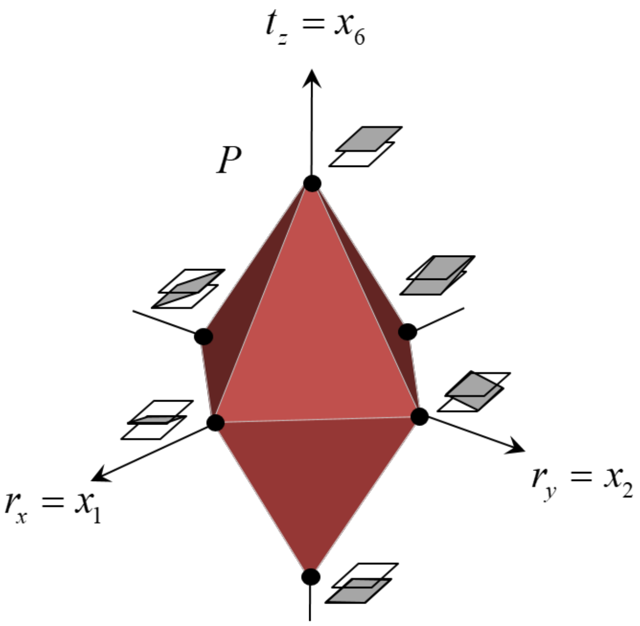


Figure 2.16: Truncated polytope (Arroyave-Tobón et al., 2017c).

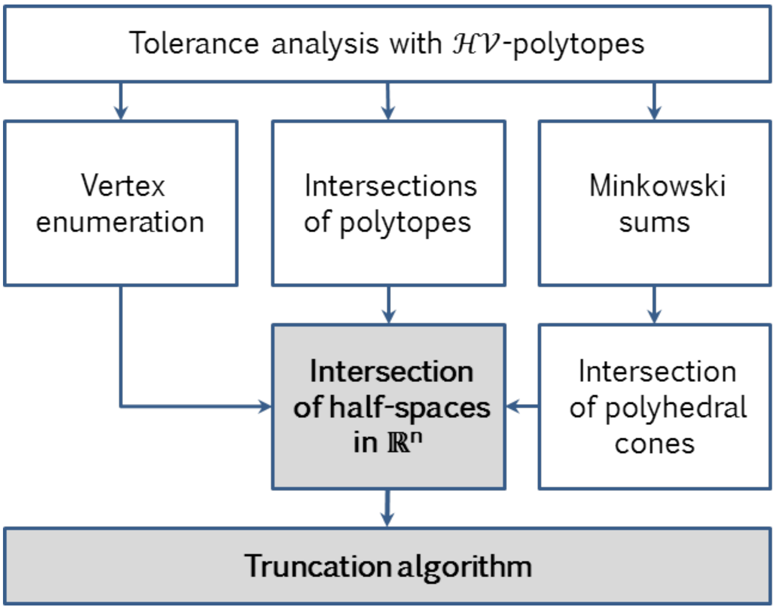

Tolerance analysis with $\mathcal{HV}$-polytopes
Vertex enumeration
Intersections of polytopes
Minkowski sums
Intersection of half-spaces in $\mathbb{R}^n$
Intersection of polyhedral cones
Truncation algorithm

Figure 2.17: Truncation algorithm in tolerance analysis (Arroyave-Tobón et al., 2017c).

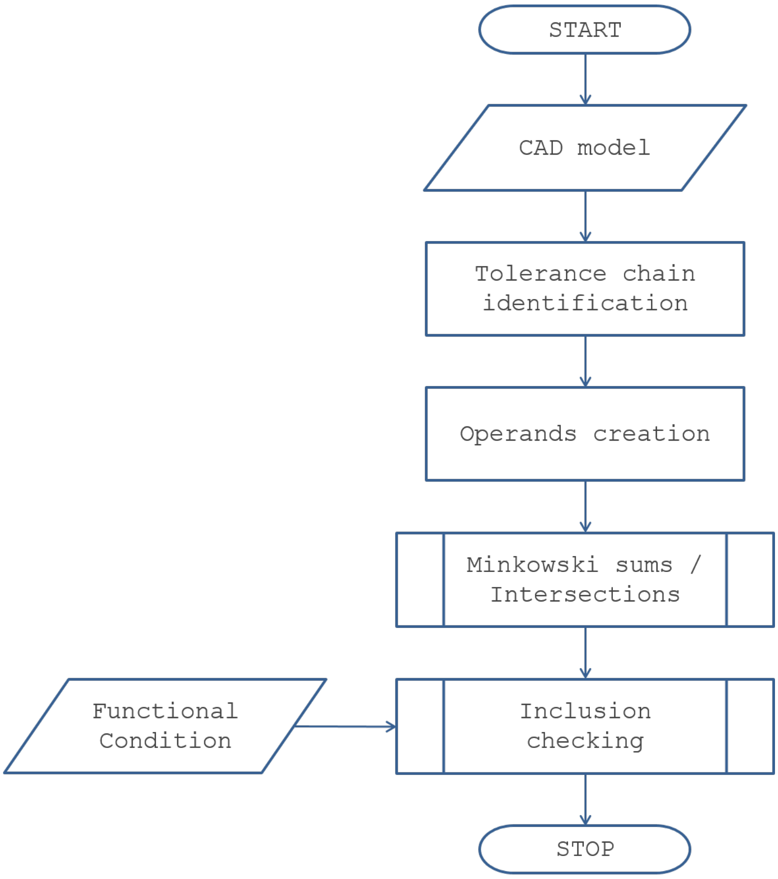
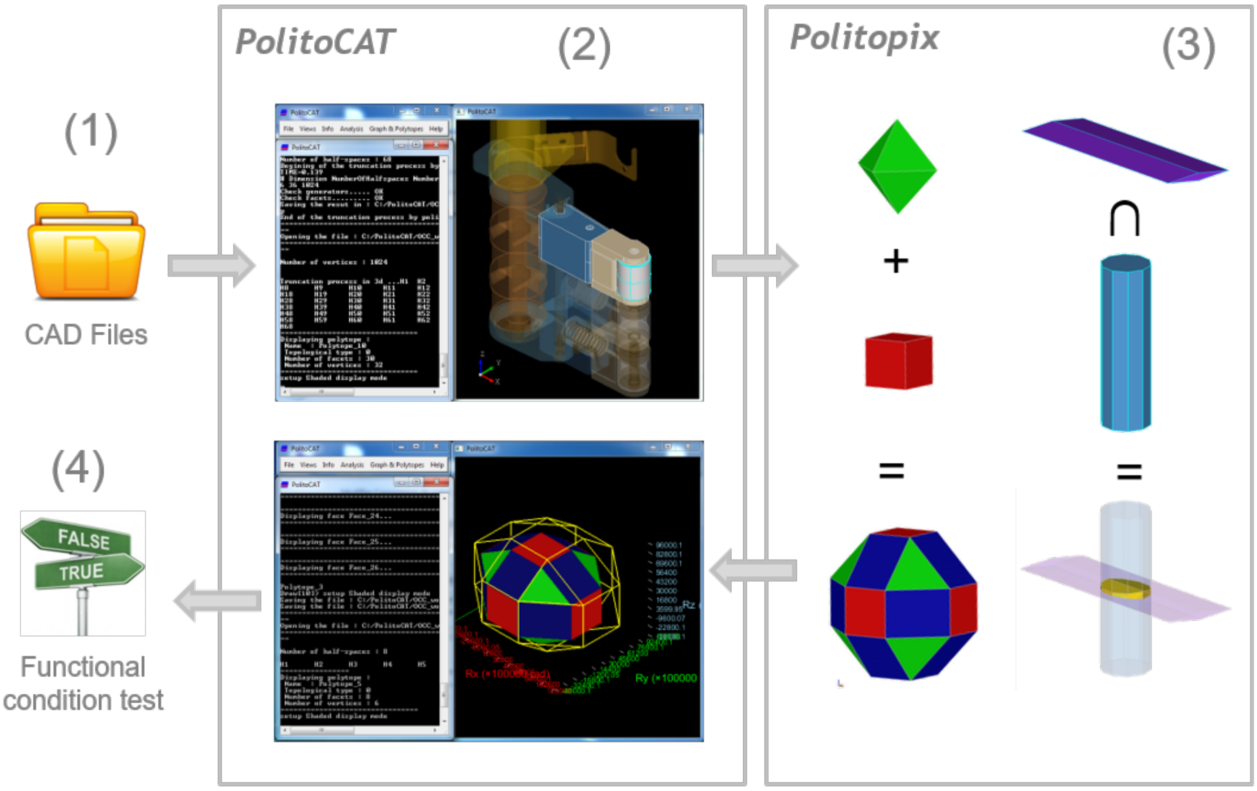


Figure 2.18: Flowchart of the current methodology for tolerance analysis with polytopes.



```
Number of half-spaces : 68
Begining of the truncation process by
TIME=0.139
# Dimension NumberOfHalfspaces Number
6 36 1024
Check generators...... OK
Check facets.......... OK
Saving the resut in : C:/PolitoCAT/OC
2
End of the truncation process by poli
=====================================
--
Opening the file : C:/PolitoCAT/OCC_w
=====================================
--

Number of vertices : 1024


Truncation process in 3d ...H1   H2
H8       H9       H10      H11      H12
H18      H19      H20      H21      H22
H28      H29      H30      H31      H32
H38      H39      H40      H41      H42
H48      H49      H50      H51      H52
H58      H59      H60      H61      H62
H68
=================================
Displaying polytope :
 Name  : Polytope_10
 Topological type : 0
 Number of facets : 30
 Number of vertices : 32
=================================
setup Shaded display mode
```

```
=====================================

=====================================
Displaying face Face_24...
=====================================

=====================================
Displaying face Face_25...
=====================================

=====================================
Displaying face Face_26...
=====================================

Polytope_3
Draw(101) setup Shaded display mode
Saving the file : C:/PolitoCAT/OCC_wo
Saving the file : C:/PolitoCAT/OCC_wo
=====================================
--
Opening the file : C:/PolitoCAT/OCC_w
=====================================
--

Number of half-spaces : 8

H1       H2       H3       H4       H5
=================
Displaying polytope :
 Name  : Polytope_5
 Topological type : 0
 Number of facets : 8
 Number of vertices : 6
=================================
setup Shaded display mode
```

(1) Importation of CAD parts
(2) Creation of operand polytopes
(3) Computation of operations
(4) Analysis and validation of results

Figure 2.19: Tolerance analysis process using PolitoCAT and politopix software tools.

bounded with two cap half-spaces $\bar{H}c_1^+$ and $\bar{H}c_2^+$ to obtain a polytope Γ_1' and be able to use the algorithm based on normal fans intersection.

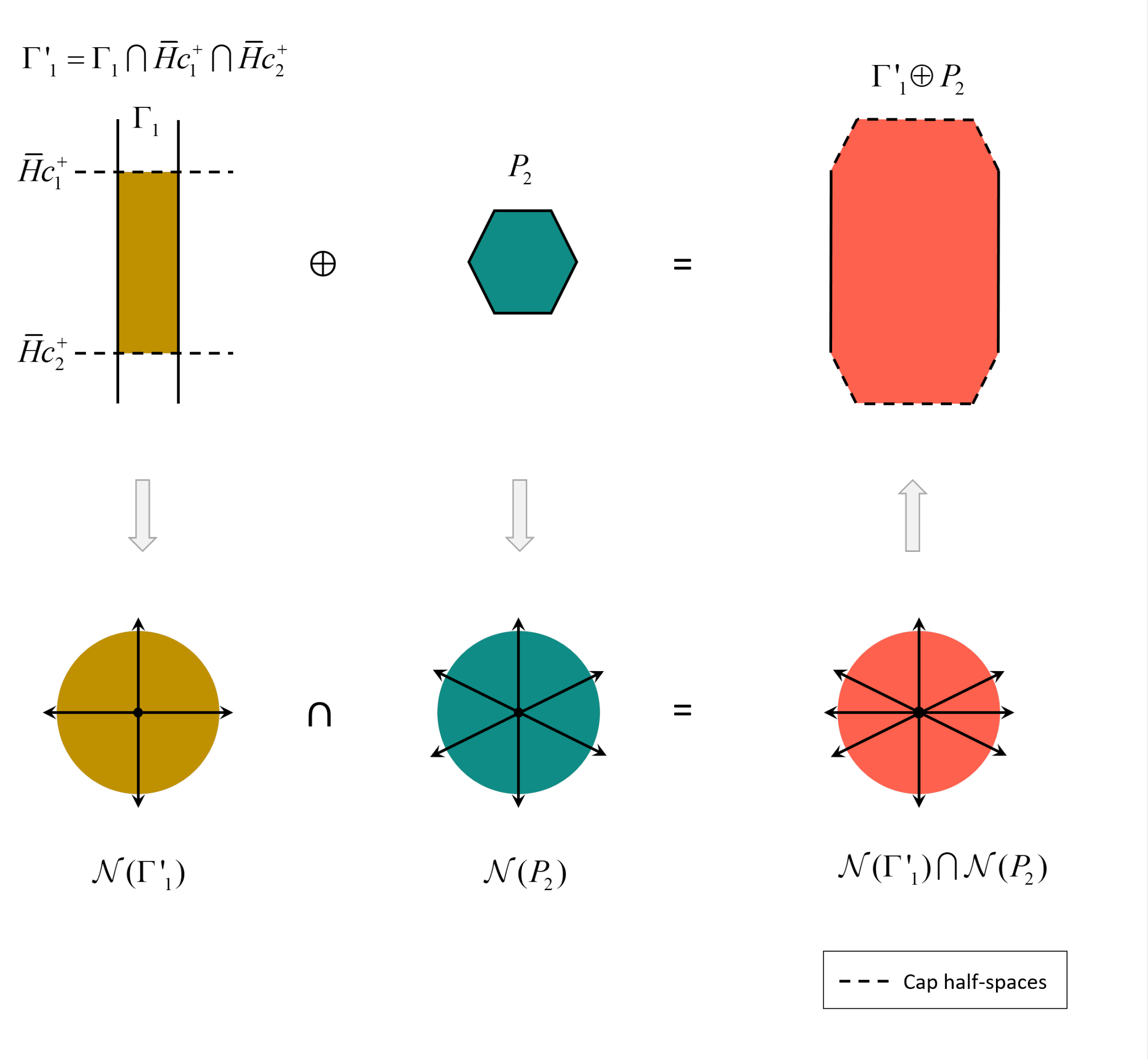


Figure 2.20: Summing 2D capped polytopes intersecting normal fans.

In the context of geometric tolerancing, these half-spaces virtually limit the unbounded displacements of the related feature or joint. Therefore, they can be characterized according to the joint type or the class of the surface and their situation elements. The idea is to introduce the strict minimum of cap facets to each polyhedron:

$$\Gamma' = \left(\bigcap_{k=1}^{k_{max}} \bar{H}_k^+ \right) \cap \left(\bigcap_{j=1}^{2d} \bar{H}c_j^+ \right) = \Gamma \cap \left(\bigcap_{j=1}^{2d} \bar{H}c_j^+ \right) \tag{2.9}$$

where d is the number of degrees of invariance (or freedom) of the toleranced (or kinematic) feature.

An example of introduction of cap facets in a case derived from a tolerancing problem is presented in Figure 2.21. In this figure, a 6D polyhedron derived from the geometric constraints of a plane surface is depicted in two partial 3D representations. The bottom left polyhedron is a partial visualization of the rotations and the right is one of the translations. As the tolerance zone imposes limits only on r_x, r_y and t_z, cap facets are required to virtually limit the unbounded displacements r_z (in the polyhedron at the left-hand side), t_x and t_y (in the polyhedron at the right-hand side). As aforementioned, the decision to add cap half-space to obtain bounded sets is only due to algorithmic reasons.

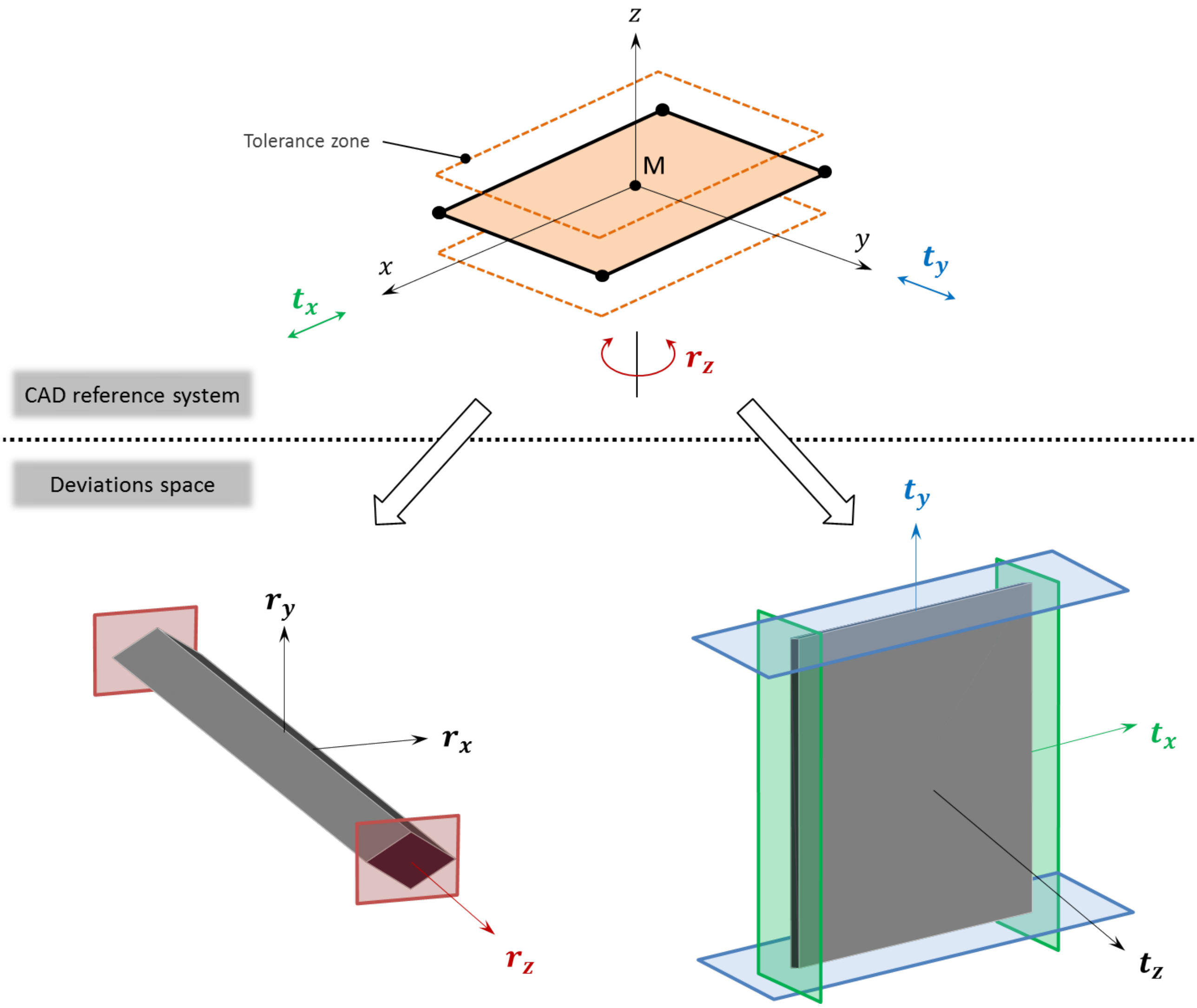


Figure 2.21: Cap half-spaces to virtually limit r_z, t_x and t_y depicted in red, blue and green respectively (Arroyave-Tobón et al., 2017a).

Even if this strategy allows to compute sums, its brings as consequence an alteration of the topology of the operand polytopes, increasing their complexity. Although such an issue does not influence the results from the tolerancing point of view, it does involve spending time calculating meaningless information. During a simulation, this problem increases after each sum due to the accumulation of the DOFs along the toleranced chains. Therefore, after the first sum, the definition of the set of cap facets of the operand polytopes is no longer minimal. As a consequence, the time for computing cap facets is in general far greater than that for computing significant facets. This phenomena is illustrated further in Section 2.6.

2.6 Case study: solution by caps-based method

In order to illustrate the concepts described along this chapter, a tolerance analysis of a braking system is presented. The assembly is made up of six parts, as depicted in Figure 2.22. The braking performance is directly correlated with the parallelism between the two planar surfaces of the brake shoes which are in contact with the disc (surfaces 1,1 and 6,1 in Figure 2.23).

As it can be noticed in the section view presented in Figure 2.23, this example has the particularity of being over-constrained. Three couples of pin-holes clamps generate redundantly suppressed DOFs. For this reason, it is not possible to solve it directly by means of parametric approaches without restrictive assumptions.

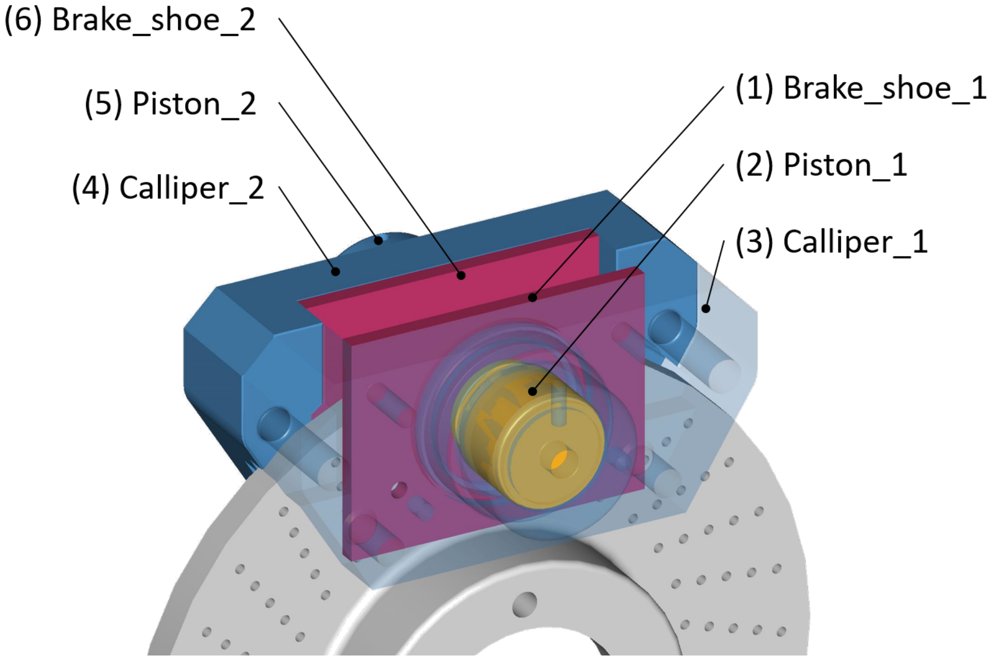


Figure 2.22: Case study: breaking system (Arroyave-Tobón et al., 2017c).

The objective of the simulation, according to the functional condition FC, is to control the relative position of the surfaces 1,1 and 6,1 considering manufacturing and contact defects on the mating parts (see FC in Figure 2.24). More specifically, the relative orientation (variables $[r_x, r_y]$) of these surfaces has to be controlled. This restriction defines the functional polyhedron P_F, which is the one that must be satisfied by the polytope resulting from the deviations propagation along the tolerance chain.

In order to determine the set of operands and operations, the analysis of the topology of the mechanism was initially made. According to the enumeration of the parts and the surfaces presented in Figure 2.23, the topological model of the assembly was created (see Figure 2.24). In this graph, the nodes are designated by two integers (a, b). The first one refers to the part number and the second one to the surface number. The nominal model of a part is represented setting b to 0. The edges in the graph represent some deviations; these may be geometric deviations, in the case of inner edges, 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 (Homri et al., 2015).

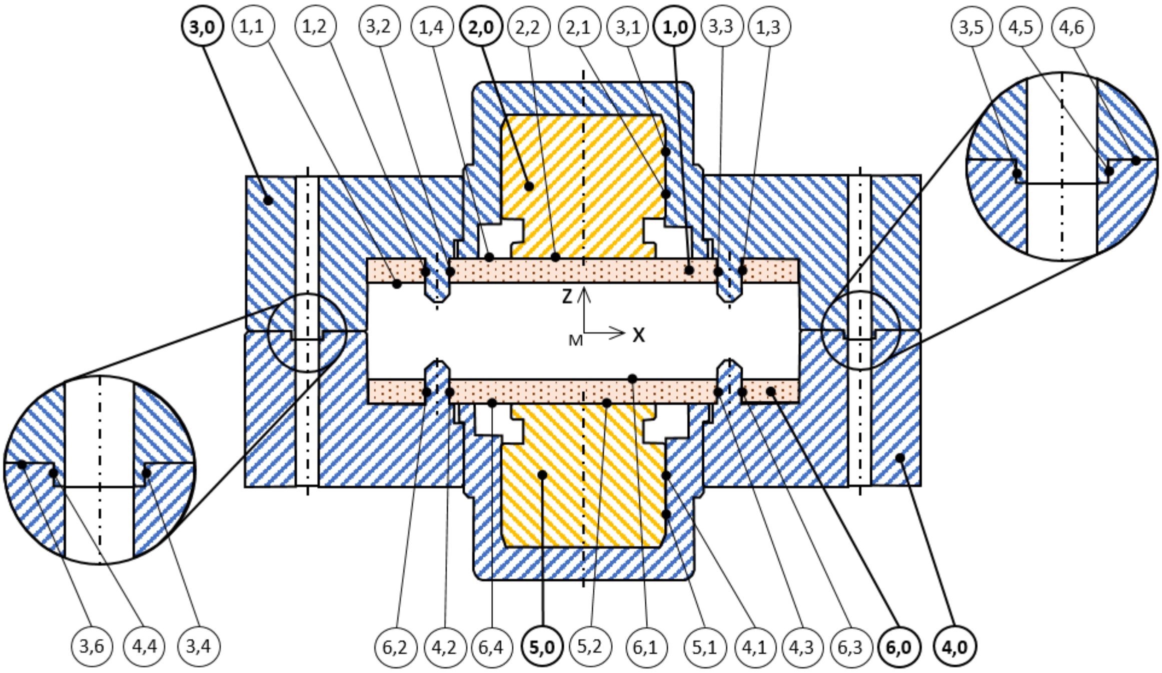


Figure 2.23: Parts and surfaces enumeration (Arroyave-Tobón et al., 2017c).

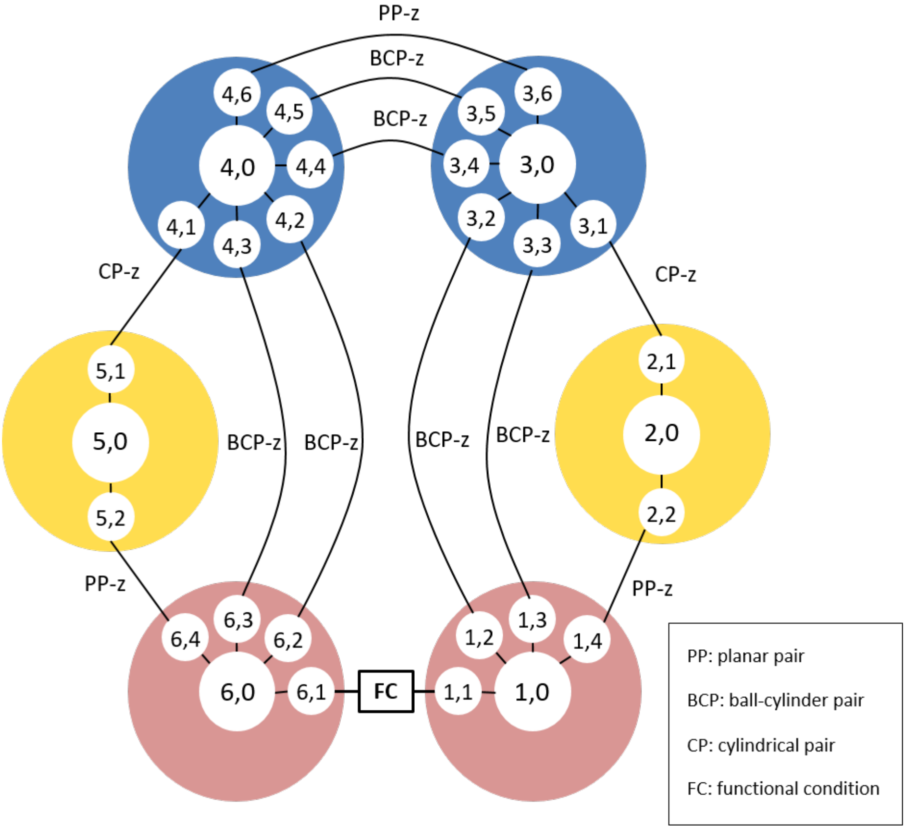


Figure 2.24: Contact graph of the mechanism (Arroyave-Tobón et al., 2017c).

2.6.1 Operands and operations definition

By analyzing the contact graph, the set of operations required to simulate the relative position of the surfaces involved in the functional condition can be determined. The following relation (Eq (2.10)) can then be deduced. In that equation and hereafter, $\Gamma_{a,b/c,d}$ represents a polyhedron describing the relative position of the surface b from the part a with respect to the surface d of the part c..

$$\Gamma_R = \Gamma_{1,1/6,1} = \Gamma_{1,1/1,0} \oplus \Gamma_{1,0/3,0} \oplus \Gamma_{3,0/4,0} \oplus \Gamma_{4,0/6,0} \oplus \Gamma_{6,0/6,1} \tag{2.10}$$

where:

$$\Gamma_{1,0/3,0} = \Gamma_{1,0/3,0-a} \cap \Gamma_{1,0/3,0-b} \cap \Gamma_{1,0/3,0-c} \tag{2.11}$$

with:

$$\Gamma_{1,0/3,0-a} = \Gamma_{1,0/1,2} \oplus \Gamma_{1,2/3,2} \oplus \Gamma_{3,2/3,0} \tag{2.12}$$

$$\Gamma_{1,0/3,0-b} = \Gamma_{1,0/1,3} \oplus \Gamma_{1,3/3,3} \oplus \Gamma_{3,3/3,0} \tag{2.13}$$

$$\Gamma_{1,0/3,0-c} = \Gamma_{1,0/1,4} \oplus \Gamma_{1,4/2,0} \oplus \Gamma_{2,0/3,0} \tag{2.14}$$

$$\Gamma_{3,0/4,0} = \Gamma_{3,0/4,0-a} \cap \Gamma_{3,0/4,0-b} \cap \Gamma_{3,0/4,0-c} \tag{2.15}$$

with:

$$\Gamma_{3,0/4,0-a} = \Gamma_{3,0/3,4} \oplus \Gamma_{3,4/4,4} \oplus \Gamma_{4,4/4,0} \tag{2.16}$$

$$\Gamma_{3,0/4,0-b} = \Gamma_{3,0/3,5} \oplus \Gamma_{3,5/4,5} \oplus \Gamma_{4,5/4,0} \tag{2.17}$$

$$\Gamma_{3,0/4,0-c} = \Gamma_{3,0/3,6} \oplus \Gamma_{3,6/4,6} \oplus \Gamma_{4,6/4,0} \tag{2.18}$$

$$\Gamma_{4,0/6,0} = \Gamma_{4,0/6,0-a} \cap \Gamma_{4,0/6,0-b} \cap \Gamma_{4,0/6,0-c} \tag{2.19}$$

with:

$$\Gamma_{4,0/6,0-a} = \Gamma_{4,0/4,2} \oplus \Gamma_{4,2/6,2} \oplus \Gamma_{6,2/6,0} \tag{2.20}$$

$$\Gamma_{4,0/6,0-b} = \Gamma_{4,0/4,3} \oplus \Gamma_{4,3/6,3} \oplus \Gamma_{6,3/6,0} \tag{2.21}$$

$$\Gamma_{4,0/6,0-c} = \Gamma_{4,0/5,0} \oplus \Gamma_{5,0/6,4} \oplus \Gamma_{6,4/6,0} \tag{2.22}$$

As some polyhedra are defined over the same feature, they are consequently homothetic. The sum of homothetic polyhedra can be performed directly by homothetic transformations and no numerical computation is needed. A sum of homothetic polytopes can be carried out summing the second member of the half-spaces of the operands. This is the case of the operands: $\Gamma_{1,4/2,0}$, $\Gamma_{2,0/3,0}$, $\Gamma_{3,0/4,0-c}$, $\Gamma_{4,0/5,0}$, $\Gamma_{5,0/6,4}$ and $\Gamma_{1,1/1,0}$.

In the case of the ball-cylinder pairs listed next, the DOF of the joints absorb the restricted rotations of the features. Considering this, the whole set of operands can be considered homothetic to the operand derived from the contact feature. This is the case of the operands: $\Gamma_{1,0/3,0-a}$, $\Gamma_{1,0/3,0-b}$, $\Gamma_{3,0/4,0-a}$, $\Gamma_{3,0/4,0-b}$, $\Gamma_{4,0/6,0-a}$ and $\Gamma_{4,0/6,0-b}$

Following the strategy summarized in Section 2.5, the unbounded displacements of the toleranced features were treated through the use of cap half-spaces. Two cap half-spaces were introduced to bound each DOF and obtain 6D polytopes. After truncating a polyhedron $\Gamma_{a,b/c,d}$, we represent the obtained capped polytope as $\Gamma'_{a,b/c,d}$. The operand polytopes required for the simulation were created with the open source software PolitoCAT (Delos and Teissandier, 2015a) (see Table 2.1). Each feature with non-linear boundary was discretized in 8 points, except in the case of the operand $\Gamma_{3,0/4,0-a}$ in which 10 points were used. This number of points implies a good compromise between precision and computation efficiency.

Table 2.1: Summary of the operand polytopes (HS: half-space, TZ: tolerance zone).

Operand	Feature type	Related surfaces	TZ dimension	DOFs	Cap HS	Non-cap HS	Vertices
$\Gamma'_{1,1/1,0}$	Plane	1,1; 6,1	0,1 mm	3	6	8	48
$\Gamma'_{1,0/3,0-a}$	Ball-and-cylinder	1,2; 3,2	0,1 mm	4	8	8	128
$\Gamma'_{1,0/3,0-b}$	Ball-and-cylinder	1,3; 3,3	0,1 mm	4	8	8	128
$\Gamma'_{1,0/1,4}$	Plane	1,4	0,1 mm	3	6	8	48
$\Gamma'_{1,4/2,0}$	Plane	2,2	0,1 mm	3	6	16	80
$\Gamma'_{2,0/3,0}$	Cylinder	2,1; 3,1	0,1 mm	2	4	16	256
$\Gamma'_{3,0/4,0-a}$	Plane	3,6; 4,6	0,1 mm	3	6	20	144
$\Gamma'_{3,0/4,0-b}$	Ball-and-cylinder	3,5; 4,5	0,1 mm	4	8	8	128
$\Gamma'_{3,0/4,0-c}$	Ball-and-cylinder	3,4; 4,4	0,1 mm	4	8	8	128
$\Gamma'_{4,0/6,0-a}$	Ball-and-cylinder	4,2; 6,2	0,1 mm	4	8	8	128
$\Gamma'_{4,0/6,0-b}$	Ball-and-cylinder	4,3; 6,3	0,1 mm	4	8	8	128
$\Gamma'_{6,4/6,0}$	Plane	6,4	0,1 mm	3	6	16	80
$\Gamma'_{5,0/6,4}$	Plane	5,2	0,1 mm	3	6	8	48
$\Gamma'_{4,0/5,0}$	Cylinder	4,1; 5,1	0,1 mm	2	4	16	256

2.6.2 Simulation run

The $\mathcal{HV}$-description of the polytopes, the intersections and the Minkowski sums defined in Eqs. (2.10) to (2.22) were performed with the software politopix (Delos and Teissandier, 2015a). The summary of the operations is presented in Table 2.2.

Table 2.2: Summary of the simulation following the current method (HS: half-space).

Operation	DOFs	Cap HS	Non-cap HS	Vertices	Time [s]
$\Gamma'_{1,0/3,0-ab} = \Gamma'_{1,0/3,0-a} \cap \Gamma'_{1,0/3,0-b}$	3	6	14	128	0,02
$\Gamma'_{1,4/3,0} = \Gamma'_{2,0/3,0} \oplus \Gamma'_{1,4/2,0}$	4	432	12	1 984	2,63
$\Gamma'_{1,0/3,0-c} = \Gamma'_{1,4/3,0} \oplus \Gamma'_{1,0/1,4}$	4	1 048	12	4 104	11,47
$\Gamma'_{1,0/3,0} = \Gamma'_{1,0/3,0-ab} \cap \Gamma'_{1,0/3,0-c}$	1	12	26	512	0,47
$\Gamma'_{3,0/4,0-ab} = \Gamma'_{3,0/4,0-a} \cap \Gamma'_{3,0/4,0-b}$	1	2	28	288	0,02
$\Gamma'_{3,0/4,0} = \Gamma'_{3,0/4,0-ab} \cap \Gamma'_{3,0/4,0-c}$	0	0	36	504	0,06
$\Gamma'_{4,0/6,0-ab} = \Gamma'_{4,0/6,0-a} \cap \Gamma'_{4,0/6,0-b}$	3	6	14	128	0,02
$\Gamma'_{4,0/6,4} = \Gamma'_{4,0/5,0} \oplus \Gamma'_{5,0/6,4}$	4	432	12	1 984	2,53
$\Gamma'_{4,0/6,0-c} = \Gamma'_{4,0/6,4} \oplus \Gamma'_{6,4/6,0}$	4	1 048	12	4 104	8,53
$\Gamma'_{4,0/6,0} = \Gamma'_{4,0/6,0-ab} \cap \Gamma'_{4,0/6,0-c}$	1	12	26	512	0,39
$\Gamma'_{1,1/3,0} = \Gamma'_{1,1/6,1} \oplus \Gamma'_{1,0/3,0}$	4	1 246	12	3 728	7,23
$\Gamma'_{1,1/4,0} = \Gamma'_{1,1/3,0} \oplus \Gamma'_{3,0/4,0}$	4	16 406	28	33 374	751,20
$\Gamma'_{R} = \Gamma'_{1,1/6,1} = \Gamma'_{1,1/4,0} \oplus \Gamma'_{4,0/6,0}$	4	64 400	28	108 860	10 348,51

Computations performed with the library politopix with an Intel Core i7-3740QM.

Let us show in detail the computation of the operand $\Gamma'_{1,0/3,0-c}$. The operands required to carry out this operation are shown in Figure 2.25. The first operation is depicted in Figure 2.26a. A projection into the three-dimensional subspace spanned by $[r_x, r_y, t_z]$ was carried out for visualization purposes; since their original polytopes belong to spaces with a dimension larger than three.

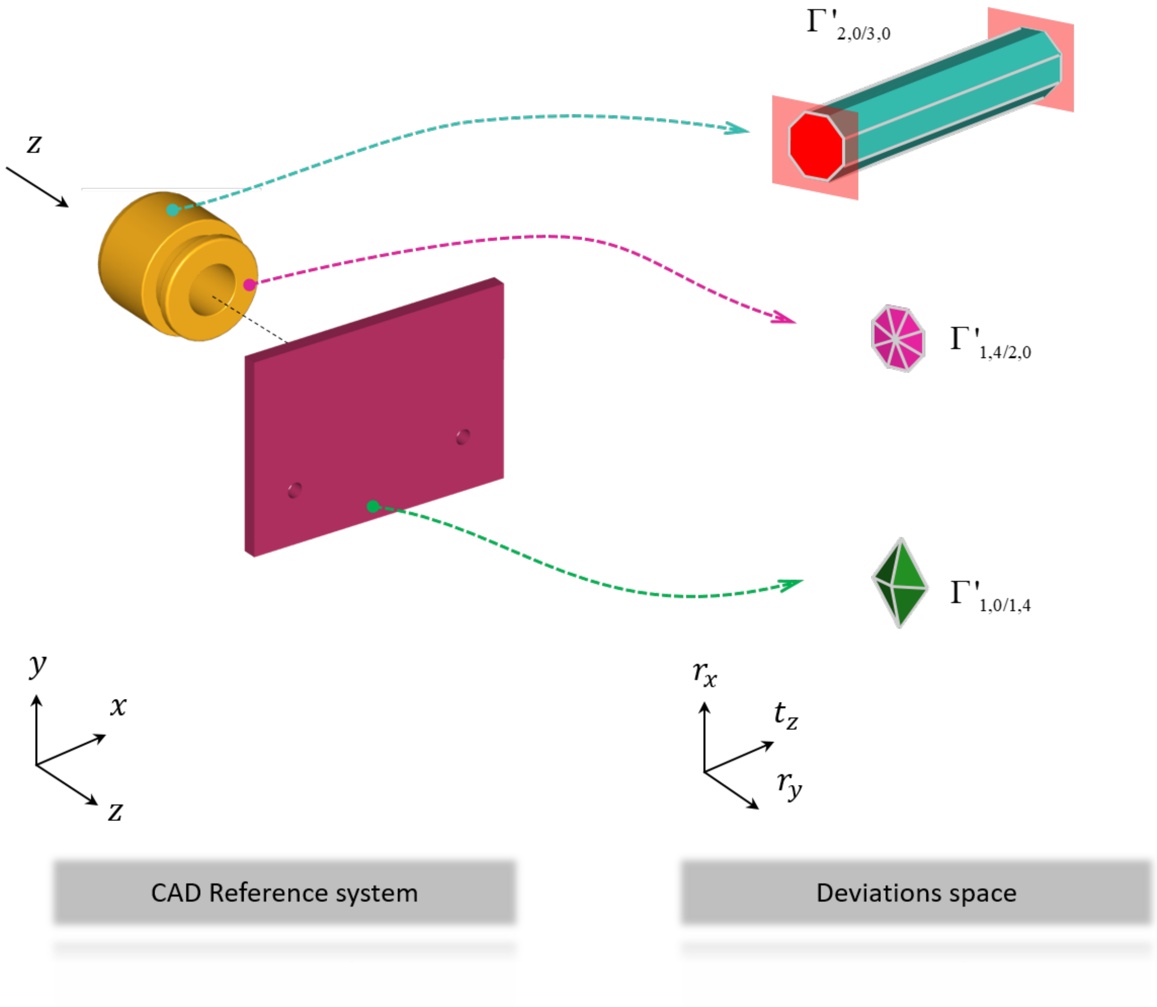
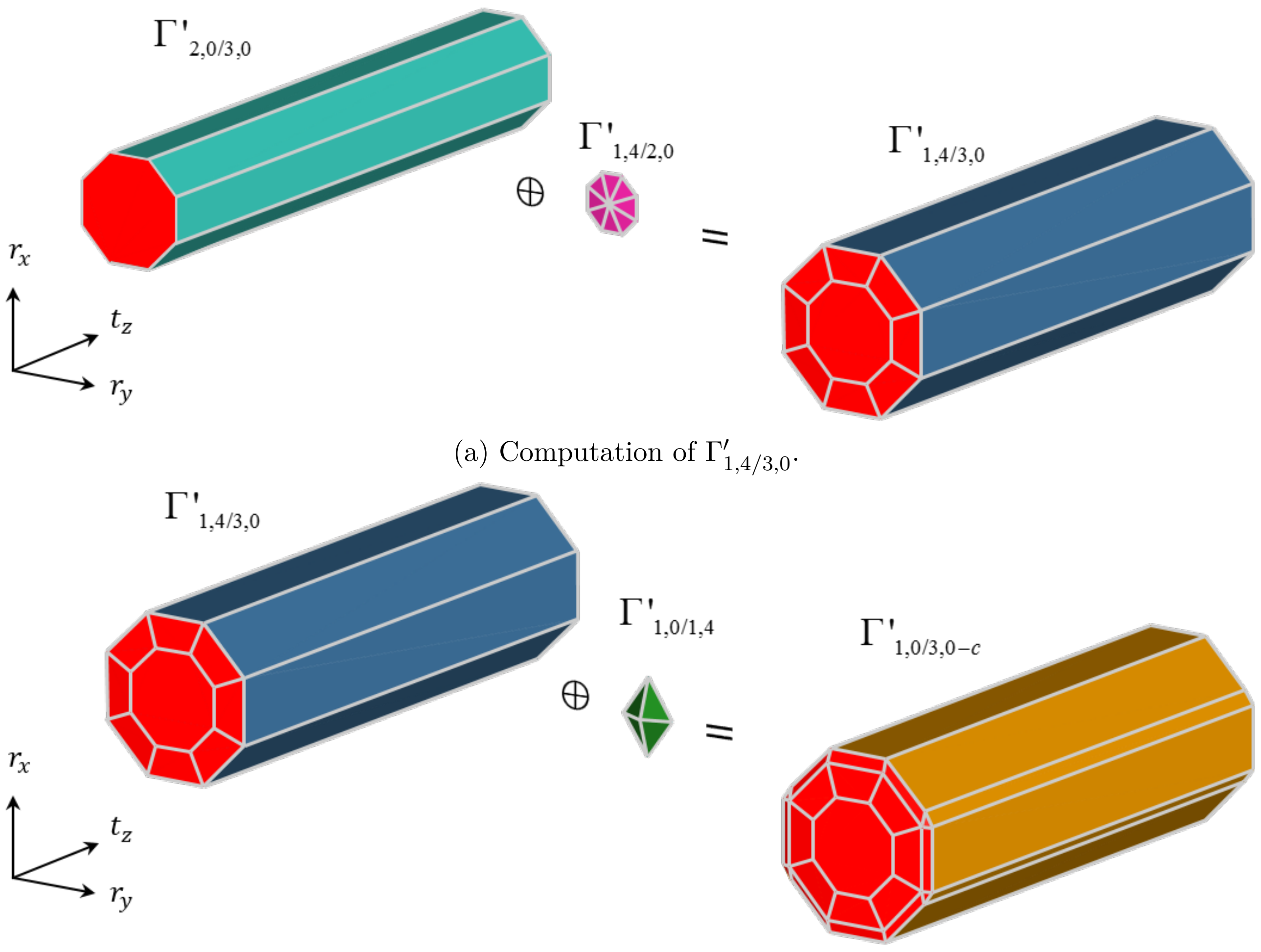


Figure 2.25: 3D representation of some operands. Cap facets are shown in red.

Notice that a strict minimum set of cap facets (facets in red) in the operands, were introduced to bound the sets. However, many cap facets, and consequently many unnecessary vertices, appear in the calculated polytope $\Gamma'_{1,4/3,0}$; having a pair of caps for each DOF is no longer guaranteed. This phenomenon is caused by the propagation of the unconstrained displacement t_z contained in the operand $\Gamma'_{2,0/3,0}$. The result of this sum, which models the possible deviations of a plane surface oriented along the $\boldsymbol{y}$-axis (face 1,4) with respect to a cylindrical surface oriented along the $\boldsymbol{y}$-axis (face 2,1), shows that only two displacements can be controlled: r_x and r_y. This conclusion is justified further on (in Chapter 4) with a kinematic analysis.

This problem, caused by the propagation of the DOFs, worsens when the calculated polytope, highly 'contaminated' with caps, is used again as an operand for a subsequent sum. This is actually the scenario of the operand $\Gamma'_{1,0/3,0-c}$, illustrated in Figure 2.26b. At this point, 1048 caps appear to bound 4 DOFs, where only 8 were enough.



(a) Computation of $\Gamma'_{1,4/3,0}$.

(b) Computation of $\Gamma'_{1,0/3,0-c}$.

Figure 2.26: 3D representation of the computation of $\Gamma'_{1,0/3,0-c}$. Cap facets are shown in red.

The same phenomena appears all along the simulation making it to take more than 3hrs (see Table 2.2). Most of this time was wasted calculating unnecessary data: 99.96% of the calculated facets come from cap half-spaces, and therefore they have no meaning in the related tolerancing problem.

2.6.3 Analysis of results

A 3D representation of the final polytope Γ'_R is presented in Figure 2.27. Some of the cap facets (facets in red) can be seen in this particular projection. The maximal values of the bounded displacements r_x and r_y (displayed on the figure) depend on the topological structure of the mechanism, the geometry of the toleranced features and joints and the dimension of the tolerance zone assumed for each operand. The maximal values along the t_z-axis are not displayed since they depend on the second member of the cap half-spaces and therefore they are of no interest in the simulation.

The functional polytope P_F (derived from the FC), representing the allowable limits in r_x and r_y, is illustrated in Figure 2.28. The calculated polytope Γ'_R was projected into the same subspace in order to compare both. For this particular tolerance scheme, it can be concluded that the functional condition is satisfied. However, the optimal case is when the boundary of Γ'_R touches the boundary of P_F, i.e. when $d = 0$. In the case when $P_F \not\subseteq \Gamma'_R$, the tolerance values of the features and the clearances of the joints have to be tight until reach the fitting.

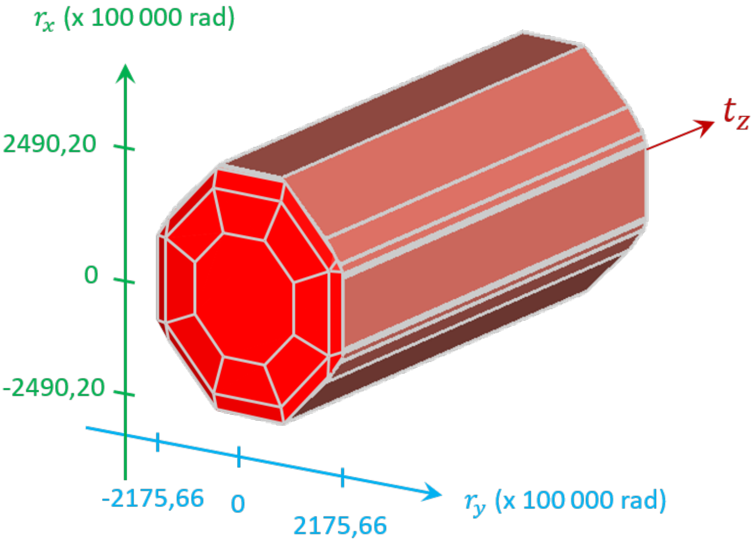


Figure 2.27: 3D representation of Γ'_R.

It is worth mentioning that the point M, in which the constraints were defined, does not influence the obtained results. Changing this point will generate an affine transformation of the operands and consequently of Γ'_R, but also of P_F and if the inclusion is initially satisfied it will be for any other point (Homri, 2014).

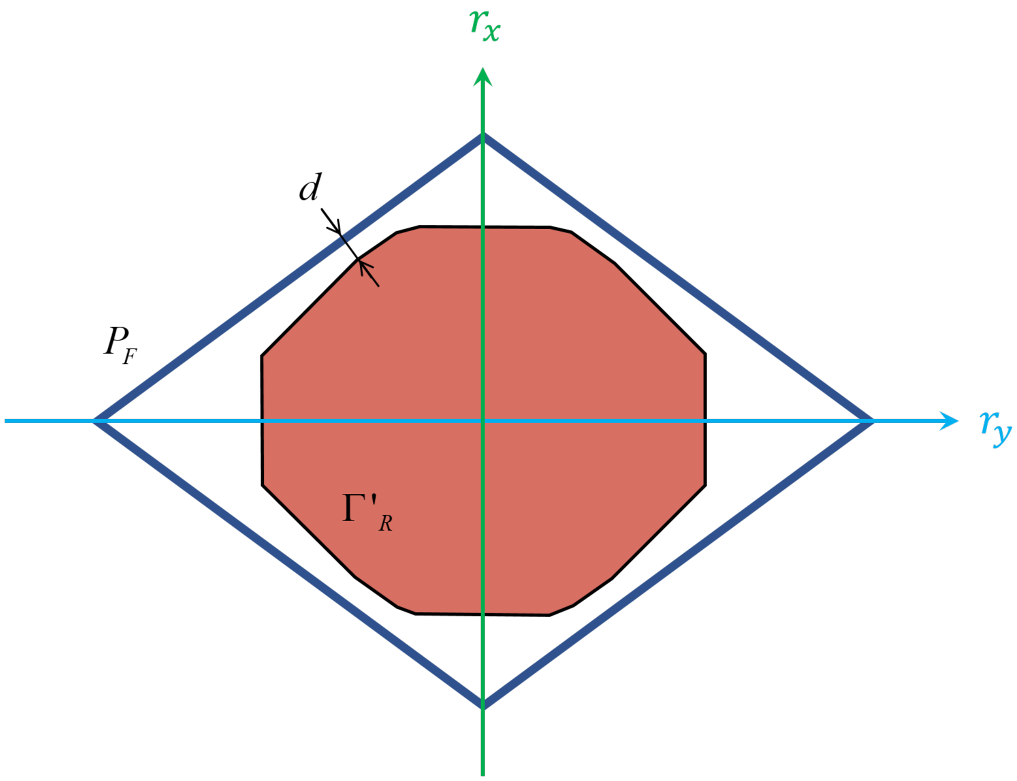


Figure 2.28: Inclusion test.

2.7 Conclusions

In this chapter we explained how the restrictions imposed by the tolerance zones on the tolerance features can be represented by 6-dimensional sets of constraints. As features and joints usually have some unconstrained displacements (those related to the degrees of invariance or freedom), these sets are unbounded polyhedra. We reviewed a strategy to deal with this problem, which consists in imposing additional fictitious limits to the unbounded displacements.

We also showed how the propagation of geometric defects in assemblies can be modelled with operations of $\mathcal{HV}$-polytopes. We presented the truncation algorithm, which is the computational core of these operations.

We illustrated this approach by means of a case study. This consists of a braking system made up of several over-constrained joints. By means of a simulation with $\mathcal{HV}$-polytopes, all the possible configurations of the mechanism (due to the manufacturing and contact defects) were considered simultaneously. This resulted in not only a pair of extreme values, but also a set of all the possible relative positions (the calculated polytope) of the surfaces involved in the functional condition.

As has been shown, this approach has the advantage of being robust enough to treat even over-constrained mechanisms. However, when performing tolerance simulations following this method, a great deal of unnecessary data are computed. These data come from the propagation, along the kinematic chain, of the fictitious bounds (cap constraints) introduced into the joints. We found that the fact of adding bounding or cap half-spaces increases the model's complexity. This complexity increases after each operation until it becomes far too significant, making the approach very time-consuming and error-prone.

In addition to the waste of computational resources, this implies that the results obtained must be analysed to differentiate real displacement limits from fictitious ones (new cap constraints), which could lead to misinterpretations.

Chapter 3

Controlling the effects of DOF propagation

In this chapter, we formalize and test a strategy to deal with the problem caused by DOF propagation when summing sets of geometric constraints. The basic idea is to identify and label the bounding or cap constraints when the operand sets are defined. We suggest tracing the caps during tolerance simulations to keep the generation of new caps under control. The mathematical support for this strategy is based on a formal definition of a cap half-space. Using this definition, cap-spreading rules are formalized for intersections and sums. The advantages of this strategy are illustrated by means of the same case study used in the previous chapter, with the aim of comparing both results.

3.1 Cap half-space definition

Before giving a formal mathematical definition of a cap half-space, we need first to present the formal concept of face of a polytope and the combinatorial representation of a face.

Definition 3.1.1 (Face) *Let P be a polytope of $\mathbb{R}^n$. The intersection of P with a supporting hyperplane H is called a face. H supports P if $H \cap P \neq \emptyset$ and P lies in one of the two closed half-spaces bounded by H. The dimension of a face is the dimension of its affine hull: a 0-face is called a vertex, a 1-face is an edge and in $\mathbb{R}^n$ a $(n-1)$-face is a facet.*

In Figure 3.1, for example, the plane H intersects the polytope P at its frontier and it is completely included in the closed half-space $\bar{H}^+$.

Now, let us write as $\mathcal{L}_P$ the list of all the faces of P and as $\mathcal{L}_P^j$ the list of faces of dimension j only. Then $\mathcal{L}_P^0 = \mathcal{V}_P$ is the list of vertices of P, $\mathcal{L}_P^1$ its list of edges and $\mathcal{L}_P^{n-1}$ its list of facets. Therefore, list of all the faces of P is $\mathcal{L}_P = \bigcup\limits_{j=0}^{n} \mathcal{L}_P^j$, which corresponds to the lattice of P.

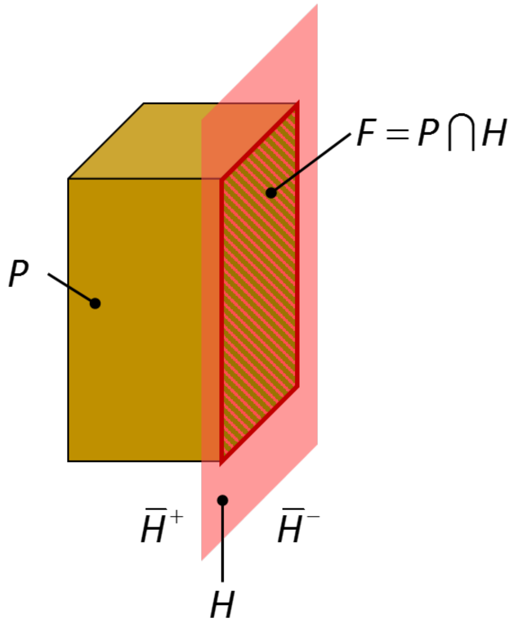


Figure 3.1: Concepts of supporting hyperplane and face of a polytope.

Definition 3.1.2 (Subface) *Let F and F' be two faces of a polytope P. F is a subface of F' if and only if $F' \subset F$*

Definition 3.1.3 (Combinatorial representation (Fukuda and Rosta, 1994)) *The combinatorial representation $\mathcal{CR}(F)$ of a given face F of a polytope P, is the set of facet indices j of $\mathcal{L}_P^{n-1}$ such that F is a subface of F_j.*

For example, the combinatorial representation of the vertex $\boldsymbol{v}$ of the polytope at the right-hand side of Figure 3.2 is: $\mathcal{CR}(\boldsymbol{v}) = \{1, 2, 3\}$.

Based on the former definitions, we formalize the concept of cap half-space.

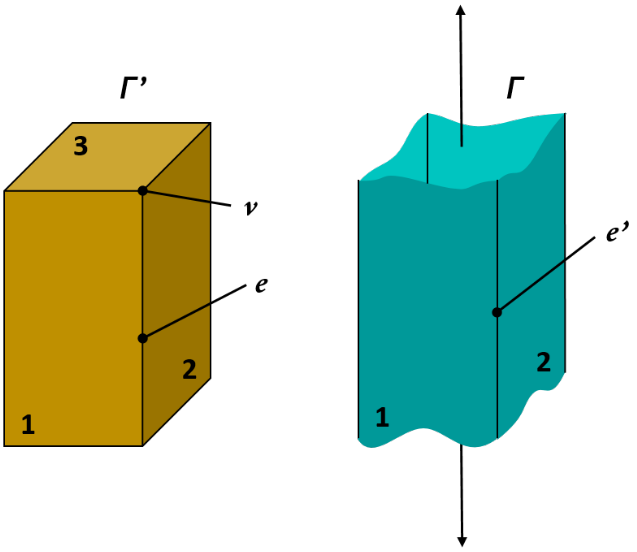


Figure 3.2: Combinatorial representation.

Definition 3.1.4 (Cap half-space) *Let* Γ *be a polyhedron in* $\mathbb{R}^n$*. The half-spaces in the list* $\mathcal{H}^{cap} = \{\bar{H}_u^+, u = 1, ..., w\}$ *are caps for* Γ *if and only if:*

- $\Gamma' = \Gamma \cap (\cap_u \bar{H}_u^+)$ *is a full-dimensional polytope in* $\mathbb{R}^n$
- *the combinatorial representations of the faces of* Γ *are also combinatorial representations of faces of* Γ'*.*

Definition 3.1.5 (Cap facet) *A cap facet is a non-redundant intersection of dimension* $(n-1)$ *of a polytope of* $\mathbb{R}^n$ *with the frontier hyperplane of a cap half-space.*

In Figure 3.1, for example, if $\bar{H}^+$ is a cap half-space, then F is a cap facet.

Definition 3.1.6 (Capped polytope) *A capped polytope is a polytope derived from a polyhedron that has been bounded with cap half-spaces.*

The concept of combinatorial representation is interesting in the sense that it allows to identify the same topological elements whether they belong to a polyhedron or to a capped polytope associated to it. Such a property is illustrated in Figure 3.2, where the edges e and e', belonging to a capped polytope and polyhedron, have the same combinatorial representation: $\mathcal{CR}(e) = \mathcal{CR}(e') = \{1, 2\}$.

By means of former concepts we can characterize next how caps interact during sums, intersections and inclusions tests.

3.2 Tracing caps: Minkowski sums

The sum of unbounded polyhedra is also an unbounded polyhedron. In order words, the unbounded directions of the operands of a Minkowski sum appear also in the calculated polyhedron. When summing capped polytopes, cap facets of the operands appear in the calculated polytope. Furthermore, new cap facets appear due to a 'contamination' phenomenon (as we shown with the case study in Section 2.6).

According to Definition 3.1.4, it is possible to identify the cap half-spaces in a polytope representing a polyhedron. The question now is: when summing two capped polytopes, how do the caps of the calculated polytope can be identified to recover the sets of half-spaces belonging to the calculated polyhedron?

To answer this question we have, first, to ensure that the main characteristics of the polyhedra are preserved by their associated polytopes. This means ensuring that the topological structure of each polyhedron is included inside that of its associated polytope. Thus, the facets belonging to the calculated polyhedron can be differentiated among the cap facets.

Based on some theorems, we describe below the way cap facets spread during Minkowski sums.

3.2.1 Decomposition theorem

According to Fukuda (2004) and Weibel (2007), in a Minkowski sum of polytopes, the faces of the operands generating each face of the calculated polytope can be identified. This property is supported by the following theorem:

Theorem 3.2.1 (Decomposition) *Let P_1 and P_2 be polytopes in $\mathbb{R}^n$, and let F_P be a face of the Minkowski sum $P = P_1 \oplus P_2$. Then there are faces F_{P_1}, F_{P_2} of P_1, P_2 respectively such that $F_P = F_{P_1} \oplus F_{P_2}$. Such a decomposition is unique.*

Figure 3.3 illustrates this property. Notice that although F_P is a facet, F_{P_1} and F_{P_2} might not be. In the general case, F_P, F_{P_1} and F_{P_2} have different dimensions. In $\mathbb{R}^2$ for example, a facet F_P is a 1-face and we have 3 cases:

- F_{P_1} is a vertex (0-face) and F_{P_2} is a facet (1-face). This is equivalent to the translation of the facet.
- F_{P_1} is a facet (1-face) and F_{P_2} is a vertex (0-face). This is equivalent to the translation of the facet.
- F_{P_1} is a facet (1-face) and F_{P_2} is a facet (1-face). This case occurs when F_{P_1} and F_{P_2} are parallel.

3.2.2 Caps propagation theorem

From Definition 3.1.4 and the decomposition theorem, we can describe how cap faces propagate when summing capped polytopes.

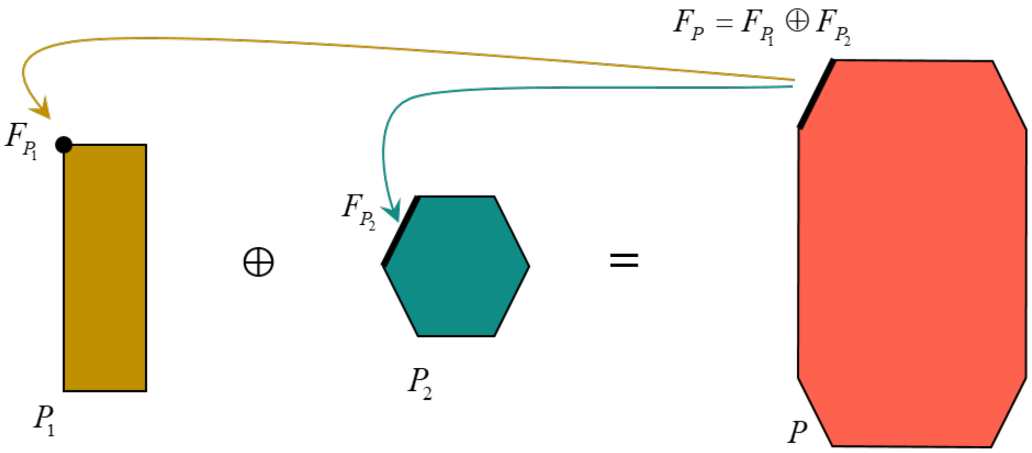


Figure 3.3: Faces decomposition in Minkowski sums.

Theorem 3.2.2 (Caps propagation) *Let Γ_1, Γ_2 be two polyhedra and Γ'_1, Γ'_2 their corresponding capped polytopes. Let $F_{\Gamma'}$ be a facet of $\Gamma'_1 \oplus \Gamma'_2$ and $F_{\Gamma'_1} + F_{\Gamma'_2}$ its decomposition into faces of Γ'_1 and Γ'_2. $F_{\Gamma'}$ has a matching facet $F_\Gamma \in \mathcal{L}^{n-1}_{\Gamma_1+\Gamma_2}$ such that $\mathcal{CR}(F_{\Gamma'}) = \mathcal{CR}(F_\Gamma)$ if and only if the combinatorial representation of $F_{\Gamma'_1}$ and $F_{\Gamma'_2}$ does not contain any cap facet.*

Proof. We have $F_{\Gamma'} \in \mathcal{L}^{n-1}_{\Gamma'_1 \oplus \Gamma'_2}$ such that $F_{\Gamma'} = F_{\Gamma'_1} + F_{\Gamma'_2}$ with no cap facets in the combinatorial representation of $F_{\Gamma'_1}$ and $F_{\Gamma'_2}$. It means that $\exists F_{\Gamma_1} \in \mathcal{L}_{\Gamma_1}, \exists F_{\Gamma_2} \in \mathcal{L}_{\Gamma_2}$ such that

$$\begin{cases} \mathcal{CR}(F_{\Gamma'_1}) = \mathcal{CR}(F_{\Gamma_1}) \text{ and } C_D(F_{\Gamma_1}) = C_D(F_{\Gamma'_1}) \\ \mathcal{CR}(F_{\Gamma'_2}) = \mathcal{CR}(F_{\Gamma_2}) \text{ and } C_D(F_{\Gamma_2}) = C_D(F_{\Gamma'_2}) \end{cases}$$

So if $\boldsymbol{u}$ is the normal to the hyperplane supporting $F_{\Gamma'_1} + F_{\Gamma'_2}$, $\boldsymbol{u} = C_D(F_{\Gamma'_1}) \cap C_D(F_{\Gamma'_2}) = C_D(F_{\Gamma_1}) \cap C_D(F_{\Gamma_2})$. As a consequence, $F_{\Gamma_1} + F_{\Gamma_2} \in \mathcal{L}^{n-1}_{\Gamma_1 \oplus \Gamma_2}$ and $F_{\Gamma_1} + F_{\Gamma_2}$ is the matching facet of $F_{\Gamma'} = F_{\Gamma'_1} + F_{\Gamma'_2}$.

The reciprocal is straightforward: let us assume $F_{\Gamma'}$ has a matching facet in $\Gamma_1 + \Gamma_2$, as a consequence $\exists F_\Gamma \in \mathcal{L}^{n-1}_{\Gamma_1+\Gamma_2}$ such that $F_{\Gamma'} /\!/ F_\Gamma$. Let us decompose the last facet, $\exists F_{\Gamma_1} \in \mathcal{L}_{\Gamma_1}, \exists F_{\Gamma_2} \in \mathcal{L}_{\Gamma_2}$ such that $F_\Gamma = F_{\Gamma_1} + F_{\Gamma_2}$. From definition 3.1.4, $\exists F_{\Gamma'_1} \in \mathcal{L}_{\Gamma'_1}, \exists F_{\Gamma'_2} \in \mathcal{L}_{\Gamma'_2}$ such that

$$\begin{cases} \mathcal{CR}(F_{\Gamma'_1}) = \mathcal{CR}(F_{\Gamma_1}) \text{ and } C_D(F_{\Gamma_1}) = C_D(F_{\Gamma'_1}) \\ \mathcal{CR}(F_{\Gamma'_2}) = \mathcal{CR}(F_{\Gamma_2}) \text{ and } C_D(F_{\Gamma_2}) = C_D(F_{\Gamma'_2}) \end{cases}$$

If $\boldsymbol{u}$ is the normal to the hyperplane supporting F_Γ we knew that $\boldsymbol{u} = C_D(F_{\Gamma_1}) \cap C_D(F_{\Gamma_2})$, now we can write $\boldsymbol{u} = C_D(F_{\Gamma'_1}) \cap C_D(F_{\Gamma'_2})$. So $F_{\Gamma'} = F_{\Gamma'_1} + F_{\Gamma'_2}$ and no half-space in the combinatorial representation of $F_{\Gamma'_1}$ or $F_{\Gamma'_2}$ is capped as $\mathcal{CR}(F_{\Gamma'_1}) = \mathcal{CR}(F_{\Gamma_1})$ and $\mathcal{CR}(F_{\Gamma'_2}) = \mathcal{CR}(F_{\Gamma_2})$.

By means of the previous theorem it can be understood why the number of cap facets soars when summing capped polytopes. For example, in face $F_{\Gamma'} = F_{\Gamma'_1} + F_{\Gamma'_2}$ having only one cap facet in the combinatorial representation of $F_{\Gamma'_1}$ or $F_{\Gamma'_2}$ is enough to transfer this property to the sum $F_{\Gamma'}$.

3.2.3 Algorithm: tracing caps in sums

Based on the former rule about caps spreading, we propose Algorithm 2. It works decomposing each facet of the calculated polytope in faces of its operands: $F_{\Gamma'} = F_{\Gamma'_1} + F_{\Gamma'_2}$. This can be done computing lists of vertices belonging to $F_{\Gamma'_1}$ and $F_{\Gamma'_2}$, $\mathcal{V}(\Gamma'_1)$ and $\mathcal{V}(\Gamma'_2)$. For each of these vertices, the list of its supporting half-spaces are identified for building the combinatorial representation of $F_{\Gamma'_1}$ and $F_{\Gamma'_2}$. The final step checks whether if there exists at least one cap facet in the combinatorial representation of $F_{\Gamma'_1}$ or $F_{\Gamma'_2}$. If this is the case, $F_{\Gamma'}$ is marked as a cap facet, if not, $F_{\Gamma'}$ can be matched directly with a facet of Γ. Figure 3.4 presents an example of this procedure.

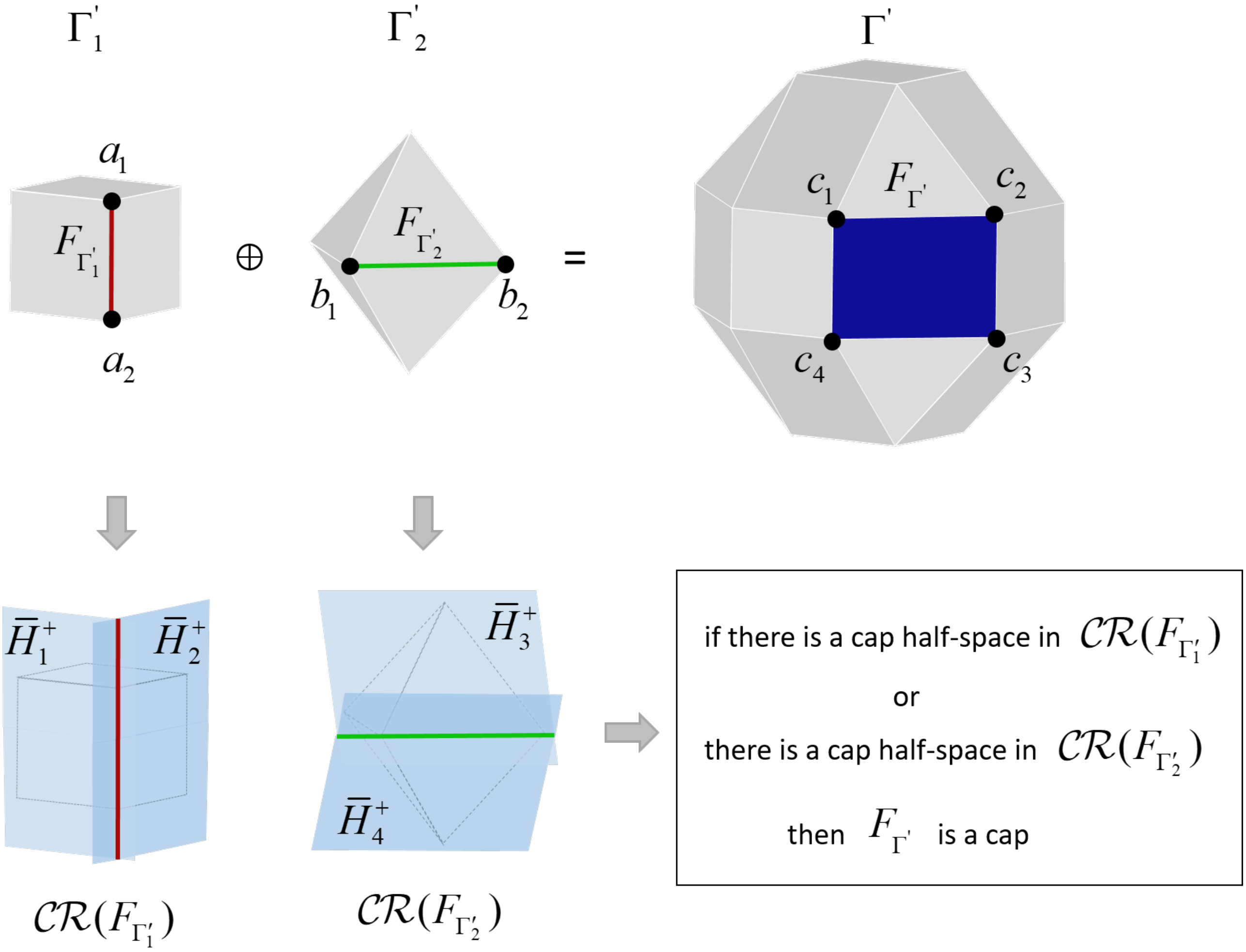

Γ'_1
Γ'_2
Γ'
a_1
$F_{\Gamma'_1}$
a_2
$\oplus$
$F_{\Gamma'_2}$
b_1
b_2
=
c_1
$F_{\Gamma'}$
c_2
c_4
c_3
$\bar{H}_1^+$
$\bar{H}_2^+$
$\bar{H}_3^+$
$\bar{H}_4^+$
$\mathcal{CR}(F_{\Gamma'_1})$
$\mathcal{CR}(F_{\Gamma'_2})$
if there is a cap half-space in $\mathcal{CR}(F_{\Gamma'_1})$
or
there is a cap half-space in $\mathcal{CR}(F_{\Gamma'_2})$
then $F_{\Gamma'}$ is a cap

Figure 3.4: Illustration of Algorithm 2.

3.2.4 Caps removal

There exists infinite possibilities to turn a polyhedron into a polytope. As we explained in the former chapter, we are interested in handling the less complex capped polytope in order to limit the computation of worthless data and speed up the simulations. The fact of having operands with a minimal set of caps does not warranty that their sum generate a polytope also with a minimal set of caps.

Algorithm 2 Trace of cap facets in a sum

```
Require: $\mathcal{V}(\Gamma'_1), \mathcal{V}(\Gamma'_2), \mathcal{V}(\Gamma'), \mathcal{L}^{n-1}_{\Gamma'_1}, \mathcal{L}^{n-1}_{\Gamma'_2}, \mathcal{L}^{n-1}_{\Gamma'}$
Require: The list of cap half-spaces in $\Gamma'_1$ and $\Gamma'_2$
Ensure: The list of cap half-spaces in $\Gamma'$
for each $F_{\Gamma'} \in \mathcal{L}^{n-1}_{\Gamma'}$ do
  // Get the vertices of $F_{\Gamma'_1}$ and $F_{\Gamma'_2}$ such that $F_{\Gamma'} = F_{\Gamma'_1} + F_{\Gamma'_2}$:
  $\mathcal{V}(F_{\Gamma'_1}) = \{\boldsymbol{a} \in \mathcal{V}(\Gamma'_1) / \exists \boldsymbol{b} \in \mathcal{V}(\Gamma'_2) \Rightarrow \boldsymbol{a} + \boldsymbol{b} \in \mathcal{V}(F_{\Gamma'})\}$
  $\mathcal{V}(F_{\Gamma'_2}) = \{\boldsymbol{b} \in \mathcal{V}(\Gamma'_2) / \exists \boldsymbol{a} \in \mathcal{V}(\Gamma'_1) \Rightarrow \boldsymbol{b} + \boldsymbol{a} \in \mathcal{V}(F_{\Gamma'})\}$
  for each $\boldsymbol{a} \in \mathcal{V}(F_{\Gamma'_1})$ do
    // Collect half-space numbers:
    Get $\mathcal{H}_{\boldsymbol{a}} = \{u, \boldsymbol{a} \in \bar{H}_u\}$
  end for
  // Get the combinatorial representation of $F_{\Gamma'_1}$:
  Compute $\mathcal{CR}(F_{\Gamma'_1}) = \{\cap \mathcal{H}_{\boldsymbol{a}}, \forall \boldsymbol{a} \in \mathcal{V}(F_{\Gamma'_1})\}$
  if No cap half-space in $\mathcal{CR}(F_{\Gamma'_1})$ then
    for each $\boldsymbol{b} \in \mathcal{V}(F_{\Gamma'_2})$ do
      // Collect half-space numbers:
      Get $\mathcal{H}_{\boldsymbol{b}} = \{v, \boldsymbol{b} \in \bar{H}_v\}$
    end for
    // Get the combinatorial representation of $F_{\Gamma'_2}$:
    Compute $\mathcal{CR}(F_{\Gamma'_2}) = \{\cap \mathcal{H}_{\boldsymbol{b}}, \forall \boldsymbol{b} \in \mathcal{V}(F_{\Gamma'_2})\}$
    if No cap half-space in $\mathcal{CR}(F_{\Gamma'_2})$ then
      $F_{\Gamma'}$ is not a cap half-space
    else
      $F_{\Gamma'}$ is a cap half-space
    end if
  else
    $F_{\Gamma'}$ is a cap half-space
  end if
end for
```

Then, we propose to simplify the calculated polytope after each sum. The idea is to generate a new associated polytope with a minimal set of cap half-spaces containing the topology of the polyhedra it represents. This can be done removing the cap facets of the calculated polytope and intersecting it with a big hypercube of the same dimension than the affine space of the polytope. By means of the Caps Propagation Theorem, it is ensured that the new obtained polytope keeps unaltered the topology of its associated polyhedron after the truncation. In other words, the data related to the tolerance analysis problem is not affected.

Let us go back to the 2D example in Figure 2.20 to illustrate the above. After the respective tracing process (trivial in this case) the cap facets of the calculated polytope are identified. The calculated polytope is not a minimal representation of the polyhedron it represents: three caps bound each unbounded direction while only one is enough. By removing the caps and truncating again the polyhedron with a square, a simpler associated polytope can be obtained. This polytope preserves the topology of the polyhedron. The comparison of the two associated polytopes (the one before and one after the truncation) is presented in Figure 3.5. From the tolerancing point of view, these polytopes represent the same displacement limits and therefore they can be called equivalent.

Definition 3.2.3 (Sum with caps removal) *We represent hereafter the Minkowski sum with caps removal as* $\Gamma'_1\tilde{+}\Gamma'_2$.

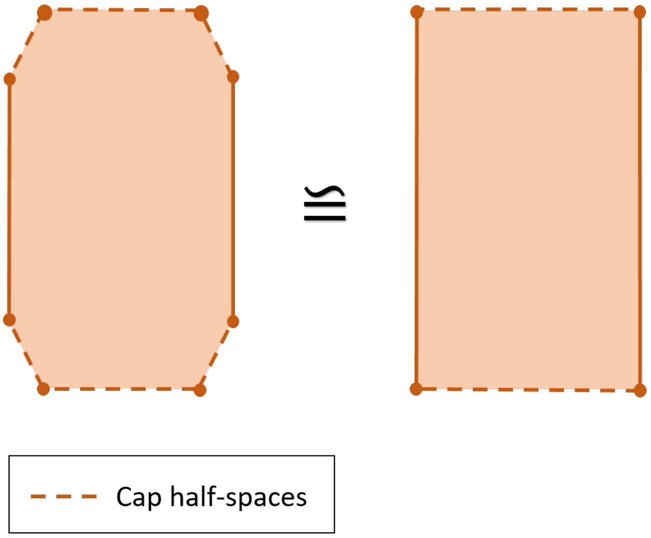


Figure 3.5: Two different associated polytopes for a polyhedron.

3.2.5 Operands commutativity in sums with caps removal

The sums with caps removal inherits from the Minkowski sum the commutativity property, i.e $\Gamma'_1\tilde{+}\Gamma'_2 = \Gamma'_2\tilde{+}\Gamma'_1$.

Proof. The Minkowski sum of two polyhedra is commutative. Then: $\Gamma'_1 \oplus \Gamma'_2 = \Gamma'_2 \oplus \Gamma'_1$. According to Theorems 3.2.1 and 3.2.2, it easy to see that the set of cap half-spaces from the sum $\Gamma'_1 \oplus \Gamma'_2$ are the same of those from $\Gamma'_2 \oplus \Gamma'_1$. We can reestablish the associated polyhedra satisfying that: $\Gamma_1 \oplus \Gamma_2 = \Gamma_2 \oplus \Gamma_1$. After truncating the polyhedra with an hypercube K, we still can verify that: $(\Gamma_1 \oplus \Gamma_2) \cap K = (\Gamma_2 \oplus \Gamma_1) \cap K$.

The former property applies also for more than two operands. This can be proved by recursion. In despite of this, operands commutativity can influence the computational time. This is the consequence of the propagation of the unbounded directions of the operands along the sum.

In order to explain the above, let us consider the operands P_1, P_2 and Γ'_3 in Figure 3.6. The sum in the order $P_1\tilde{+}P_2\tilde{+}\Gamma'_3$ (Figure 3.6a) is more expensive than the one in the order $P_1\tilde{+}\Gamma'_3\tilde{+}P_2$ (Figure 3.6b); even if both provide the same result. Notice that in the first case, $P_1\tilde{+}P_2$ produces a polytope without unbounded directions, and then, all its facets are kept. However, in the next sum $P_{12}\tilde{+}\Gamma'_3$ most of these facets disappear because of the unbounded direction introduced by Γ'_3. The second case introduces from the beginning the unbounded direction of Γ'_3, allowing to manipulate less complex polytopes in the next sum. The number of dual cones intersections required for the first sum order was 84 $(6 \times 6 + 12 \times 4)$ while for the second one 48 $(6 \times 4 + 4 \times 6)$. Based on this, it is clear that the sum $P_1\tilde{+}\Gamma'_3\tilde{+}P_2$ is less complex than $P_1\tilde{+}P_2\tilde{+}\Gamma'_3$.

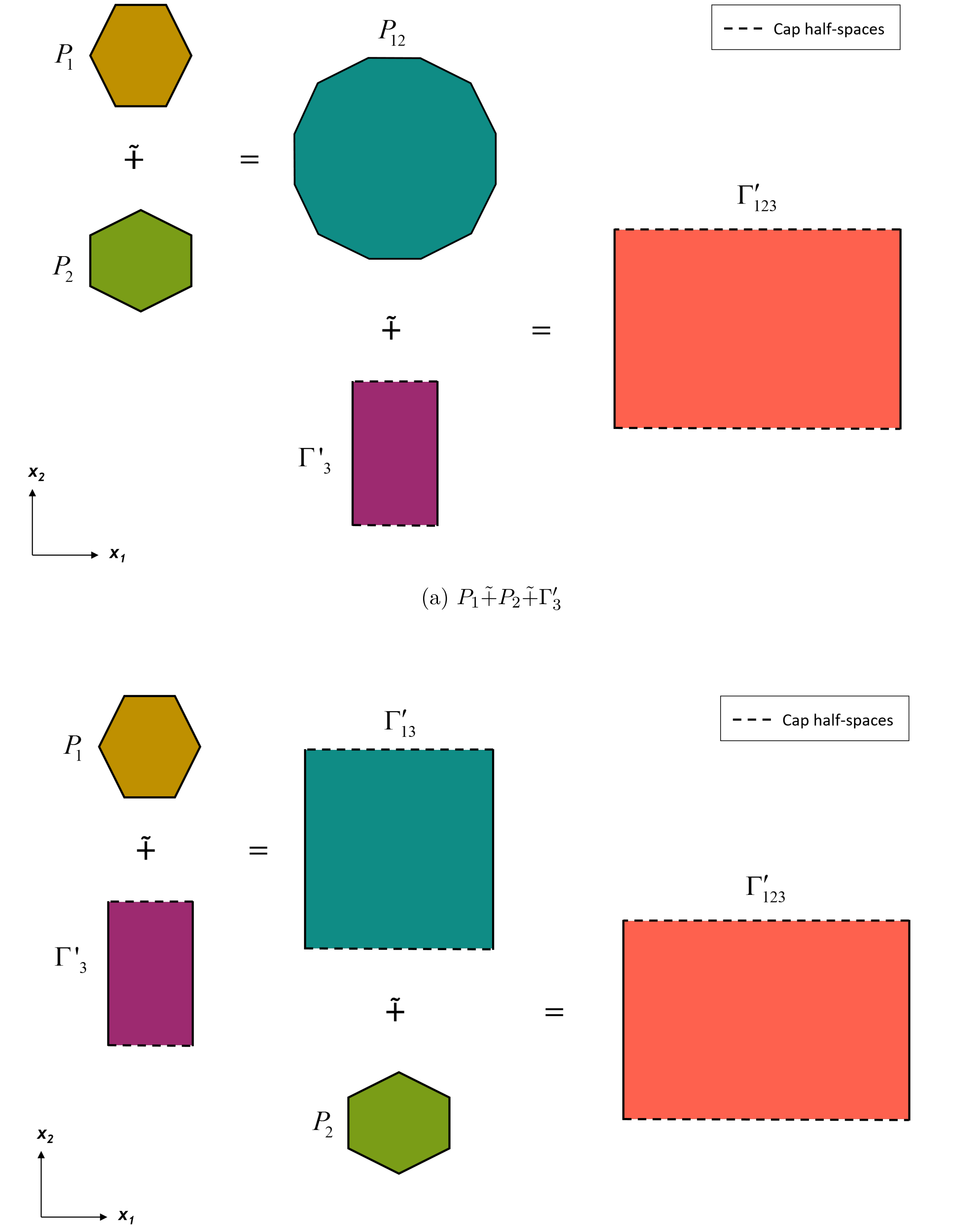
Summing first the operands producing the greatest set of linearly independent unbounded directions avoids calculating data that could be removed in a subsequent sum. These unbounded directions are actually the extreme rays of the associated polyhedron. We propose to establish a convenient summation order according to the number of linearly independent rays produced by each possible pair of operands.

The former can be done either from a geometric or from a kinematic perspective. In the first case, the characterization of the extreme rays defining each operand polyhedra is required. By checking each possible pair of operands association, the couple having the greatest set of linearly independent rays can be determined. In geometric tolerancing, this is equivalent to perform a kinematic analysis of the considered assembly, for identifying in a serial chain the couple of joints producing the largest number of DOFs.

3.2.6 Algorithm: sum with caps removal

Algorithm 3 formalizes a strategy for summing several polyhedra by means of their associated polytopes bounded with cap half-spaces. We represent the summation with caps removal as $\tilde{\sum}_{i=1}^{n_p}\Gamma'_i$.

The idea is, first, to establish a convenient summation order according to the propagation of the extreme rays of the polyhedra; and then, to compute the sum tracing the cap facets and truncate each partial result.



(a) $P_1 \tilde{+} P_2 \tilde{+} \Gamma'_3$

(b) $P_1 \tilde{+} \Gamma'_3 \tilde{+} P_2$

Figure 3.6: Operands commutativity effects in sums with caps removal.

Algorithm 3 Sum with caps removal

```
Require: Set of $n_p$ polytopes $\Gamma'_i$ in $\mathbb{R}^n$
Require: The list of half-spaces $\mathcal{H}_i^{cap}$ in $\Gamma'_i$
Ensure: The sum with caps removal $\Gamma'_R = \tilde{\sum}_{i=1}^{n_p}\Gamma'_i$
Create an hypercube $K$ in $\mathbb{R}^n$
Set a convenient summation order $\{\Gamma'_1, ..., \Gamma'_{n_p}\}$
$\Gamma'_R = \Gamma'_1$
for $i = 2 : n_p$ do
    Compute $\Gamma'_c = \Gamma'_R \oplus \Gamma'_i$
    Compute $\mathcal{H}_c^{cap}$ from $\mathcal{H}_r^{cap}$ and $\mathcal{H}_i^{cap}$
    Remove $\mathcal{H}_c^{cap}$ from $\Gamma'_c$ to obtain $\Gamma_c$
    Compute $\Gamma'_R = \Gamma_c \cap K$
end for
Return $\Gamma'_R$
```

3.3 Tracing caps: intersections

In tolerance simulations, not only sums but also polytopes intersections are required. In the general case, the result of an intersection has to be used again for computing a sum. As the set of cap half-spaces of each operand must be identified to run sums with caps removal, caps must also be traced during intersections. This allows us to warranty the trace of the half-spaces attributes throughout the entire simulation.

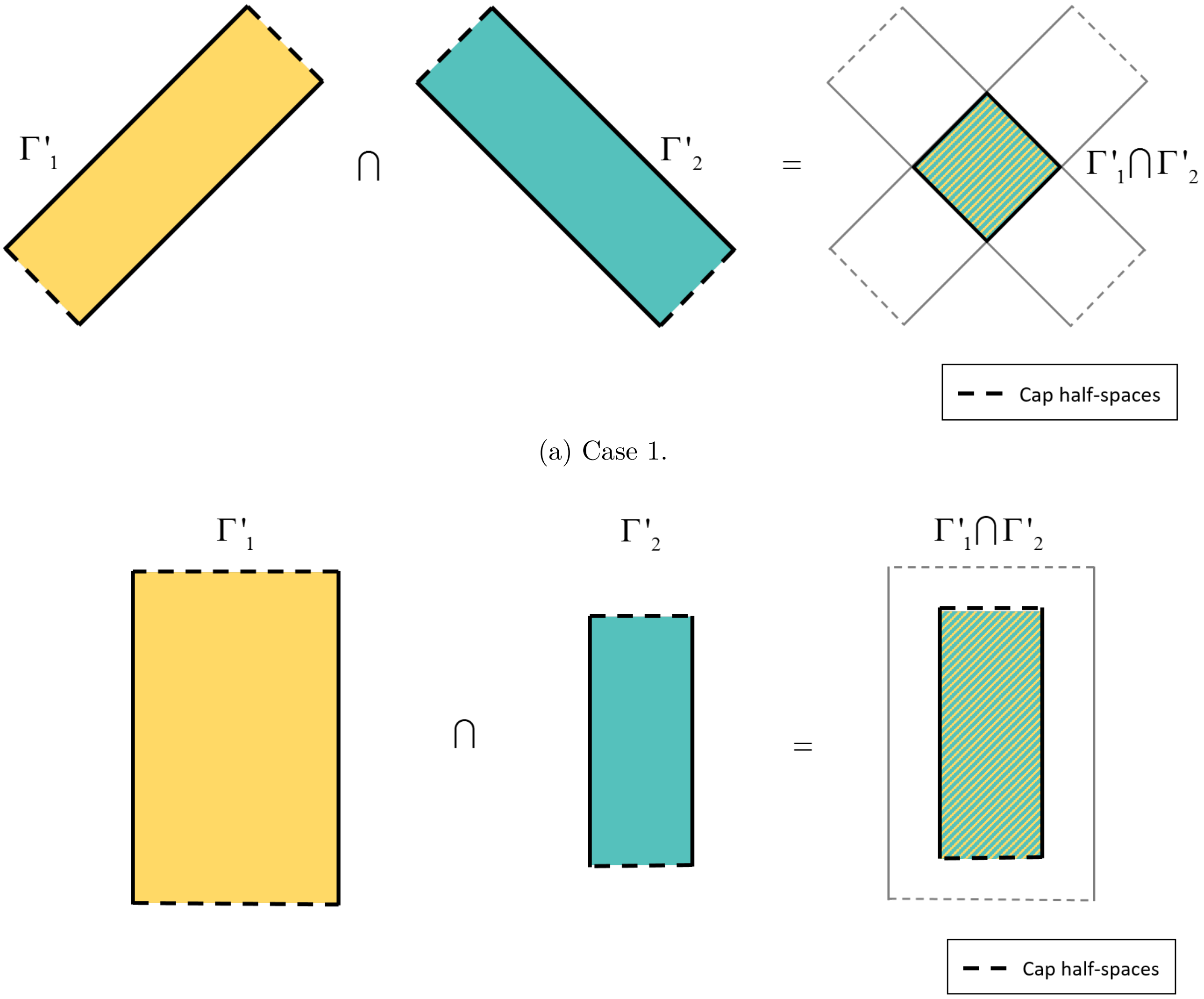
During the intersection of two capped polytopes two scenarios can occur with each cap half-space:

- it is removed because of redundancy. Figure 3.7a illustrates this case.
- it remains unaltered and keeps its status of cap. Figure 3.7b illustrates this case.

From an algorithmic point of view, this is not a complicated task. It simply involves implementing a label management in a double description algorithm.

3.4 Tracing caps: inclusion test

The final objective of a tolerance analysis is to determine whether a functional condition is satisfied. In the method based on polytopes, this can be done by checking the inclusion of the polytope representing the whole stack-up of deviations in the functional polytope.



(a) Case 1.

(b) Case 2.

Figure 3.7: Capped polytopes intersection.

When tolerance simulations are made with capped polytopes, special attention has to be paid to the inclusion tests. We want to avoid that the inclusion of a calculated polytope in a functional one depends on cap half-spaces. This situation could lead to the misleading conclusion that the design requirements are satisfied. Figure 3.8 illustrates this case: $\Gamma'_R \subset P_F$. However, as this inclusion depends on some caps, it should be concluded that the functional requirements represented by P_F cannot be guaranteed.

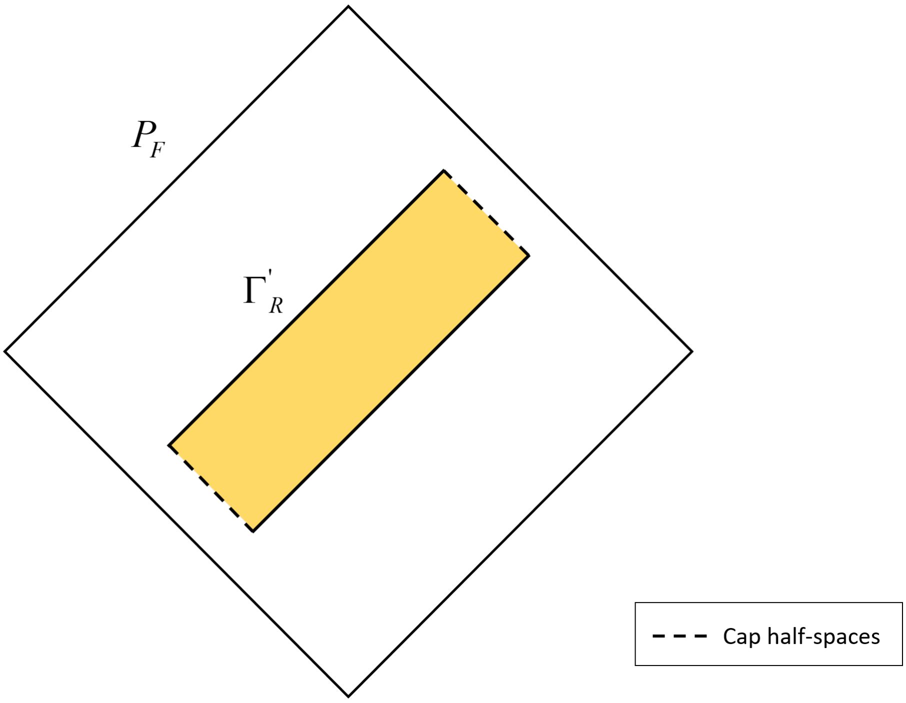


Figure 3.8: Checking the inclusion of a capped polytope in a functional polytope.

This situation can be faced by testing the inclusion not with the capped polytope but with its associated polyhedron: $\Gamma_R \subseteq ?\ P_F$. It is equivalent to check if all the constraints of the functional polytopes are satisfied by the calculated polyhedron. In this case, this cannot be performed in the way proposed in Eq. (2.8), because in the general case Γ_R has not vertices.

We propose to do so truncating P_F with the half-spaces of Γ_R. At the end of the process, the remaining half-spaces must only come from Γ_R or from cap half-spaces in P_F. If at least one non-cap half-space of P_F is found, it means that the functional conditions cannot be satisfied.

3.5 Case study: solution by caps removal method

With the aim of comparing, we solved the same case study presented in Section 2.6 following the method presented throughout this chapter chapter.

The same set of operands (Table 2.1) was used for the simulation, but in this case their cap half-spaces were identified and labeled. This was done according to the surface class from which each operand derived. The tracing process was done along the different operations.

3.5.1 Caps removal

The ‘cleaning’ process was done to keep the set of cap half-spaces at the minimum: two caps appear to bound each DOF (see Table 3.1). This is illustrated with the computation of the operands $\Gamma'_{1,4/3,0}$ and $\Gamma'_{1,0/3,0-c}$ (Figures 3.9a and 3.9b), by means of 3D representations. Although the first sum took 20% more time to compute (with respect to the caps-based method), the complexity of the polytope we obtained was notably reduced: 192 versus 1984 vertices. The increase in calculation time is due to the truncation process launched after the regular summation to reestablish a minimum set of cap half-spaces (see Figure 3.9a).

Table 3.1: Summary of the simulation with caps control (HS: half-space).

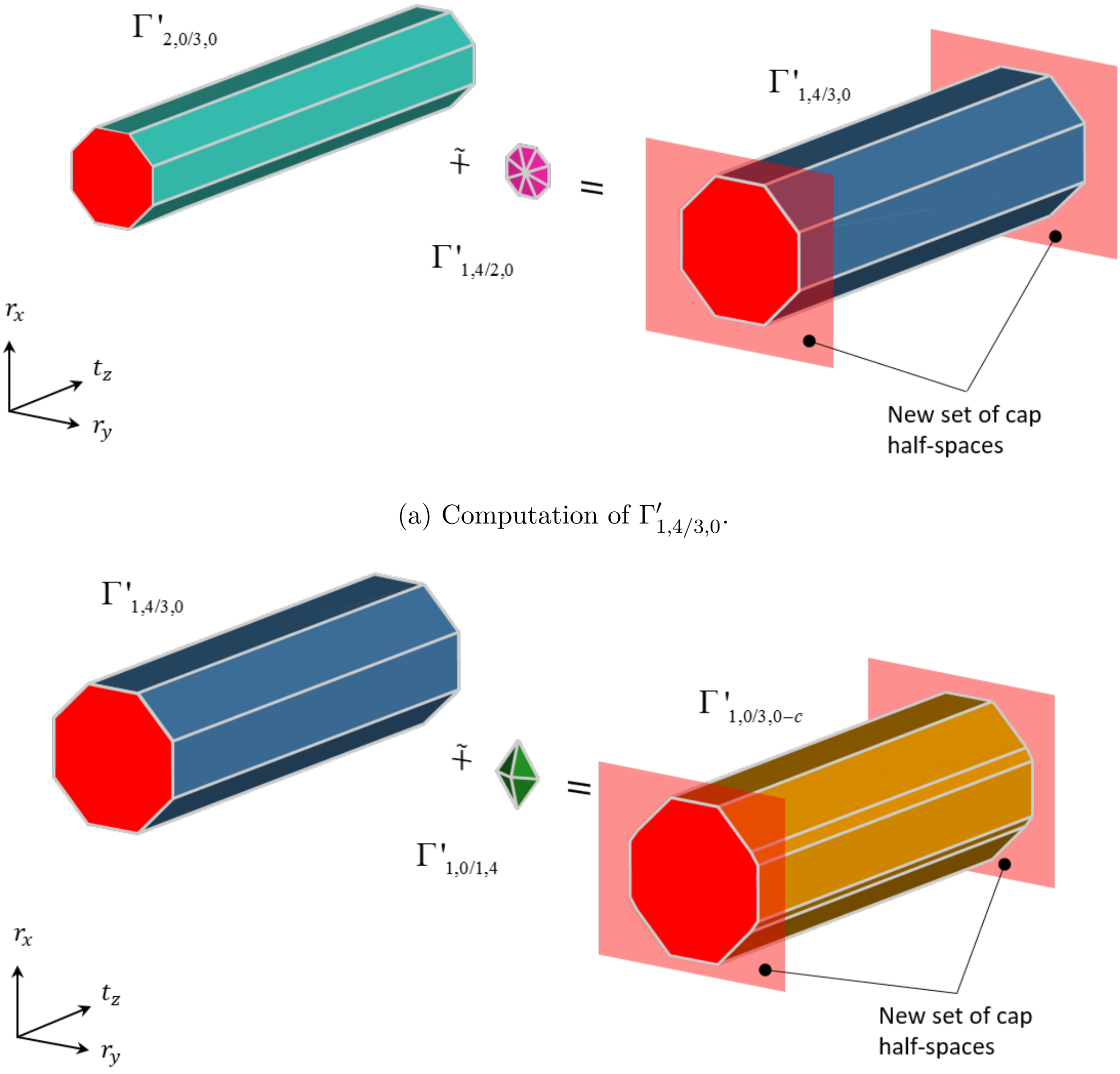
Operation			DOFs	Cap HS	Non-cap HS	Vertices	Time [s]
$\Gamma'_{1,0/3,0-ab}$	$= \Gamma'_{1,0/3,0-a}$	$\cap\, \Gamma'_{1,0/3,0-b}$	3	6	14	128	0,02
$\Gamma'_{1,4/3,0}$	$= \Gamma'_{2,0/3,0}$	$\tilde{+}\, \Gamma'_{1,4/2,0}$	4	8	8	128	3,16
$\Gamma'_{1,0/3,0-c}$	$= \Gamma'_{1,4/3,0}$	$\tilde{+}\, \Gamma'_{1,0/1,4}$	4	8	12	192	0,36
$\Gamma'_{1,0/3,0}$	$= \Gamma'_{1,0/3,0-ab}$	$\cap\, \Gamma'_{1,0/3,0-c}$	1	2	26	384	0,05
$\Gamma'_{3,0/4,0-ab}$	$= \Gamma'_{3,0/4,0-a}$	$\cap\, \Gamma'_{3,0/4,0-b}$	1	2	28	288	0,05
$\Gamma'_{3,0/4,0}$	$= \Gamma'_{3,0/4,0-ab}$	$\cap\, \Gamma'_{3,0/4,0-c}$	0	0	36	504	0,11
$\Gamma'_{4,0/6,0-ab}$	$= \Gamma'_{4,0/6,0-a}$	$\cap\, \Gamma'_{4,0/6,0-b}$	3	6	14	128	0,03
$\Gamma'_{4,0/6,4}$	$= \Gamma'_{4,0/5,0}$	$\tilde{+}\, \Gamma'_{5,0/6,4}$	4	8	8	128	3,44
$\Gamma'_{4,0/6,0-c}$	$= \Gamma'_{4,0/6,4}$	$\tilde{+}\, \Gamma'_{6,4/6,0}$	4	8	12	192	0,38
$\Gamma'_{4,0/6,0}$	$= \Gamma'_{4,0/6,0-ab}$	$\cap\, \Gamma'_{4,0/6,0-c}$	1	2	26	384	0,05
$\Gamma'_{1,1/3,0}$	$= \Gamma'_{1,1/6,1}$	$\tilde{+}\, \Gamma'_{1,0/3,0}$	4	8	12	192	8,72
$\Gamma'_{1,1/4,0}$	$= \Gamma'_{1,1/3,0}$	$\tilde{+}\, \Gamma'_{3,0/4,0}$	4	8	28	448	7,55
$\Gamma''_{R} = \Gamma'_{1,1/6,0}$	$= \Gamma'_{1,1/4,0}$	$\tilde{+}\, \Gamma'_{4,0/6,0}$	4	8	28	448	45,45

Computations performed with the library politopix with an Intel Core i7-3740QM.

The real impact of this simplification can be noticed in the next sum (see Figure 3.9b). When using again the simplified version of $\Gamma'_{1,4/3,0}$ to calculate $\Gamma'_{1,0/3,0-c}$, the required time was only 0,04% of the one needed without the simplification.

Similar phenomena appear all along the simulation, making it lasts 01:09 s. This represents a reduction of 99.37% of the time of the simulation presented in Section 2.6. It is worth mentioning that no information was lost by applying the proposed method. The gain on time is due to the calculation of meaningless information that was avoided. Thus, when comparing the two methods, we can conclude that the last one is more efficient. This because the systematic filtering of the data.

Figure 3.10 presents the comparison of the results of the simulation without and with caps removal. We represent the final polytope calculated controlling the caps spreading as Γ''_{R}. This was carried out projecting both results to the subspace of the bounded displacements (the space spanned by $[r_x, r_y]$ in this case). By doing this, the influence of the caps facets is completely avoided in the comparison test. This comparison was done numerically, testing that



(a) Computation of $\Gamma'_{1,4/3,0}$.

(b) Computation of $\Gamma'_{1,0/3,0-c}$.

Figure 3.9: 3D representation of the computation of $\Gamma'_{1,0/3,0-c}$ with caps control. Cap facets are shown in red.

the vertices of one of projected polytopes satisfied the constraints of the other, and vice versa. The $\mathcal{HV}$-description of the final calculated polytope is presented in Annex B.2.

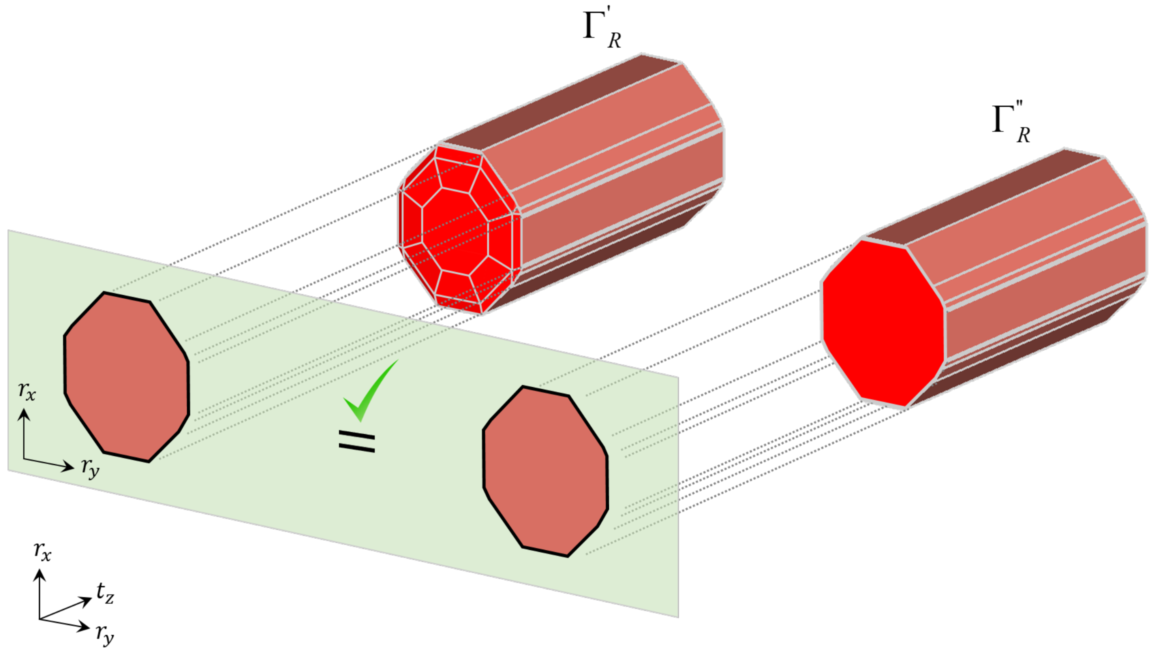


Figure 3.10: Comparison of the results of the simulation without (at the left-hand side) and with caps removal (at the right-hand side).

3.5.2 Influence of summation order

When several sums had to be computed consecutively, the operands were ordered comparing the DOFs generated by each possible pair. The couple of operands producing the greatest number of DOF were summed first.

The effects of the summation order can be explained with the computation of $\Gamma'_{1,0/3,0-c}$. When computing $\Gamma'_{1,0/1,4}\tilde{+}\Gamma'_{1,4/2,0}$, 3 DOF are generated, but one of them is after relaxed by the operand $\Gamma'_{2,0/3,0}$. In this case, several facets are kept after the first sum but are later removed during the second one. This summation order implies calculating 65 280 intersections of dual cones in 6D. According to this, a more efficient computation order would be: $\Gamma'_{1,0/3,0-c} = \Gamma'_{2,0/3,0}\tilde{+}\Gamma'_{1,4/2,0}\tilde{+}\Gamma'_{1,0/1,4}$. By doing this, 4 DOF are generated from the first sum. In contrast, this summation order implies intersecting 27 648 pairs of dual cones in 6D. This justifies the difference in the computational time: 11,63s vs 3,52s. It is worth to mention that both cases produced the same result.

Exactly the same phenomenon occurs with the final three sums of the simulation. $\Gamma'_{1,0/3,0}$ and $\Gamma'_{3,0/4,0}$ have 1 and 0 unbounded directions respectively, while $\Gamma'_{1,1/6,0}$ has 3. Then, if we sum first $\Gamma'_{1,0/3,0}\tilde{+}\Gamma'_{3,0/4,0}$ a polytope with 1 unbounded direction is obtained and no significant simplification can be made. The next sum with $\Gamma'_{1,1/6,0}$ to obtain Γ'_R took in this case more than 3hrs. This in comparison with the 45s spent when following the order proposed in Table 3.1.

3.6 Conclusions

For computational and algorithmic reasons, cap half-spaces are required to bound the polyhedra into polytopes. In order to distinguish the cap half-spaces from those representing real limits in the tolerancing problem, traceability is the key concept. Based on the combinatorial properties of polytopes, we characterized how caps spread throughout all the operations: sums, intersections and inclusion tests.

We found that tracing the cap half-spaces can reduce the complexity of the calculated sets during tolerance simulations. Among the required operations, it is during sums that caps spread. Therefore, some rules were proposed to identify and simplify cap sets.

Reducing the complexity of a polytope derived from the sum of two capped polytopes becomes useful when subsequent sums have to be calculated. As shown in Section 2.6, when a polytope highly contaminated with cap facets is used for a new sum, the complexity of the new calculated polytope is far greater.

It is worth mentioning that the proposed simplification does not imply any loss of information since cap half-spaces bound theoretically unbounded displacements of a toleranced feature. Therefore, the limits established by these half-spaces are of no interest from the tolerancing point of view.

The advantages of the strategy based on cap trace and truncation were demonstrated by means of an industrial application. The computational time could be significantly reduced compared to that required by the strategy presented in the previous chapter. In addition to the reduction in computational time, the proposed method reduces the complexity of the operands, and hence the probability of having numerical problems during the calculations.

By means of the case study, we discussed the effects of the operand order when computing more than two sums with cap removal. Even though it was demonstrated that this operation is commutative, the summation order affects its algorithmic complexity. We therefore suggest first summing the operands producing the greatest number of unbounded directions.

The strategy of labelling and tracing half-spaces is an important step towards tolerance synthesis. In this way, the topological elements of a calculated polytope can be linked to our input data, the initial set of ISO geometrical specifications. With this information, the most influential features in the tolerance chain can be determined.

Even if the strategy presented in this chapter is able to reduce the complexity of the manipulated models and consequently the computational time, the ideal scenario is when no cap half-spaces are used at all in the computations, which is actually the objective of the method proposed in the next chapter.

Chapter 4

Kinematic decomposition of geometric constraints

In this chapter, we propose to use the theory of screws to model the mobility conditions of an assembly during tolerance simulation. This enables us to simplify sets of geometric constraints, decomposing them into the sum of a polytope (a bounded set) and a sum of straight lines (an unbounded set). The unbounded part of the polyhedra is characterised by the degrees of freedom of the toleranced feature or the joint. Therefore, this decomposition can be performed by kinematic analysis. Due to the similarities between their mathematical definition and the geometric constraints we manipulate in tolerance analysis, we propose to do this by means of screws. The idea behind this strategy is, instead of summing polyhedra in $\mathbb{R}^6$*, to sum only their underlying polytopes by isolating the unbounded part of the operands. This implies a reduction in operand complexity and consequently a reduction in the computational time. Other advantages of using prismatic polyhedra for modelling geometric constraints are discussed. The proposed strategy is illustrated at the end of the chapter using the case study already presented.*

4.1 Kinematic and tolerance analysis

Due to the similarities in their mathematical definition with the geometric constraints, the theory of displacement subgroups and the theory of screws are suitable for modelling rigid body kinematics during tolerance analysis. Although both theories are finally based on the same mathematical concepts (theory of orthogonal spaces), they have been used independently by the scientific community, as we explain below.

4.1.1 Theory of displacement subgroups

A rigid transformation (also called isometry) is a geometric transformation acting over a set of points such that the lengths of the vectors as well as the angular orientations are maintained. In the Euclidean affine space of dimension 3, the set of rigid transformations D has an algebraic structure of a continuous group; more specifically, it is a Lie group of dimension 6 (Hervé, 1982). There exists in this group several subsets which are stable under the product operations (i.e. the product of two elements of this subset belongs to the subset). This subsets are called subgroups. The list of the subgroups of D is presented in Table 4.1.

Table 4.1: Subgroups of D and their properties (Clément et al., 1998).

Symbolic name	Description	DOF	Constraints free variables
D	Rigid displacement	6	3 transl., 3 rot.
RP_3	Free revolute pair translating in space	4	3 transl., 1 rot.
P_3	Spatial translation	3	3 transl.
S_3	Spherical rotation	3	3 rot.
F_3	Planar motion	3	2 transl., 1 rot.
C	Cylindrical motion	2	1 transl., 1 rot.
P_2	Planar translation	2	2 transl.
P	Translation around an axis	1	1 transl.
R	Rotation around an axis	1	1 rot.
H	Helical motion	1	Coupled transl. and rot.
E	Identity motion	0	-

The combination of these subgroups has been used in robotics to analyze the mobility conditions of kinematic chains. Fanghella (1988) presented an exhaustive list of all possible combinations considering different geometric relations: coincidence, parallelism, intersection, etc. (see Annex A.1). Hervé (1994) presented a more general procedure using the exponential maps of the Lie groups to analytically generate the Lie sub-algebras, which are equivalent to the aforementioned subgroups.

In geometric tolerancing we have special interest in the subgroups that leave a surface invariant. This surface is actually the one subject to metrological control. A displacement D_i leaves a surface S invariant in position if each point $\boldsymbol{M} \in S$ still belongs to S after being transformed by the displacement: $\boldsymbol{M'} \in S$ with $\boldsymbol{M'} = D_i\boldsymbol{M}$ (Clément et al., 1998). Among the whole list, seven subgroups are compliant with this property, which define seven surface classes: spherical, planar, cylindrical, helical, of revolution, prismatic and complex.

Although the product of two successive displacements is also a displacement (Hervé, 1999), the displacement subgroups cannot explain all rigid motions in space. In some cases, a rigid motion is a displacement subset but not a subgroup itself (Li et al., 2004).

4.1.2 Theory of screws

Theory of screws, developed around Chasles' and Poinsot's theorem (Ball, 1900), aims to reduce the displacement of a rigid body to its simplest form (see Figure 4.1).

- Chasles' theorem: any rigid body motion can be represented instantaneously as a rotation about a unique line and a translation along that same line.
- Poinsot's theorem: any system of moments and forces acting on a rigid body can be represented instantaneously as a one moment and one force.

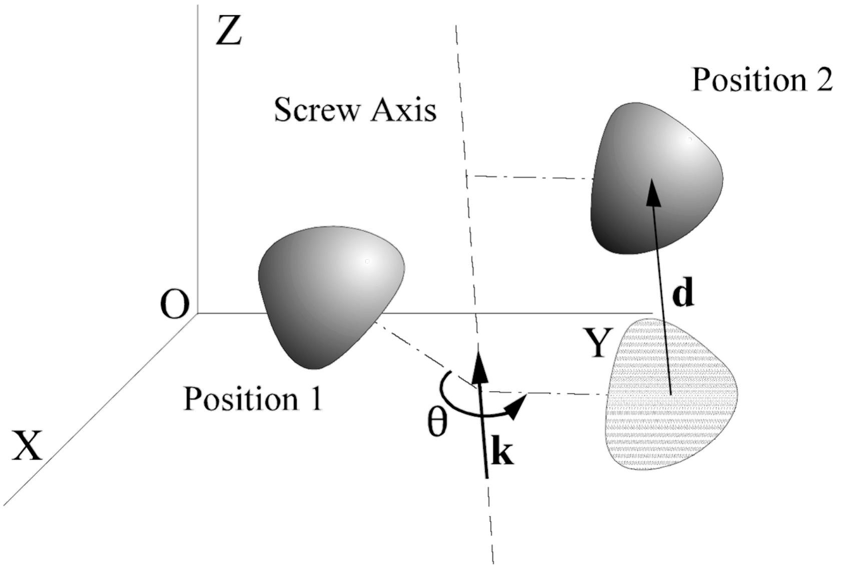


Figure 4.1: Screw rotation and translation applied to a 3D rigid body (Smith, 2001).

From these theorems, the concepts of twist and wrench derive. Twists can be analyzed as allowable motions while wrenches as forbidden motions (Su and Yue, 2013). Both, twists $\hat{\boldsymbol{T}}$ and wrenches $\hat{\boldsymbol{W}}$ are 1x6 row vectors written as (Adams, 1998):

$$\hat{\boldsymbol{T}} = [\boldsymbol{\omega} \mid \boldsymbol{v}] = [\boldsymbol{\omega} \mid \boldsymbol{r} \times \boldsymbol{\omega}] \tag{4.1}$$

$$\hat{\boldsymbol{W}} = [\boldsymbol{f} \mid \boldsymbol{m}] = [\boldsymbol{f} \mid \boldsymbol{r} \times \boldsymbol{f}] \tag{4.2}$$

where $\boldsymbol{\omega}$ is a unit angular velocity vector, $\boldsymbol{v}$ is a unit linear velocity vector, $\boldsymbol{f}$ is a unit force vector, $\boldsymbol{m}$ is unit moment vector about the point $\boldsymbol{r}$.

By computing systematically the union and intersection of screws, the mobility condition of any two parts of a mechanical system can be determined (Adams and Whitney, 1999; Gerbino and Arrichiello, 2004; Rico et al., 1999). Open and closed chains can be analyzed even if the former present redundantly suppressed degrees of freedom.

Let us take the example of a planar pair to illustrate these concepts. Assuming that the normal of the plane corresponds to the $\boldsymbol{z}$-axis, we can define each mobility at a point $\boldsymbol{M}$ by means of the following twists:

$$\begin{aligned}\hat{\boldsymbol{T}}_{\boldsymbol{r_z}} &= [\boldsymbol{z} \mid \boldsymbol{OM} \times \boldsymbol{z}]\\ \hat{\boldsymbol{T}}_{\boldsymbol{t_x}} &= [\boldsymbol{0} \mid \boldsymbol{x}]\\ \hat{\boldsymbol{T}}_{\boldsymbol{t_y}} &= [\boldsymbol{0} \mid \boldsymbol{y}]\end{aligned}$$

where $\boldsymbol{OM}$ is the position vector of the point $\boldsymbol{M}$ from the origin of the reference system $\boldsymbol{O}$.

Twist-space and Wrench-space

In screw terms, the mobility of each kinematic pair of a mechanical system characterizes a subspace of $\mathbb{R}^6$: the twist-space. This subspace is spanned by the set of d twists describing each of the unbounded movements of the joint (Konkar and Cutkosky, 1995):

$$\mathcal{T} = \begin{bmatrix} \hat{\boldsymbol{T}}_{\boldsymbol{1}} \\ \hat{\boldsymbol{T}}_{\boldsymbol{2}} \\ \cdots \\ \hat{\boldsymbol{T}}_{\boldsymbol{d}} \end{bmatrix} \tag{4.3}$$

$\mathcal{T}$ is usually called twist-matrix. Its dual vector space corresponds to the wrench-space (under the considerations declared in the next section). It is represented by a wrench-matrix and corresponds to the subspace of forbidden motions.

Going back to the example of the planar pair, we can define the derived twist-matrix $\mathcal{T}_p$ corresponding to the three-dimensional twist-space:

$$\mathcal{T}_p = \begin{bmatrix} \hat{\boldsymbol{T}}_{\boldsymbol{r_z}} \\ \hat{\boldsymbol{T}}_{\boldsymbol{t_{x/M}}} \\ \hat{\boldsymbol{T}}_{\boldsymbol{t_{y/M}}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{z} \mid \boldsymbol{OM} \times \boldsymbol{z} \\ \boldsymbol{0} \mid \boldsymbol{x} \\ \boldsymbol{0} \mid \boldsymbol{y} \end{bmatrix}$$

Notice, that the mobility conditions depend on the point $\boldsymbol{M}$ in which the displacements are expressed. The unbounded rotation r_z can generate additional translations depending on the vector $\boldsymbol{OM}$. Equivalently, we can also define the wrench-matrix:

$$\mathcal{W}_p = \begin{bmatrix} \hat{\boldsymbol{W}}_{\boldsymbol{r_x}} \\ \hat{\boldsymbol{W}}_{\boldsymbol{r_y}} \\ \hat{\boldsymbol{W}}_{\boldsymbol{t_{z/M}}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x} \mid \boldsymbol{OM} \times \boldsymbol{x} \\ \boldsymbol{y} \mid \boldsymbol{OM} \times \boldsymbol{y} \\ \boldsymbol{0} \mid \boldsymbol{z} \end{bmatrix}$$

As a toleranced feature has no mobility, it is possible to see it as a joint (usually called internal joint) mating the nominal feature and the substitute one. The mobility, in this case, is characterized by the set of displacements leaving the toleranced feature invariant. In other words, the degree of invariance of an internal joint (i.e. a toleranced feature) can be modelled the same way we model the degree of freedom of a kinematic joint. Tolerance chains generally involve internal and kinematic joints alternatively.

We can then model identically, by means of screws, internal and kinematic joints for tolerance analysis purposes. By doing this, it is possible to differentiate, before each operation with sets of constraints, the motion directions which can be controlled (the wrench space) and those that cannot (the twist space).

Reciprocity

Under the consideration of zero virtual work between wrenches on twists, the twist-space and the wrench-space become orthogonal subspaces in $\mathbb{R}^6$: $\mathcal{T}^{\perp} = \mathcal{W}$, and therefore: $\dim(\mathcal{T})+\dim(\mathcal{W}) = 6$

This can be said considering that the constrained parts are rigid, the contact surfaces are frictionless and the contact between parts does not break (Gerbino and Arrichiello, 2004); which is compliant with our assumptions.

This property, called reciprocity of screws, is one of the most important properties of this theory since changing from one subspace to the other is made easy. This can be done calculating the nullspace.

Union of screws

We reviewed how to model the mobility conditions of internal and kinematic joints using screws. Now, let us describe how to model the mobility conditions of a kinematic chain.

When a set of n kinematic pairs are in a serial configuration, their mobility propagate. In terms of screws, it can be modelled by means of the union of the respective twist-matrices (Konkar and Cutkosky, 1995):

$$\mathcal{T}_{union} = \bigcup_{i=1}^{n} \mathcal{T}_i = \begin{bmatrix} \mathcal{T}_1 \\ \mathcal{T}_2 \\ \dots \\ \mathcal{T}_n \end{bmatrix} \tag{4.4}$$

Analogously, when a set of n kinematic pairs are in a parallel configuration, their mobility can be modelled by computing the intersection of the derived twist-matrices, or more simply, as the union of the corresponding wrenches (Konkar and Cutkosky, 1995; Adams, 1998):

$$\mathcal{T}_{inter} = \bigcap_{i=1}^{n} \mathcal{T}_i = \left[\bigcup_{i=1}^{n} \mathcal{W}_i \right]^{\perp} = \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{W}_2 \\ \dots \\ \mathcal{W}_n \end{bmatrix}^{\perp} \tag{4.5}$$

When joining screws, the resulting matrices are commonly of non-full rank. This occurs because several joints can introduce (in the case of twists) or restrict (in the case of wrenches) the same mobility. Therefore, it is usual to compute the reduced row echelon form of the matrices during kinematic analysis.

In short, by means of unions and intersections of screws, the mobility conditions of any couple of surfaces in a tolerance chain can be characterized.

4.2 Prismatic polyhedra

Let us describe how the theory of screws can be used to decompose geometric constraints.

4.2.1 Definition and properties

According to the Minkowski-Weyl theorem (Ziegler, 1995), a polyhedron $\Gamma \subset \mathbb{R}^n$ can be decomposed into the sum of a bounded set and an unbounded set, namely a polytope P and a polyhedral cone C (see Figure 4.2):

$$\Gamma = P \oplus C \tag{4.6}$$

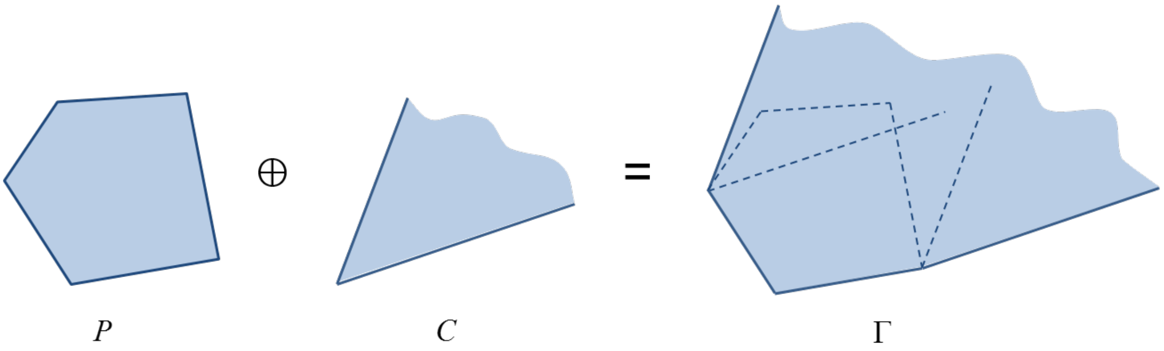


Figure 4.2: Decomposition of polyhedra (Ziegler, 1995).

Definition 4.2.1 (Prismatic polyhedron) *The Minkowski sum of a polytope with a finite set of linearly independent straight lines Δ_j generates a prismatic polyhedron of $\mathbb{R}^n$:*

$$\Gamma = P \oplus \sum_{j=1}^{d} \Delta_j \textit{ with } d \leq n \tag{4.7}$$

Even if an infinite number of polytopes can generate a prismatic polyhedron (Eq. 4.7), we have special interest in manipulating the one contained in the subspace orthogonal to the straight lines:

$$P \subset H \text{ with } H = \bigcap_{i=1}^{k} H_i \text{ and } \Delta_i \perp H_i, \forall i$$

We justify our interest for this polytope in two reasons. From a numerical point of view, this polytope is more stable. This in contrast, for example, with a polytope included in a subspace near to be parallel to one of the straight lines. Additionally, from the tolerancing point of view, the $\mathcal{V}$-description of this polytope is independent of the unbounded displacements of the related feature or joint.

An example illustrating a prismatic polyhedron and its decomposition is presented in Figure 4.3. Since Γ is made up only by one straight line, H is defined only by one plane.

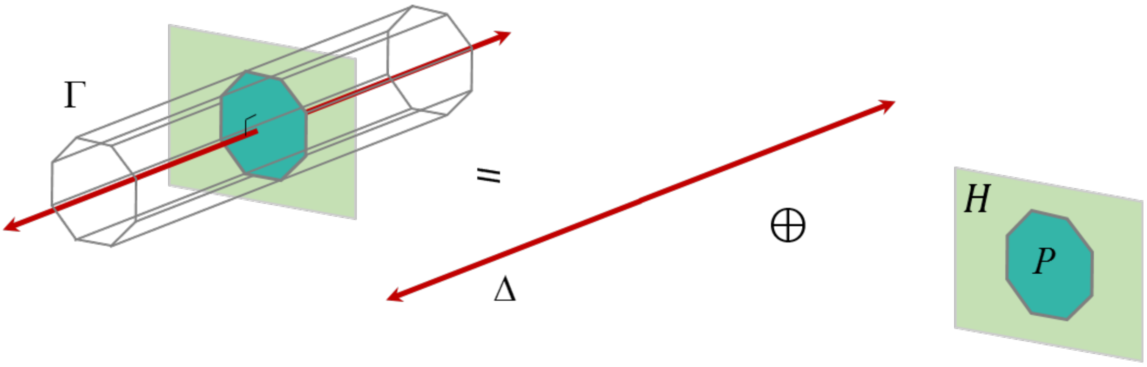


Figure 4.3: Decomposition of a polyhedron into the sum of a polytope and a straight line.

4.2.2 Polyhedra decomposition in geometric tolerancing

In the most of the cases, polyhedra derived from geometric tolerancing are prismatic. Each degree of freedom (or invariance) implies a sweeping operation of a polytope (derived from geometric or contact constraints) along a straight line, defining prismatic operands. Therefore, their decomposition can be done considering the mobility conditions of the feature or joint.

Due to the similarities between their mathematical definition and the geometric constraints we manipulate in tolerance analysis, we propose to do this by means of screws. A screw is a vector in $\mathbb{R}^6$ (Huang et al., 2008). For a given feature or joint, a twist can be seen as the director vector of a straight line Δ_j in $\mathbb{R}^6$ characterizing a non-restricted displacement. By duality, this vector represents a hyperplane H_{Δ_j} (see Figure 4.3). In turn, the intersection of the hyperplanes orthogonal to each of the twists of a feature characterizes the wrench-space. As the geometric constraints obtained in tolerancing are limits in the directions of bounded displacements, they define a polytope in the wrench-space.

In order to illustrate this, we applied such decomposition to the most common geometric features and kinematic joints used in tolerancing (Tables 4.2, 4.3 and 4.4). We distinguish three types of kinematic joints according to their contact feature type: surfaces, lines and point.

Following the classification proposed by ISO standards (ISO-3952-1, 1981), we consider: spherical, planar, cylindrical, revolute and prismatic pairs. These joint types can be treated the same way as their associated feature type. Seven invariance classes are considered (Desrochers and Clément, 1994): spherical, plane, cylindrical, of revolution, prismatic and complex classes (see Table 4.2). The helical pair and the helical class are not treated because they are not of practical interest in geometric tolerancing. In the case of linear contact features, we consider ball-and-cylinder and cylinder-and-plane pairs. In the case of punctual contact feature, we consider the ball-and-plane pair (see Table 4.4). The general procedure for identifying the unbounded displacements in a tolerance chain is summarized in Figure 4.4.

Once the straight lines of the polyhedron are identified, the underlying polytope can be obtained. This can be done by intersecting the original set of constraints (which is obtained by applying double inequality (2.3) to a set of points belonging to a toleranced feature) with a set of hyperplanes H_{Δ_j} such that $H_{\Delta_j} \perp \Delta_j \; \forall j \in \{1, ..., d\}$.

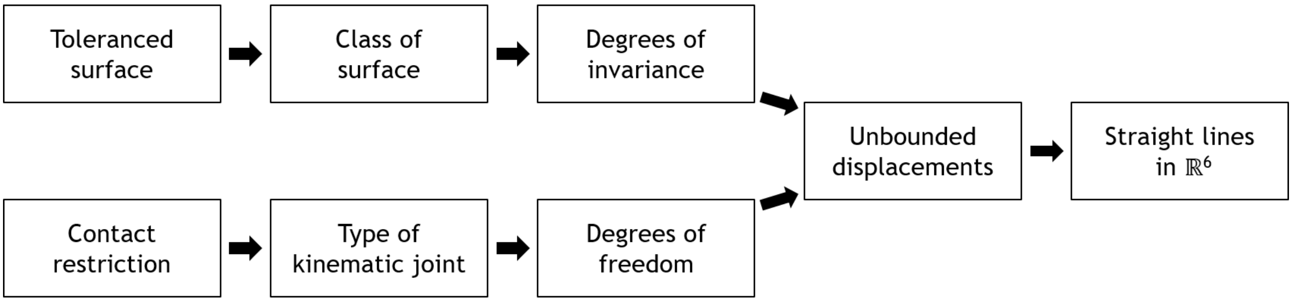


Figure 4.4: Flow chart of the decomposition of polyhedra (Arroyave-Tobón et al., 2017a).

4.3 Sum of prismatic polyhedra

The polyhedra decomposition proposed in the previous section gives way to simplified techniques for calculating accumulation of manufacturing defects in mechanical systems. Instead of operating polyhedra in $\mathbb{R}^6$ or its corresponding capped polytopes, we propose to deal with simplified sets of constraints by excluding the straight lines derived from the unbounded displacements (according to Tables 4.2, 4.3 and 4.4).

Even if it is possible to directly sum the underlying polytopes of prismatic polyhedra, we found that it is more efficient to perform the sum of their projections in the subspace common to the operands. Both strategies are detailed next.

4.3.1 Summing the underlying polytopes of two prismatic polyhedra

Let Γ_1 and Γ_2 be two prismatic polyhedra. According to Eq. (4.6) we have:

$$\Gamma_1 \oplus \Gamma_2 = P_1 \oplus P_2 \oplus C_1 \oplus C_2 \tag{4.8}$$

We can isolate the two polyhedral cones C_1 and C_2 from the sum and calculate directly $P_1 \oplus P_2$. This sum is, in general, a sum of two non-full-dimensional polytopes of $\mathbb{R}^n$ (with $1 \leq n \leq 6$). In that case, the way we use the truncation algorithm based on intersection of normal fans (described in Section 2.3.2) is not suitable. This because the normal cones to be intersected are also non-full-dimensional.

Other algorithms, such as those presented in (Delos and Teissandier, 2015c) and in (Fukuda, 2004), are able to sum non-dimensional polytopes. The first algorithm takes advantage of the property of the uniqueness of the Minkowski vertices decomposition for computing only the vertices defining the convex hull of the resulting polytope. The second one exploits the adjacency properties of the operands vertices for computing only the Minkowski vertices of the resulting polytope.

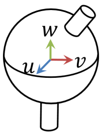
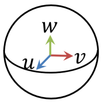
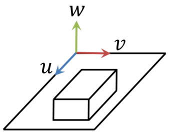
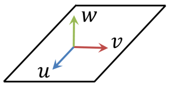
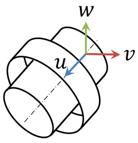
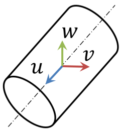
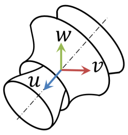
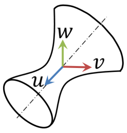
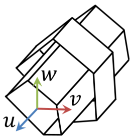
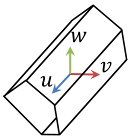
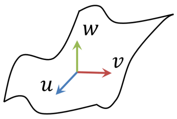
Type of kinematic joint	Class of toleranced surface	Tolerancing constraints	Lines of unbounded displacements ($\lambda \in \mathbb{R}$)
Spherical pair 	Spherical surface 	$P = \bigcap_{k=1}^{2m} \bar{H}_k^+ \bigcap H_{\Delta_1} \bigcap H_{\Delta_2} \bigcap H_{\Delta_3}$	$\Delta_1 = \lambda\hat{T}_{r_u} = \lambda[\boldsymbol{u} \mid \boldsymbol{N}_i\boldsymbol{M} \times \boldsymbol{u}]$ $\Delta_2 = \lambda\hat{T}_{r_v} = \lambda[\boldsymbol{v} \mid \boldsymbol{N}_i\boldsymbol{M} \times \boldsymbol{v}]$ $\Delta_3 = \lambda\hat{T}_{r_w} = \lambda[\boldsymbol{w} \mid \boldsymbol{N}_i\boldsymbol{M} \times \boldsymbol{w}]$
Planar pair 	Plane surface 	$P = \bigcap_{k=1}^{2m} \bar{H}_k^+ \bigcap H_{\Delta_3} \bigcap H_{\Delta_4} \bigcap H_{\Delta_5}$	$\Delta_3 = \lambda\hat{T}_{r_w} = \lambda[\boldsymbol{w} \mid \boldsymbol{0}]$ $\Delta_4 = \lambda\hat{T}_{t_u} = \lambda[\boldsymbol{0} \mid \boldsymbol{u}]$ $\Delta_5 = \lambda\hat{T}_{t_v} = \lambda[\boldsymbol{0} \mid \boldsymbol{v}]$
Cylindrical pair 	Cylindrical surface 	$P = \bigcap_{k=1}^{2m} \bar{H}_k^+ \bigcap H_{\Delta_1} \bigcap H_{\Delta_4}$	$\Delta_1 = \lambda\hat{T}_{r_u} = \lambda[\boldsymbol{u} \mid \boldsymbol{N}_i\boldsymbol{M} \times \boldsymbol{u}]$ $\Delta_4 = \lambda\hat{T}_{t_u} = \lambda[\boldsymbol{0} \mid \boldsymbol{u}]$
Revolute pair 	Surface of revolution 	$P = \bigcap_{k=1}^{2m} \bar{H}_k^+ \bigcap H_{\Delta_1}$	$\Delta_1 = \lambda\hat{T}_{r_u} = \lambda[\boldsymbol{u} \mid \boldsymbol{N}_i\boldsymbol{M} \times \boldsymbol{u}]$
Prismatic pair 	Prismatic surface 	$P = \bigcap_{k=1}^{2m} \bar{H}_k^+ \bigcap H_{\Delta_4}$	$\Delta_4 = \lambda\hat{T}_{t_u} = \lambda[\boldsymbol{0} \mid \boldsymbol{u}]$
N/A	Complex surface 	$P = \bigcap_{k=1}^{2m} \bar{H}_k^+$	N/A

Table 4.2: Decomposition of polyhedra derived from surfaces (Arroyave-Tobón et al., 2017a).

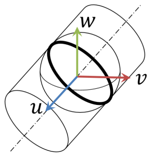
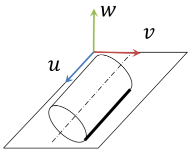
Type of kinematic joint	Tolerancing constraints	Lines of unbounded displacements ($\lambda \in \mathbb{R}$)
Ball-and-cylinder pair  	$P = \bigcap_{k=1}^{2m} \bar{H}_k^+ \bigcap H_{\Delta_1} \bigcap H_{\Delta_2}$ $\bigcap H_{\Delta_3} \bigcap H_{\Delta_4}$	$\Delta_1 = \lambda \hat{T}_{r_u} = \lambda[\boldsymbol{u} \mid \boldsymbol{N}_i\boldsymbol{M} \times \boldsymbol{u}]$ $\Delta_2 = \lambda \hat{T}_{r_v} = \lambda[\boldsymbol{v} \mid \boldsymbol{N}_i\boldsymbol{M} \times \boldsymbol{v}]$ $\Delta_3 = \lambda \hat{T}_{r_w} = \lambda[\boldsymbol{w} \mid \boldsymbol{N}_i\boldsymbol{M} \times \boldsymbol{w}]$ $\Delta_4 = \lambda \hat{T}_{t_u} = \lambda[\boldsymbol{0} \mid \boldsymbol{u}]$,
Cylinder-and-plane pair  	$P = \bigcap_{k=1}^{2m} \bar{H}_k^+ \bigcap H_{\Delta_1} \bigcap H_{\Delta_3}$ $\bigcap H_{\Delta_4} \bigcap H_{\Delta_5}$ with $m = 2$	$\Delta_1 = \lambda \hat{T}_{r_u} = \lambda[\boldsymbol{u}, \boldsymbol{N}_i\boldsymbol{M} \times \boldsymbol{u}]$ $\Delta_3 = \lambda \hat{T}_{r_w} = \lambda[\boldsymbol{w} \mid \boldsymbol{0}]$ $\Delta_4 = \lambda \hat{T}_{t_u} = \lambda[\boldsymbol{0} \mid \boldsymbol{u}]$ $\Delta_5 = \lambda \hat{T}_{t_v} = \lambda[\boldsymbol{0} \mid \boldsymbol{v}]$

Table 4.3: Decomposition of polyhedra derived from linear features (Arroyave-Tobón et al., 2017a).

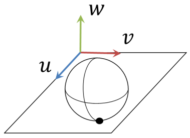
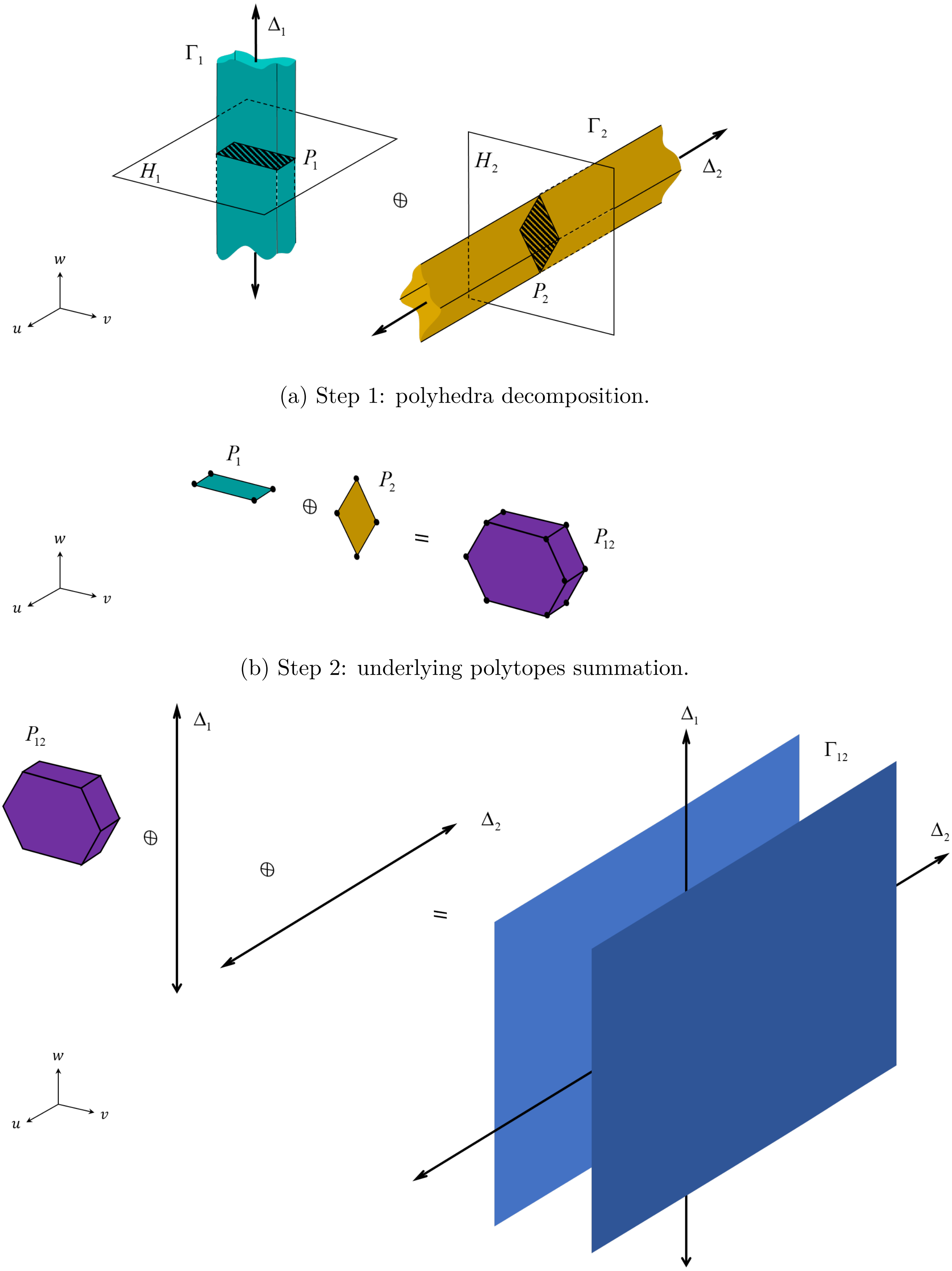
Type of kinematic joint	Tolerancing constraints	Lines of unbounded displacements ($\lambda \in \mathbb{R}$)
Ball-and-plane pair  	$P = \bigcap_{k=1}^{2m} \bar{H}_k^+ \bigcap H_{\Delta_1} \bigcap H_{\Delta_2}$ $\bigcap H_{\Delta_3} \bigcap H_{\Delta_4} \bigcap H_{\Delta_5}$ with $m = 1$	$\Delta_1 = \lambda \hat{T}_{r_u} = \lambda[\boldsymbol{u} \mid \boldsymbol{0}]$ $\Delta_2 = \lambda \hat{T}_{r_v} = \lambda[\boldsymbol{v} \mid \boldsymbol{0}]$ $\Delta_3 = \lambda \hat{T}_{r_w} = \lambda[\boldsymbol{w} \mid \boldsymbol{0}]$ $\Delta_4 = \lambda \hat{T}_{t_u} = \lambda[\boldsymbol{0} \mid \boldsymbol{u}]$ $\Delta_5 = \lambda \hat{T}_{t_v} = \lambda[\boldsymbol{0} \mid \boldsymbol{v}]$

Table 4.4: Decomposition of polyhedra derived from punctual contacts (Arroyave-Tobón et al., 2017a).

Considering the above, the synthesis of an algorithm to compute the sum of decomposed polyhedra is straightforward (see Algorithm 4).

Let us use the example presented in Figure 4.5 to illustrate the algorithm. As a first step, the prismatic polyhedra are decomposed, isolating their straight lines (see Figure 4.5a). Then, the sum of the polytopes is computed using their $\mathcal{V}$-description. This allowed to obtain P_{12}, an underlying polytope of the polyhedron Γ_R (see Figure 4.5b). Finally, P_{12} is extruded along the straight lines Δ_1 and Δ_2 (see Figure 4.5c) to generate Γ_{12} (two parallel planes).

Notice that in the last step, many vertices of the calculated polytope P_{12} are finally placed at the infinity. In other words, some data computed during an intermediary step is removed after; which is a similar situation of that described in Chapter 3. The most of the information contained in the calculated polytope P_{12} has not meaning for the final result. Just the limits along the $\boldsymbol{v}$-axis are important in this case. The rest of the information is after absorbed during the extrusion process. Therefore, a much more simpler polytope (actually a 1-dimensional) can represent the polyhedron Γ_{12}. Furthermore, this strategy involves the computation of the



(a) Step 1: polyhedra decomposition.

(b) Step 2: underlying polytopes summation.

(c) Step 3: straight lines addition.

Figure 4.5: Summing the underlying polytopes of two polyhedra.

Algorithm 4 Sum of decomposed polyhedra

```
Require: $\Gamma_1$, $\Gamma_2$
Ensure: $\Gamma_R = \Gamma_1 \oplus \Gamma_2$
decompose $\Gamma_1 = P_1 \oplus C_1$
decompose $\Gamma_2 = P_2 \oplus C_2$
if $\dim(C_1 \oplus C_2) = 6$ then
    $\Gamma_R = \mathbb{R}^6$
else
    compute $\mathcal{V}$-description of $P_1$, $\mathcal{V}_1 = conv(\boldsymbol{a_1}, ..., \boldsymbol{a_r})$
    compute $\mathcal{V}$-description of $P_2$, $\mathcal{V}_2 = conv(\boldsymbol{b_1}, ..., \boldsymbol{b_s})$
    compute sum $\mathcal{V}_R = \mathcal{V}_1 \oplus \mathcal{V}_2$
    $\Gamma_R = \mathcal{V}_R \oplus C_1 \oplus C_2$
end if
```

convex hull of the calculated cloud of points to obtain the $\mathcal{H}$-description.

A way to avoid these problems is computing the sum in the subspace common to the operand polyhedra, as we explain next.

4.3.2 Sum of projections of decomposed polyhedra

Theorem 4.3.1 (Sum of prismatic polyhedra) *Let Γ_1 and Γ_2 be two prismatic polyhedra in $\mathbb{R}^n$ such that:*

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

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

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

The sum $\Gamma_1 \oplus \Gamma_2$ 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 represents the orthogonal projection on the space H.

Successive orthogonal projection

Without loss of generality we're going to prove the theorem for $\Gamma_1 = P_1 \oplus \Delta_1$ and $\Gamma_2 = P_2 \oplus \Delta_2$ being prismatic polyhedra spanned by only one straight line, with Δ_1 and Δ_2 not parallel and consequently H_1 and H_2 also not parallel:

$$\Gamma_1 \oplus \Gamma_2 = P_1 \oplus P_2 \oplus \Delta_1 \oplus \Delta_2 \text{ with } \Delta_1 \text{ and } \Delta_2 \text{ not parallel}$$

Let us prove, first, that:

$$P \oplus \Delta = \pi_H(P) \oplus \Delta \text{ with } \Delta \perp H$$

If $\boldsymbol{x} \in P$ then $\pi_H(\boldsymbol{x}) = \boldsymbol{x} \ - (\delta_0 + \boldsymbol{\delta} \cdot \boldsymbol{x})\boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is a unit vector of Δ and δ_0 the constant of H, so this is the sum of a point of P and a vector of Δ (see Figure 4.6).

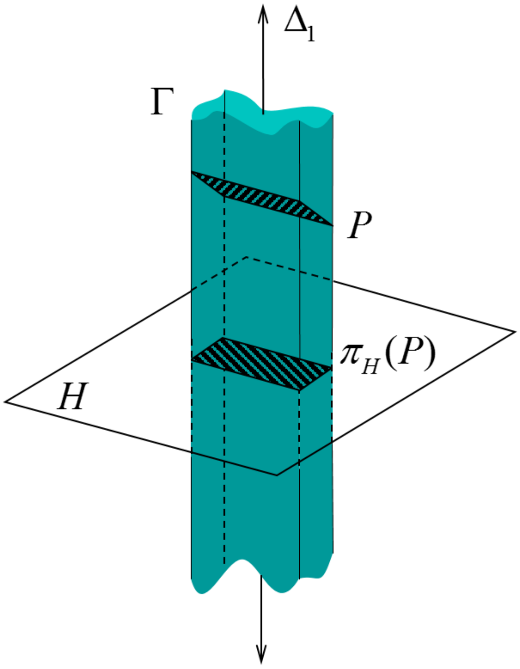


Figure 4.6: Illustration of $P \oplus \Delta = \pi_H(P) \oplus \Delta$.

$$\begin{aligned} P_1 \oplus \Delta_1 \oplus \Delta_2 &= \pi_{H_1}(P_1) \oplus \Delta_1 \oplus \Delta_2 \\ &= \pi_{H_1}(P_1) \oplus \Delta_2 \oplus \Delta_1 \end{aligned}$$

Since $\pi_{H_1}(P_1)$ is also a polytope, we can follow the same logic to get:

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

As a consequence we can perform as many projections on H_1 and then on H_2 as we want:

$$P_1 \oplus \Delta_1 \oplus \Delta_2 = \pi_{H_2}\Big(\pi_{H_1}\Big(...\pi_{H_2}(\pi_{H_1}(P_1))\Big)\Big) \oplus \Delta_1 \oplus \Delta_2 \tag{4.9}$$

To prove the theorem 4.3.1 we need to know what is the limit of all these successive projections $\pi_{H_2}\Big(\pi_{H_1}\Big(...\pi_{H_2}(\pi_{H_1}(P_1))\Big)\Big)$.

According to Cheney and Goldstein (1959) and Boyd and Dattorro (2003), projecting a point $\boldsymbol{x}$ alternately on two convex sets H_1 and H_2 (with $H_1 \cap H_2 \neq \emptyset$) converges towards a point $\boldsymbol{x}^* \in H_1 \cap H_2$ whatever the initial point is (see Figure 4.7).

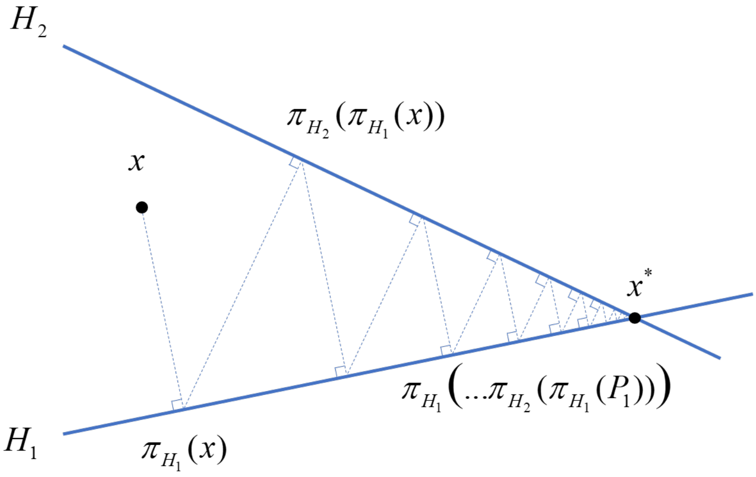

H_2
$\pi_{H_2}(\pi_{H_1}(x))$
x
x^*
$\pi_{H_1}\big(...\pi_{H_2}(\pi_{H_1}(P_1))\big)$
H_1
$\pi_{H_1}(x)$

Figure 4.7: Successive projections in a 2D vector space.

Using this property in equation (4.9) provides:

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

By considering the same for P_2, we obtain:

$$P_1 \oplus P_2 \oplus \Delta_1 \oplus \Delta_2 = \pi_{H_1 \cap H_2}(P_1) \oplus \pi_{H_1 \cap H_2}(P_2) \oplus \Delta_1 \oplus \Delta_2$$

Following the same logic, we can include another straight line Δ_3:

$$P_1 \oplus P_2 \oplus \Delta_1 \oplus \Delta_2 \oplus \Delta_3 = \pi_{H_1 \cap H_2 \cap H_3}(P_1) \oplus \pi_{H_1 \cap H_2 \cap H_3}(P_2) \oplus \Delta_1 \oplus \Delta_2 \oplus \Delta_3$$

We can finally prove by recurrence Theorem 4.3.1:

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

4.3.3 Algorithm: projection-based sum

In a sum of prismatic polyhedra, instead of summing directly the underlying polytopes (as suggested above in Algorithm 4), we propose to compute, first, the common subspace to the operands and compute after the sum into it (see Algorithm 5).

Algorithm 5 Projection-based sum

```
Require: Set of $n_p$ prismatic polyhedra $\Gamma_i$ in $\mathbb{R}^n$
Ensure: $\Gamma_R = \sum_{i=1}^{n_p} \Gamma_i$
for each polyhedron $\Gamma_i$ $(i = 1 : n_p)$ do        // Decompose each operand
    decompose $\Gamma_i = P_i \oplus C_i$
    compute $H_i$ from $C_i$
end for
compute $H_R = \bigcap_{i=1}^{n_p} H_i$
if $\dim(\sum_{i=1}^{n_p} C_i) = n$ then        // Check the feasibility of the sum
    $\Gamma_R = \mathbb{R}^n$
else
    $P_R = \pi_{H_R}(P_1)$
    for each polytope $P_i$ $(i = 2 : n_p)$ do        // Project and sum each operand
        compute projection $\pi_{H_R}(P_i)$
        compute $P_R = P_R \oplus \pi_{H_R}(P_i)$
    end for
    compute $\Gamma_R = P_R \oplus \sum_{i=1}^{n_p} C_i$        // Go back to the original space $\mathbb{R}^n$
end if
```

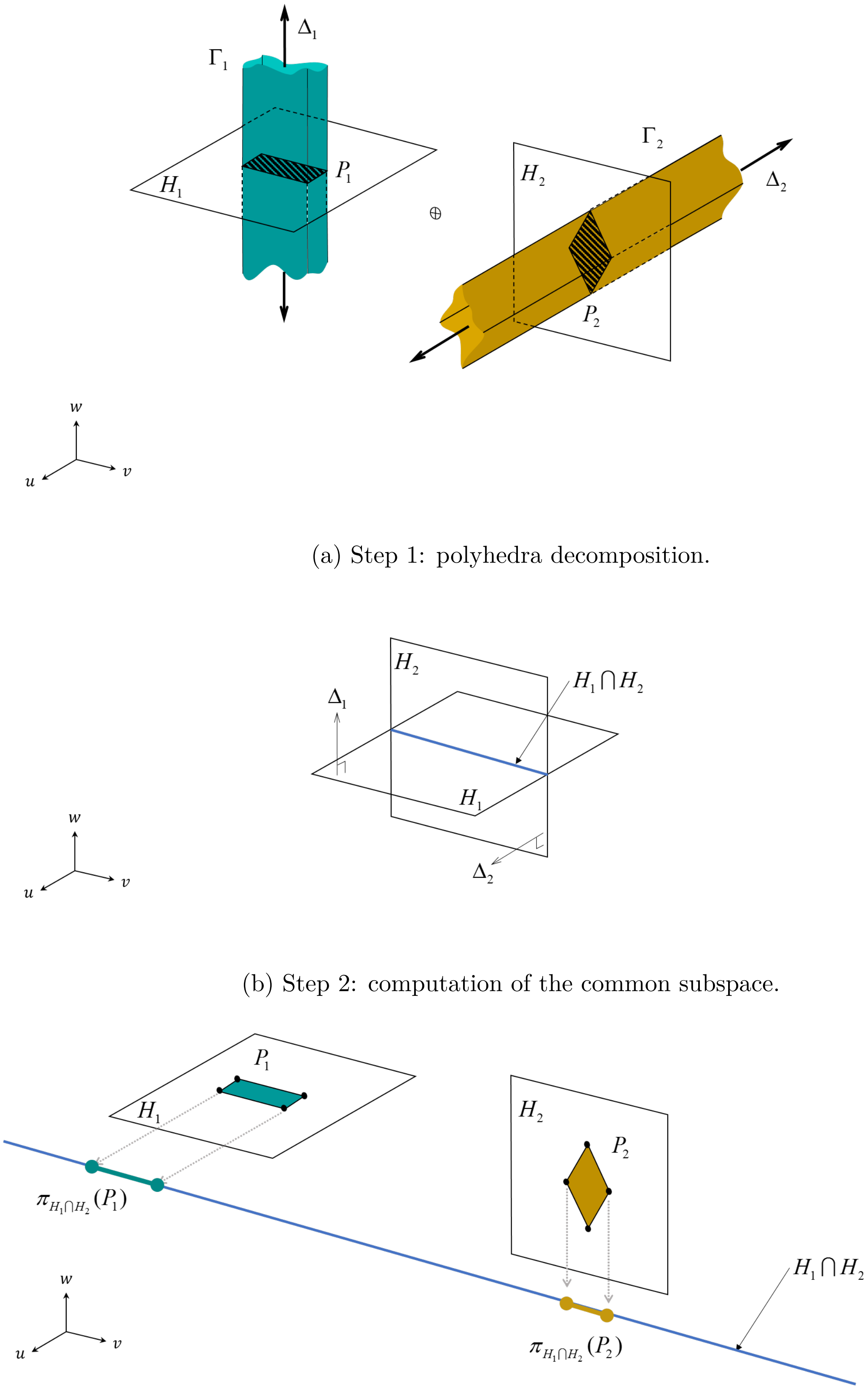
In order to illustrate the proposed procedure, let us pick up the example in Figure 4.5. In this case, instead of summing directly P_1 and P_2, we compute first the intersection of the subspaces in which P_1 and P_2 live (see Figure 4.8b). As H_1 and H_2 are secant planes, their intersection is a straight line. This line represents the greatest subspace in which the sum of the projections of P_1 and P_2 generates a bounded set, i.e. a polytope. P_1 and P_2 are then projected on $H_1 \cap H_2$ (see Figure 4.8c), for computing after $P_R = \pi_{H_1 \cap H_2}(P_1) \oplus \pi_{H_1 \cap H_2}(P_2)$ (see Figure 4.8d). Finally, the straight lines Δ_1 and Δ_2 are added to P_R to obtain Γ_R (see Figure 4.8e). In this case, Γ_R is composed by two parallel planes which impose limits only along the $\boldsymbol{v}$-axis.

In comparison with the procedure carried out in Section 4.3.1, we could reduced a sum of 3D polytopes to a sum of 1D polytopes, for finally obtaining the same result.

4.4 Simulation feasibility test

When the tolerance chain involve just few parts, the designer is able to anticipate without the need of formal calculations if the mobility conditions of the assembly allow to satisfy the functional requirements. In a general case, however, this is not an easy task. Therefore, the designer is forced to run numerical simulations even without knowing if they are feasible.

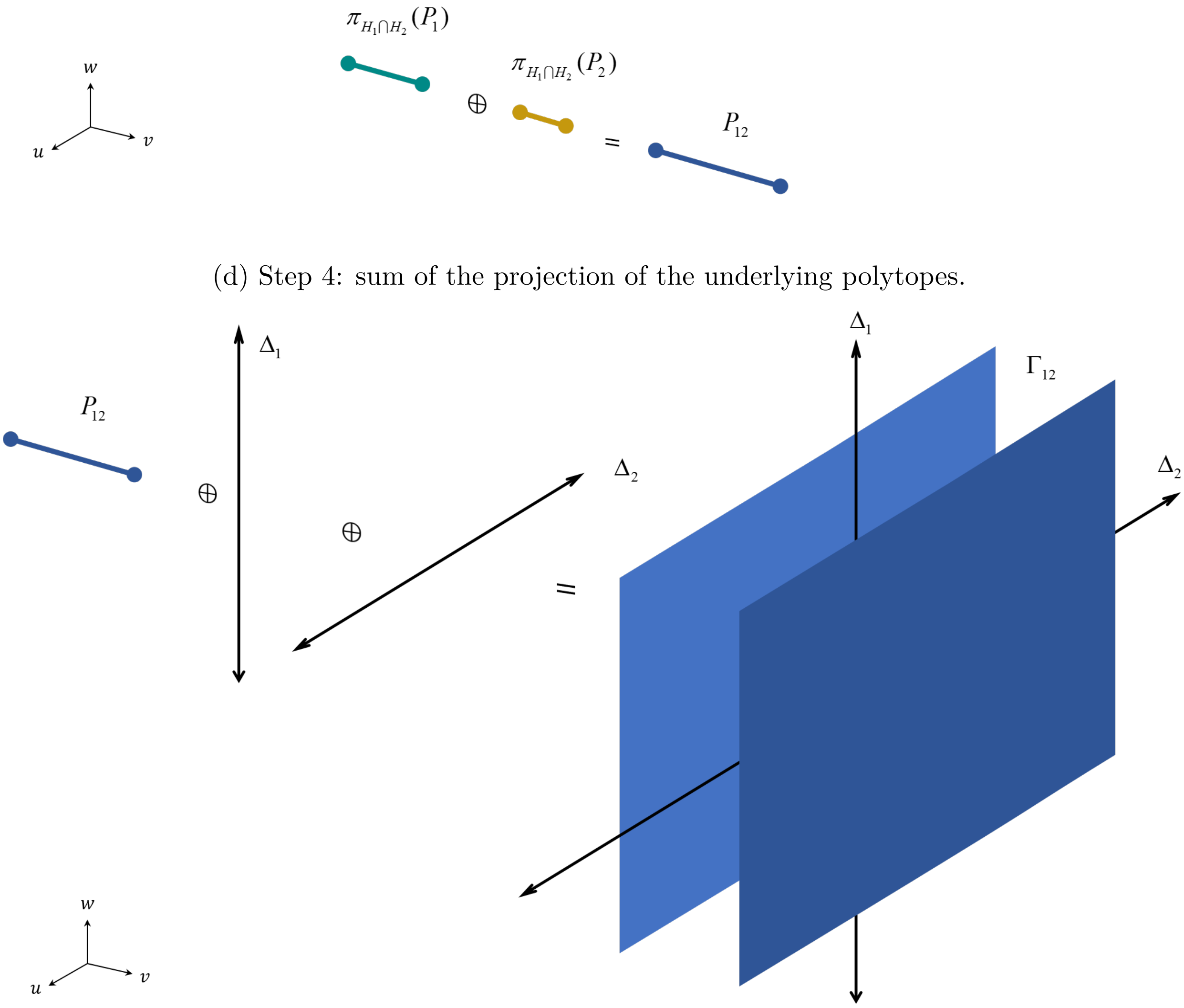
Performing kinematic analysis during tolerance analysis opens the possibility of anticipating, before starting numerical simulations, if the functional requirements of the assembly may be



(a) Step 1: polyhedra decomposition.

(b) Step 2: computation of the common subspace.

(c) Step 3: projection of the underlying polytopes.



(d) Step 4: sum of the projection of the underlying polytopes.

(e) Step 5: straight lines addition.

Figure 4.8: Summing the projection of the underlying polytopes of two polyhedra.

satisfied. This can be carried out by comparing mobility of the whole system respecting with that restricted by the functional specifications, as described next.

Let $\mathcal{W}_R$ be wrench-space characterizing the mobility conditions of a tolerance chain and $\mathcal{W}_F$ be the one related to the restriction of the functional requirements. The limits established by the FC may be satisfied if and only if $\mathcal{W}_F \subseteq \mathcal{W}_R$.

According to this, we propose to include a feasibility test in the tolerance analysis process, as presented in Figure 4.9.

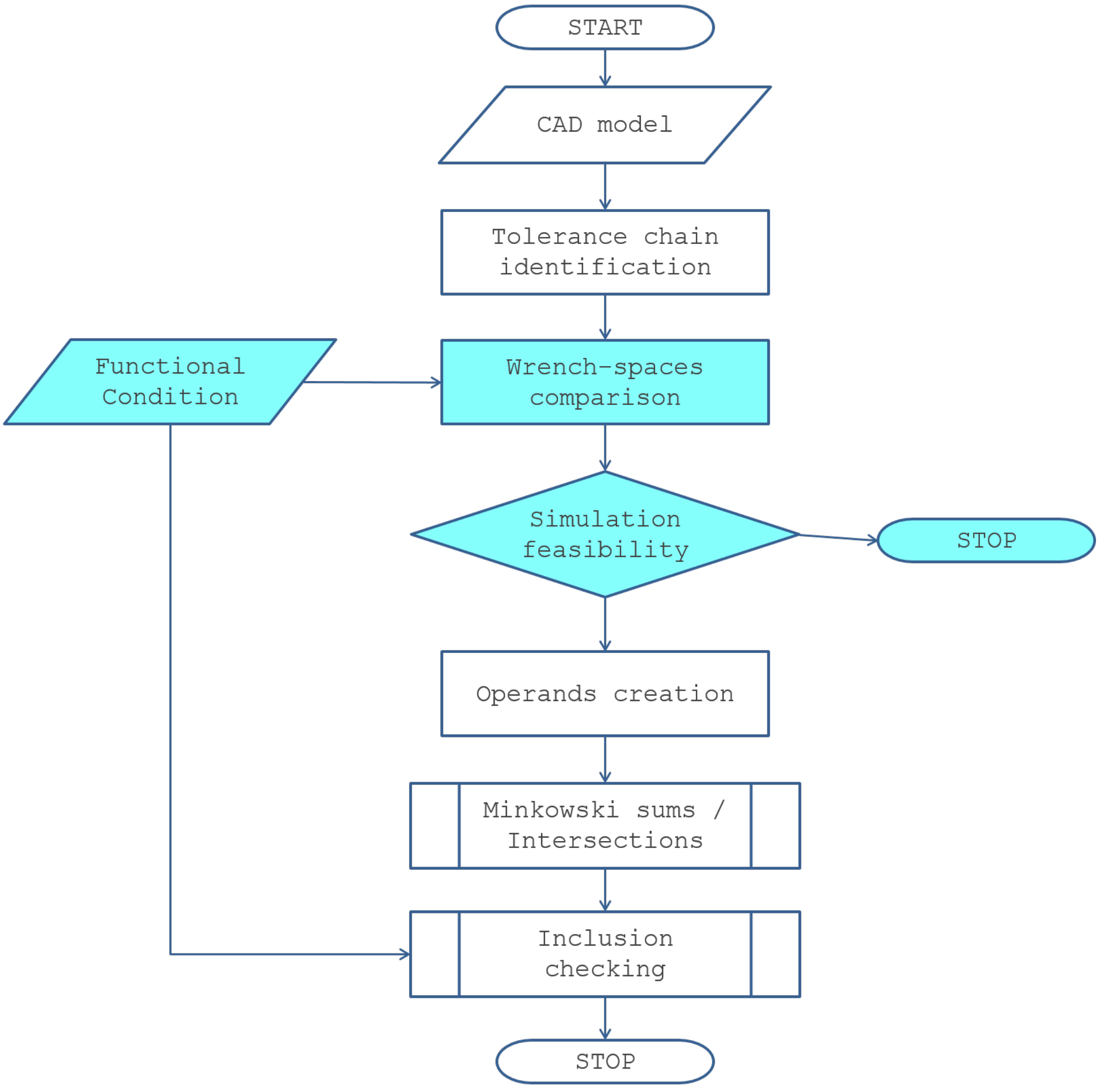


Figure 4.9: Flowchart of the proposed methodology considering the simulation feasibility.

4.5 Prismatic polyhedra and ISO standards compatibility

Tolerancing approaches based on point-to-point constraint solving are partially compliant with the tolerance standards that specify variation within tolerance zones (Ameta et al., 2011).

Approaches based on set of constraints are precisely created for representing tolerances zones derived from manufacturing specifications. However, some special cases defined in the standards can not be directly model following the regular constraints definition process. We identified two common situations generating these special cases and propose a solution taking advantage of the notion of prismatic polyhedra.

4.5.1 Not fully constrained tolerance zones

The regular process for turning geometric constraints into algebraic ones is based on the nominal definition of features. Therefore, following the regular procedure, the obtained tolerance zones inherit the degrees of invariance of their related feature.

In some cases defined in the standards, however, the nominal geometry of the feature is not the same of that of the tolerance zone. A similar situation occurs when the tolerance zone is not fully constrained. This entails that the displacements allowed by the tolerance zone do not coincide with the degrees of invariance of the toleranced feature. This occurs, for example, in the cases explained bellow.

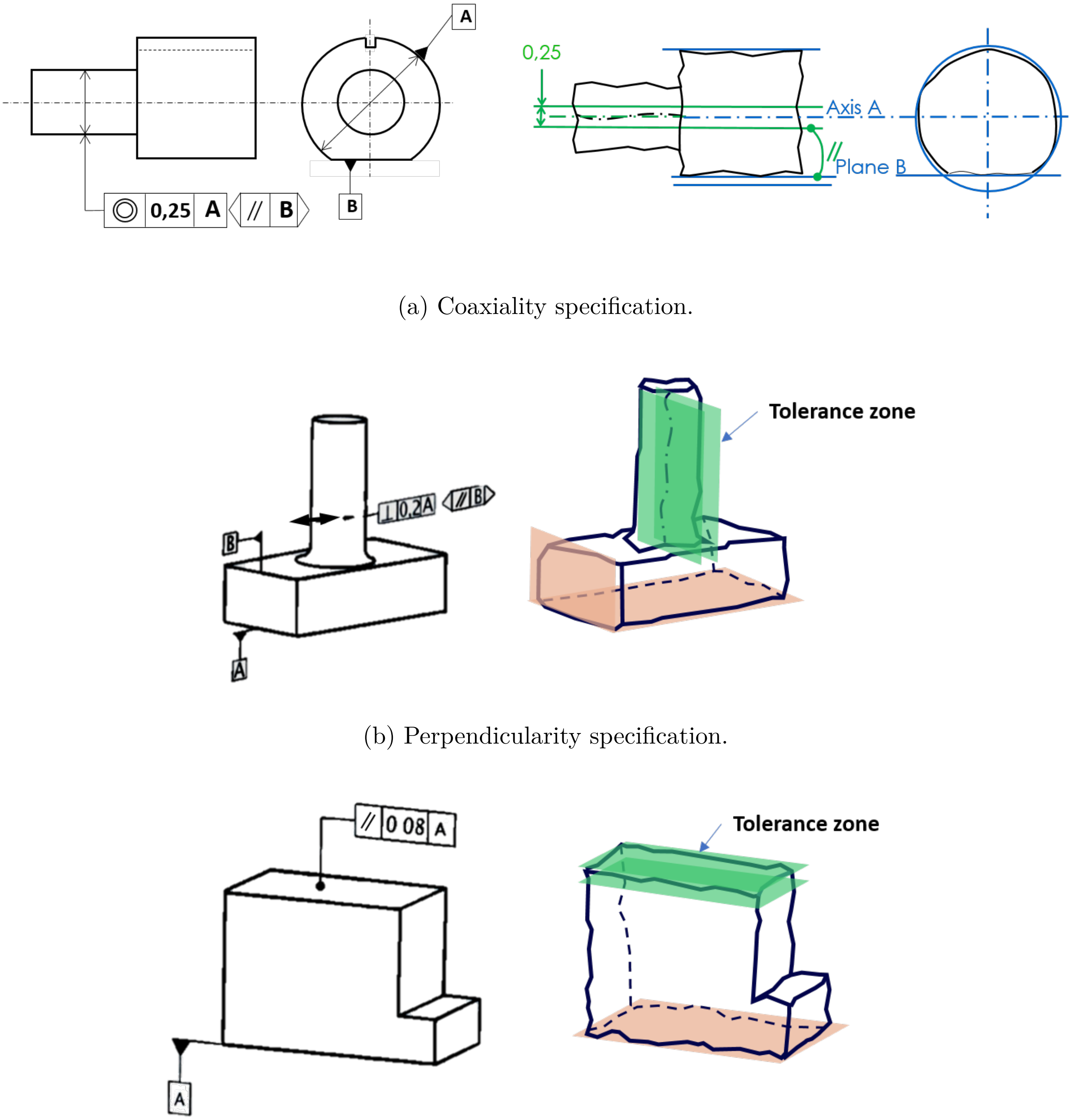
Let us take the case of the coaxiality specification illustrated in Figure 4.10a. By following the regular process in tolerancing by set of constraints, a cylindrical tolerance zone is generated. However, the tolerance zone specified in this case is instead defined by two parallel planes. A similar situation, but with orientation specification, occurs in the case shown in Figure 4.10b.

In the case of a parallelism specification (see Figure 4.10c), the tolerance zone is made up of the same surface type of the tolerance feature. However, their unbounded displacements do not coincide. The tolerance zone is not constrained along the normal nominal plane, even if this displacement does not leave invariant the surface.

The situation described above can be faced handling the derived sets of constrants as prismatic polyhedra. The additional unbounded displacements can be included to the set of geometric constraints derived from the nominal feature until obtain the same mobility conditions of the tolerance zone. This implies an addition of straight lines to a polyhedron, or equivalently, an extrusion of the polyhedron along the straight lines.

Let us go back to the example of the parallelism specification to illustrate the proposed solution. Once the set of constraints derived from the toleranced plane is defined (polytope P) we introduce a straight line $\Delta = \lambda \hat{\boldsymbol{T}}_{\boldsymbol{t_z}}$ (with $\lambda \in \mathbb{R}$) representing the free translation of the tolerance zone along the $\boldsymbol{z}$-axis (see Figure 4.11a). In order to simplify the definition of the new polyhedron, a projection $\pi_H(P)$ can be performed, where $H = \Delta^{\perp}$ (see Figure 4.11b). Finally, the extrusion $\pi_H(P) \oplus \Delta$ generates Γ, the new polyhedron representing the parallelism specification (see Figure 4.11c).

The same procedure can be carried out in the same manner for the other cases (perpendicularity and angularity).

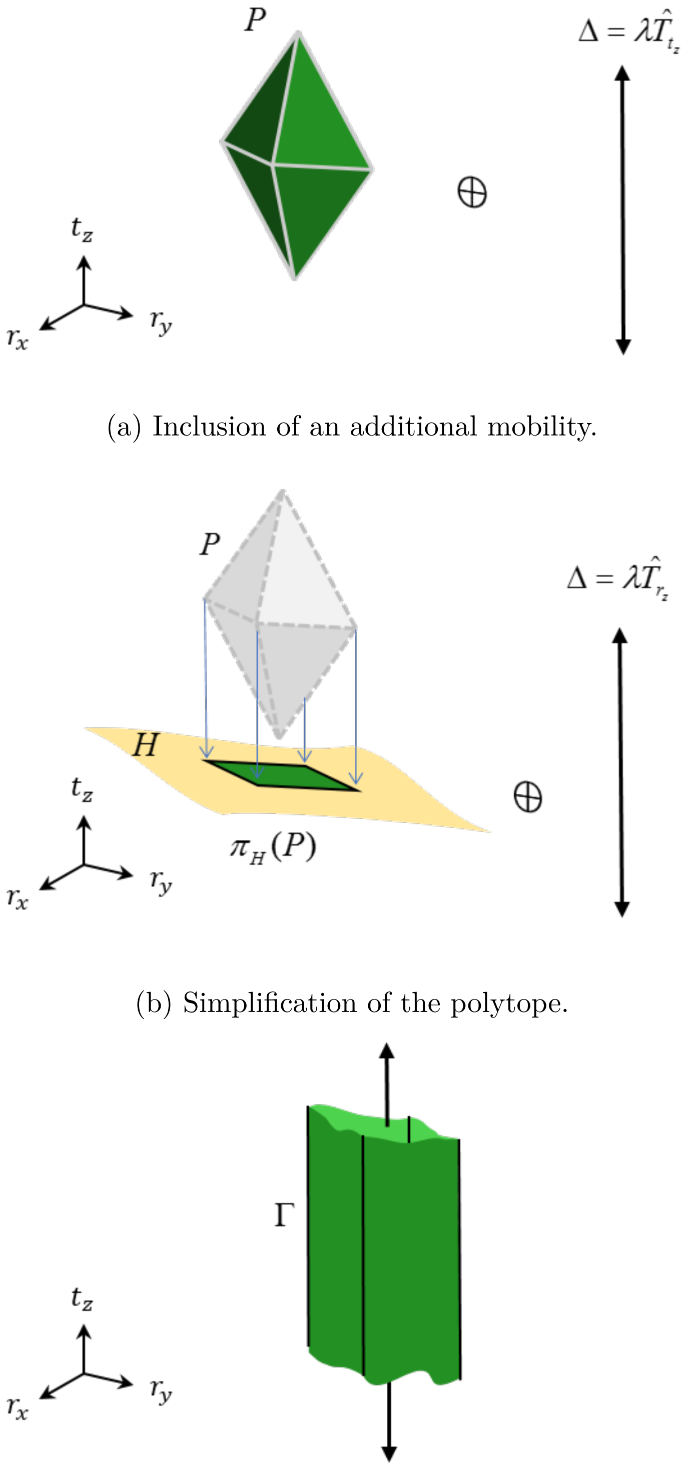


(a) Coaxiality specification.

(b) Perpendicularity specification.

(c) Parallelism specification.

Figure 4.10: Not fully constrained tolerance zones (Charpentier, 2012).



(a) Inclusion of an additional mobility.

(b) Simplification of the polytope.

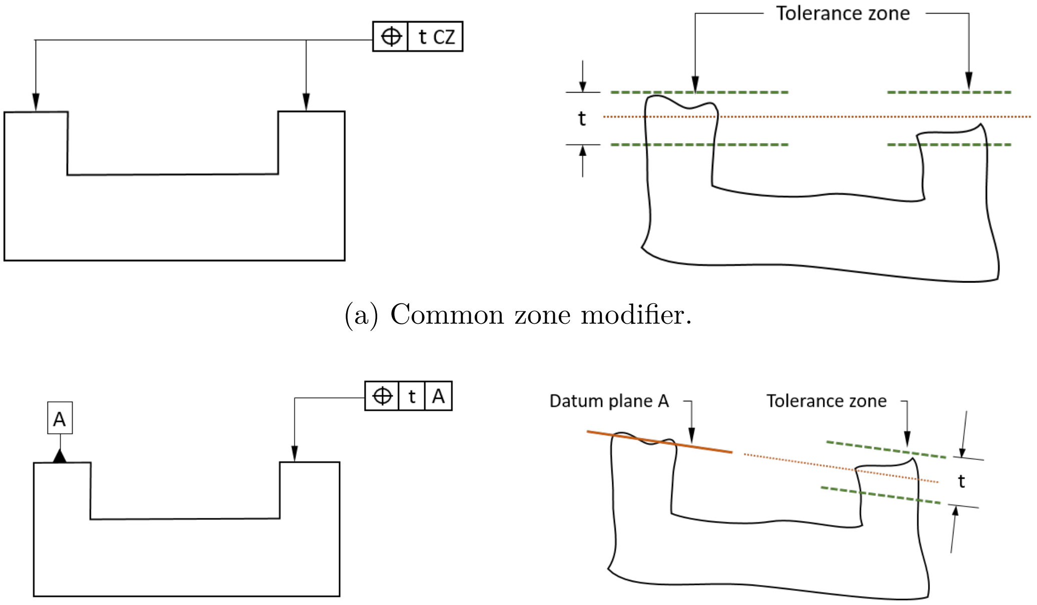
(c) Polytope extrusion.

Figure 4.11: Inclusion of an additional mobility.

4.5.2 Datum features

Other situation demanding a special treatment occurs when a feature is used as datum. In this case, a tolerance zone with zero value is generated, but its mobility remains. In algebraic terms it is equivalent to have a set of half-spaces of $\mathbb{R}^6$ whose second member is zero, i.e. passing through the origin. In such a case, the obtained set can be wrongly consider as empty. In tolerance simulation, what is important to consider of datum features is the mobility they introduce to the kinematic chain. This mobility can also be treated in terms of screws.

In some cases, this situation can be avoided using a ‘common zone’ modifier. As illustrated in Figure 4.12a, for example, each surface has a tolerance zone which can be modelled with polytopes following the regular procedure. The accumulation of defects can be calculated computing the Minkowski sum of both polytopes.



(a) Common zone modifier.

(b) Position specification.

Figure 4.12: Tolerance zone definition with the common zone (CZ) modifier.

However, when one of the surfaces is used as datum (see Figure 4.12b), the situation is different. For the feature within a tolerance zone, a full dimensional polyhedron can be obtained; and for the datum feature, three pairs of half-spaces of $\mathbb{R}^6$ passing through the origin are obtained. This situation can be well modelled using prismatic polyhedra. The tolerance zone with zero value represents a point (the origin) in the wrench-space. While the mobility represents some straight lines in the twist-space. Thus, the point plus the straight lines represent the generated polyhedron. For this case, the polyhedron corresponds to a 2D plane in $\mathbb{R}^6$.

4.6 Case study: solution by projection-based method

To illustrate the concepts exposed throughout this chapter chapter, let us go back to the case study of the brake system (Section 2.6). The tolerance simulation was carried out following the proposed kinematic decomposition of geometric and contact polyhedra.

4.6.1 Kinematic analysis and feasibility test

The mobility analysis of the mechanism was conducted according the topology of the assembly (described by the graph in Figure 2.24). All the toleranced features and joints, and their interaction, were simulated with screws and operations on them. The centroid of surface 2,1 was chosen as the point to express the screws for the mobility analysis. For consistence, it is the same point used to express the polytopes in Section 2.6.

The required operations with screws are equivalent to those with polytopes defined in Eqs. (2.10) to (2.22). Following the same notation used for the polyhedra, $\mathcal{T}_{a,b/c,d}$ is the twist-matrix representing the mobility conditions of the surface b from the part a with respect to the surface d of the part c. The mobility of the whole tolerance chain can be calculated as:

$$\mathcal{T}_R = \mathcal{T}_{1,1/6,1} = \mathcal{T}_{1,1/1,0} \cup \mathcal{T}_{1,0/3,0} \cup \mathcal{T}_{3,0/4,0} \cup \mathcal{T}_{4,0/6,0} \cup \mathcal{T}_{6,0/6,1} \tag{4.10}$$

where:

$$\mathcal{T}_{1,0/3,0} = \mathcal{T}_{1,0/3,0-a} \cap \mathcal{T}_{1,0/3,0-b} \cap \mathcal{T}_{1,0/3,0-c} \tag{4.11}$$
$$\mathcal{T}_{1,0/3,0-a} = \mathcal{T}_{1,0/1,2} \cup \mathcal{T}_{1,2/3,2} \cup \mathcal{T}_{3,2/3,0} \tag{4.12}$$
$$\mathcal{T}_{1,0/3,0-b} = \mathcal{T}_{1,0/1,3} \cup \mathcal{T}_{1,3/3,3} \cup \mathcal{T}_{3,3/3,0} \tag{4.13}$$
$$\mathcal{T}_{1,0/3,0-c} = \mathcal{T}_{1,0/1,4} \cup \mathcal{T}_{1,4/2,2} \cup \mathcal{T}_{2,2/2,0} \cup \mathcal{T}_{2,0/2,1} \cup \mathcal{T}_{2,1/3,1} \cup \mathcal{T}_{3,1/3,0} \tag{4.14}$$
$$\mathcal{T}_{3,0/4,0} = \mathcal{T}_{3,0/4,0-a} \cap \mathcal{T}_{3,0/4,0-b} \cap \mathcal{T}_{3,0/4,0-c} \tag{4.15}$$
$$\mathcal{T}_{3,0/4,0-a} = \mathcal{T}_{3,0/3,4} \cup \mathcal{T}_{3,4/4,4} \cup \mathcal{T}_{4,4/4,0} \tag{4.16}$$
$$\mathcal{T}_{3,0/4,0-b} = \mathcal{T}_{3,0/3,5} \cup \mathcal{T}_{3,5/4,5} \cup \mathcal{T}_{4,5/4,0} \tag{4.17}$$
$$\mathcal{T}_{3,0/4,0-c} = \mathcal{T}_{3,0/3,6} \cup \mathcal{T}_{3,6/4,6} \cup \mathcal{T}_{4,6/4,0} \tag{4.18}$$
$$\mathcal{T}_{4,0/6,0} = \mathcal{T}_{4,0/6,0-a} \cap \mathcal{T}_{4,0/6,0-b} \cap \mathcal{T}_{4,0/6,0-c} \tag{4.19}$$
$$\mathcal{T}_{4,0/6,0-a} = \mathcal{T}_{4,0/4,2} \cup \mathcal{T}_{4,2/6,2} \cup \mathcal{T}_{6,2/6,0} \tag{4.20}$$
$$\mathcal{T}_{4,0/6,0-b} = \mathcal{T}_{4,0/4,3} \cup \mathcal{T}_{4,3/6,3} \cup \mathcal{T}_{6,3/6,0} \tag{4.21}$$
$$\mathcal{T}_{4,0/6,0-c} = \mathcal{T}_{4,0/4,1} \cup \mathcal{T}_{4,1/5,1} \cup \mathcal{T}_{5,1/5,0} \cup \mathcal{T}_{5,0/5,2} \cup \mathcal{T}_{5,2/6,4} \cup \mathcal{T}_{6,4/6,0} \tag{4.22}$$

Let us describe in detail the computation of the twist $\mathcal{T}_{1,0/3,0-c}$ (Eq. (4.14)). Considering that each joint has the same mobility conditions of its related geometric features, the following simplifications can be directly made without the need of numerical computations:

$$\mathcal{T}_{1,0/3,0-c} = \mathcal{T}_{1,0/2,0} \cup \mathcal{T}_{2,0/3,0}$$

Let us now describe the mobility of the joints with twist-matrices.

$\mathcal{T}_{1,0/2,0}$ derives from a planar surface with normal vector parallel to the $\boldsymbol{z}$-axis:

$$\mathcal{T}_{1,0/2,0} = \begin{bmatrix} \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{14}M} \\ \boldsymbol{0} \mid \boldsymbol{x} \\ \boldsymbol{0} \mid \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.23}$$

The cross product $\boldsymbol{z} \times \boldsymbol{O_{14}M}$ describes the translation generated by the free rotation along the $\boldsymbol{z}$-axis a the point $\boldsymbol{M}$. $\boldsymbol{O_{14}}$ could be any point over the surface 1,4.

$\mathcal{T}_{2,0/3,0}$ derives from a cylindrical surface oriented along the $\boldsymbol{z}$-axis:

$$\mathcal{T}_{2,0/3,0} = \begin{bmatrix} \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{0} \\ \boldsymbol{0} \mid \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.24}$$

The free rotation around the $\boldsymbol{z}$-axis does not generate an additional translation because the point $\boldsymbol{M}$ is over the axis of the cylindrical surface.

As the joints are linked in a serial configuration, the resulting mobility can be modelled as the union of these twist-matrices. After computing the reduced row echelon form of the generated matrix to remove the linear dependencies between vectors, we have:

$$\mathcal{T}_{1,0/3,0-c} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.25}$$

The orthogonal complement of the former matrix corresponds to the subspace of bounded displacements defining the mobility of the surface 1,4 with respect to the surface 3,1:

$$\mathcal{W}_{1,0/3,0-c} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.26}$$

Following the same logic, the mobility analysis of the whole mechanism was carried out (see Annex B.1) to obtain $\mathcal{W}_R$. It allowed us to concluded that, according with the topological structure of the assembly, it is possible to control the relative orientation of the brake shoes:

$$\mathcal{W}_R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.27}$$

As we stated in Section 2.6, the functional requirement of the brake system impose limits in the relative orientation of the brake shoes. Therefore, $\mathcal{W}_R = \mathcal{W}_F$, where $\mathcal{W}_F$ is the subspace spanned by the displacements restricted by the functional condition. This result leads us to the conclusion that the tolerance simulation is feasible.

4.6.2 Simulation run

In order to run the simulation following the strategy of the projections, the initial operands were decomposed according their degrees of invariance (for the case of toleranced features) or degrees of freedom (for the case of toleranced joints).

Let us present in detail the computation of $\Gamma_{1,0/3,0-c}$. The operands required to calculate it, $\Gamma_{2,0/3,0}$ (coming from a planar surface oriented along the $\boldsymbol{z}$-axis), $\Gamma_{1,4/2,0}$ (coming from cylindrical surface oriented along the $\boldsymbol{z}$-axis) and $\Gamma_{1,0/1,4}$ (coming from planar surface oriented along the $\boldsymbol{z}$-axis) were decomposed as:

$$\Gamma_{2,0/3,0} = P_{2,0/3,0} \oplus C_{2,0/3,0} \tag{4.28}$$

$$\Gamma_{1,4/2,0} = P_{1,4/2,0} \oplus C_{1,4/2,0} \tag{4.29}$$

$$\Gamma_{1,0/1,4} = P_{1,0/1,4} \oplus C_{1,0/1,4} \tag{4.30}$$

The polyhedral cone $C_{2,0/3,0}$ is defined by the twist-matrix in Eq. (4.24). The polyhedral cones $C_{1,4/2,0}$ and $C_{1,0/1,4}$ derive from the same surface, and therefore, can be represented by the twist-matrix in Eq. (4.23).

As it was explained above, the resulting mobility of the union of these joints corresponds to the union of their twist-matrices (Eq. (4.25)). This means that the underlying polytope of $\Gamma_{1,0/3,0-c}$ lives in such subspace. Therefore:

$$P_{1,0/3,0-c} = \pi_H(P_{2,0/3,0}) \oplus \pi_H(P_{1,4/2,0}) \oplus \pi_H(P_{1,0/1,4}) \tag{4.31}$$

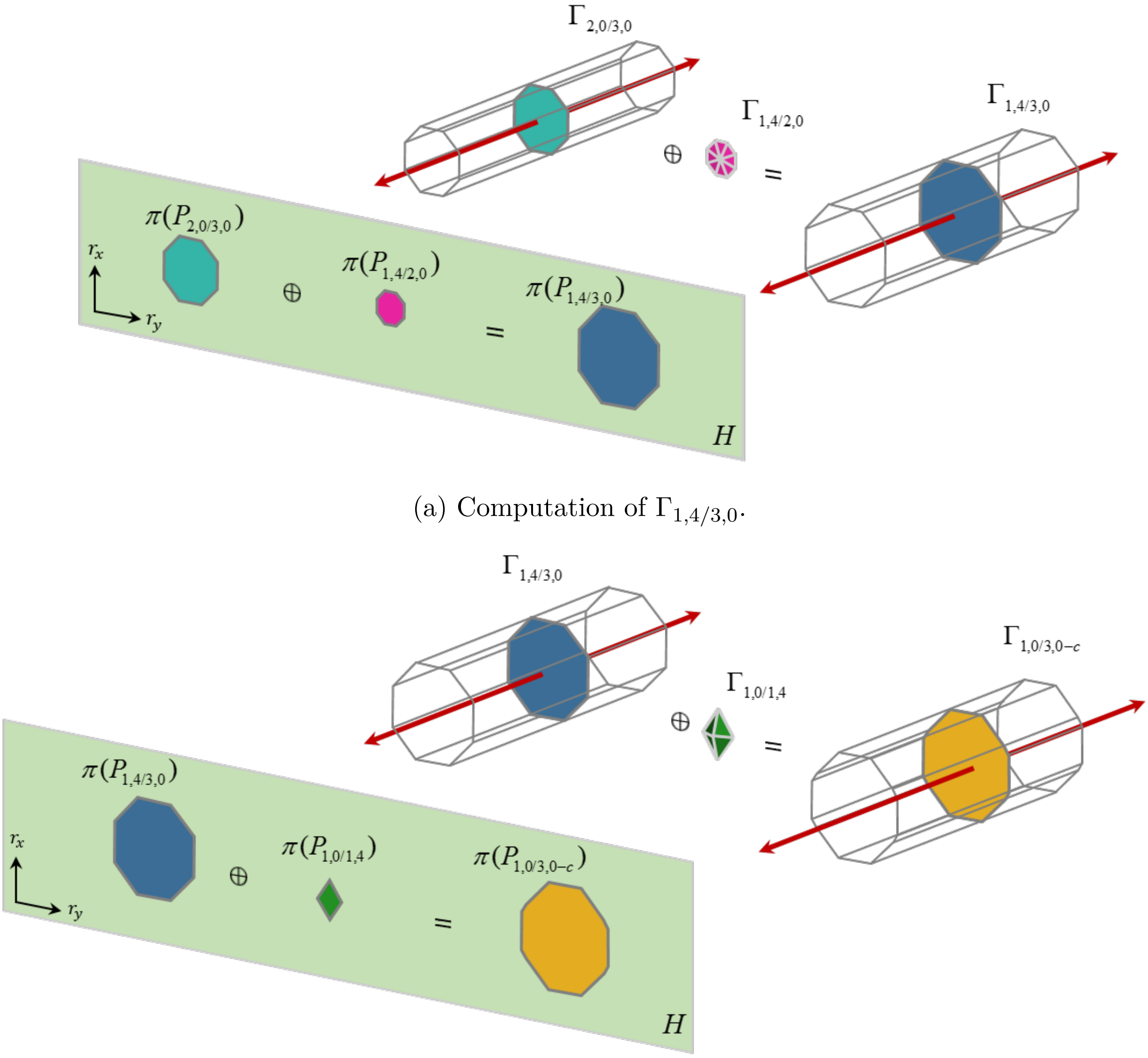
The subspace H corresponds in this case to the space spanned by the rows of the wrench-matrix $\mathcal{W}_{1,0/3,0-c}$ (Eq. (4.26)). The illustration of the projection and addition of the operands are presented in Figure 4.13.

The summary of the whole simulation following the method based on decomposed polyhedra is presented in Table 4.5.

4.6.3 Analysis of results

When applying this strategy to the whole simulation, the computational time could be reduced significantly in comparison with the other two strategies. Regarding the solution based on 6D capped polytopes, a reduction of 99,997% was achieved; and regarding the solution based on caps removal, the reduction was of 99,539%. Table 4.6 presents a summary of the three simulations.

This reduction is justified by the fact that only those meaningful facets in the associated tolerancing problem were calculated. The kinematic analysis performed during the computation acts as a sort of filter, to determine for each operation, the required set of variables to completely define the relative position of the related features.



(a) Computation of $\Gamma_{1,4/3,0}$.

(b) Computation of $\Gamma_{1,0/3,0-c}$.

Figure 4.13: 3D representation of the computation of $\Gamma_{1,0/3,0-c}$ by the projection-based method.

Table 4.5: Summary of the simulation with projection-based sums (HS: half-space).

Operation	Dim	DOFs	Cap HS	Non-cap HS	Vertices	Time [s]
$\Gamma'_{1,0/3,0-ab} = \Gamma'_{1,0/3,0-a} \cap \Gamma'_{1,0/3,0-b}$	6	3	6	14	128	0,02
$P_{1,4/3,0} = \pi(P_{2,0/3,0}) \tilde{+} \pi(P_{1,4/2,0})$	2	4	0	8	8	0,001
$P_{1,0/3,0-c} = \pi(P'_{1,4/3,0}) \tilde{+} \pi(P_{1,0/1,4})$	2	4	0	12	12	0,001
$\Gamma'_{1,0/3,0} = \Gamma'_{1,0/3,0-ab} \cap \Gamma'_{1,0/3,0-c}$	6	1	2	26	384	0,05
$\Gamma'_{3,0/4,0-ab} = \Gamma'_{3,0/4,0-a} \cap \Gamma'_{3,0/4,0-b}$	6	1	2	28	288	0,05
$\Gamma'_{3,0/4,0} = \Gamma'_{3,0/4,0-ab} \cap \Gamma'_{3,0/4,0-c}$	6	0	0	36	504	0,11
$\Gamma'_{4,0/6,0-ab} = \Gamma'_{4,0/6,0-a} \cap \Gamma'_{4,0/6,0-b}$	6	3	6	14	128	0,03
$P_{4,0/6,4} = \pi(P_{4,0/5,0}) \tilde{+} \pi(P_{5,0/6,4})$	2	4	0	8	8	0,001
$P_{4,0/6,0-c} = \pi(P_{4,0/6,4}) \tilde{+} \pi(P_{6,4/6,0})$	2	4	0	12	12	0,001
$\Gamma'_{4,0/6,0} = \Gamma'_{4,0/6,0-ab} \cap \Gamma'_{4,0/6,0-c}$	6	1	2	26	384	0,05
$P_{1,1/3,0} = \pi(P_{1,1/6,1}) \tilde{+} \pi(P_{1,0/3,0})$	2	4	0	12	12	0,001
$P_{1,1/4,0} = \pi(P_{1,1/3,0}) \tilde{+} \pi(P_{3,0/4,0})$	2	4	0	28	28	0,001
$P_R = P_{1,1/6,1} = \pi(P_{1,1/4,0}) \tilde{+} \pi(P_{4,0/6,0})$	2	4	0	28	28	0,002

Computations performed with the library politopix with an Intel Core i7-3740QM.

Table 4.6: Summary of the simulations.

	Γ'_R	Γ''_R	P_R
Method	Caps	Caps removal	Projection-based
Dim	6	6	2
DOFs	4	4	4
Caps HS	64 400	8	0
Non-cap HS	28	28	28
Vertices	108 860	448	28
Time [s]	11 133	69	0,32

It is worth mentioning that the same result was obtained by the three methods. The difference is the additional (and unnecessary) information calculated by each one of them. The equality of the results was verified projecting the polyhedra to space of bounded displacements (the space spanned by $[r_x, r_y]$ in this case), as illustrated in Figure 4.14. By doing this, the influence of the caps facets is completely avoided in the comparison test. The $\mathcal{HV}$-description of the final calculated polytope is presented in Annex B.2.

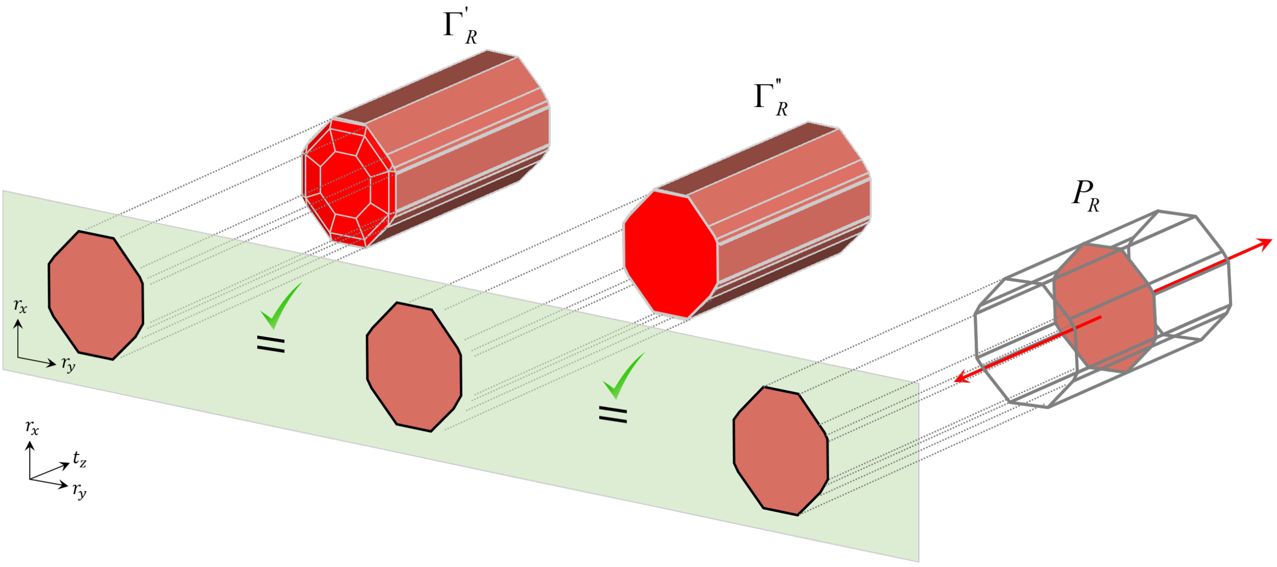


Figure 4.14: Comparison of the simulation results following the three methods.

The fact of obtaining the same results from three different methods, allowed us to conclude that they work correctly. The accuracy of the simulation in quantitative terms does not depend on the choice of the strategy to treat the DOFs, it depends on the initial discretization of the nominal surfaces used to define the sets of constraints. In terms of efficiency, the method based on projections is the most interesting due to the space reduction it proposes and the computational resources it demands.

4.7 Conclusions

In this chapter, we introduced a new model to represent sets of geometric constraints. The concept of prismatic polyhedra was presented in the context of tolerance analysis. We showed that prismatic polyhedra can be decomposed into the sum of polytopes (the bounded part of the polyhedron) and polyhedral cones (the unbounded part of the polyhedron). This decomposition can be conducted according to the mobility conditions of the toleranced features, using the theory of screws.

Advantages of representing geometric constraints as prismatic polyhedra were discussed. When performing a sum of polyhedra, for example, it is possible to exploit the properties of the model by taking only the bounded parts to calculate the sum. This new way of operating polyhedra, which is based on a reduction of the space dimension, was detailed in an algorithm. We demonstrated that reducing the space to calculate sums of sets of constraints avoids calculating fictitious displacements. This implies a significant reduction in the model's complexity and therefore in computational time.

By using prismatic polyhedra, some special cases of the ISO geometrical specifications can be addressed. We presented and discussed how some orientation specifications and the case of features used as datum can be correctly treated during tolerance simulations with sets of constraints.

The way to proceed and the advantages of applying the method were shown in a case study. Compared with the results from the methods described in Chapters 2 and 3, the method based on prismatic polyhedra is significantly more efficient.

Conclusions and future prospects

Conclusions

Due to inevitable geometric deviations, tolerance management is a key aspect during a product's life cycle. This is why many studies can be found in the literature addressing this issue.

We found that existing approaches usually consist of a tolerancing model and a solution strategy. First, the model serves to simulate geometric variations at the part and assembly levels. Assumptions are usually made when defining the model because of the difficulty of considering all real situations that influence products' geometric variations. Second, the solution strategy determines how deviation limits are considered (according to a probabilistic distribution or a worst-case scenario) and how to solve the problem (tolerance analysis or tolerance synthesis).

Among the tolerancing approaches, those based on the manipulation of sets of geometric constraints have the advantage of being robust enough to treat over-constrained assemblies. These methods represent geometric constraints derived from the tolerance zones as sets of algebraic constraints in a 6D abstract space. Stack-ups of geometric variations along a tolerance chain are proposed to be computed by summing and intersecting these operand sets. Using this approach, however, the unbounded displacements linked to the DOF of joints generate unbounded sets of constraints in 6D, which are difficult to deal with from the algorithmic and computational perspective. In order to address this issue, we asked how to deal with the DOFs of the joints in tolerance simulations with sets of constraints.

A solution to this question was proposed by Homri (2014). His proposal, summarized in section 2.5, consists in setting fictitious limits to each free displacement of each joint to obtain bounded 6D sets. These limits are represented by additional constraints, called cap constraints or cap facets.

This solution may avoid the problem of the manipulation of unbounded sets during tolerance stack-up computations, but it nevertheless involves simulating much more complex kinematic chains. We have found that when performing tolerance simulations following this method a great deal of unnecessary information is computed (new cap constraints). We could conclude that these unnecessary data come from propagation along the kinematic chain of the fictitious bounds introduced to the joints. For each simulation, stack-ups of displacements in all possible directions are computed, even if some of them are originally unbounded and consequently unrelated to the associated problem. This implies that when a designer uses this method for computing tolerance propagation, he/she must analyse the resulting data and be able to differentiate real displacement limits from fictitious ones (new cap constraints), which could lead to misinterpretations.

In short, the caps method overcomes the problem of manipulating unbounded sets, but with a high computational cost. This explains the results obtained and the time required to simulate the case study: a calculated polytope made up of 64400 facets takes more than 3 hours to calculate. By means of graphical analysis of the results, we realized that most of these facets derived from cap constraints, and were therefore meaningless for the simulation.

From these results, we concluded that management of the mechanism mobility is a crucial aspect in tolerance simulations with sets of constraints. The way the mobility of joints is treated impacts on the definition of the tolerancing model and the computational efficiency of the general approach.

Based on this conclusion, we proposed two methods for modelling the propagation of geometric deviations with sets of constraints. The aim of the first was to improve the caps method by controlling the propagation of fictitious bounds (or cap facets) during the simulations, as detailed in Chapter 3. To do this, we defined propagation rules to automatically differentiate the real displacement limits from the fictitious ones (those derived from caps) after each operation. We were thus able:

- after each sum, to restore a minimum set of cap facets (a couple of min-max limits for each unbounded displacement) to interrupt their successive propagation along the kinematic chain,
- during intersections, to ensure that the fictitious limits did not hide the real ones,
- during inclusion tests, to avoid misinterpretations over the satisfaction of a functional requirement.

This new strategy, which we called the caps removal method, was tested on a case study. The results showed a significant reduction in unnecessary information calculated and consequently an improvement in computational efficiency. The validity of the new method was verified by comparing the results against those of the initial one.

In short, the caps removal method reduced the complexity of tolerance simulations with sets of constraints. However, as the method simulates all joints with zero DOF, intermediate fictitious displacements are still being calculated, even if they are systematically suppressed after each operation.

The second method we formalized, presented in Chapter 4, consider the internal mobility of the joints along the tolerance chain to avoid calculating fictitious displacements. More precisely, we proposed to determine systematically the parameters required to define the relative position of the surfaces involved. The number of parameters required to control the relative position of two surfaces is usually smaller than that required to define separately the position of the surfaces regarding a reference system. Therefore, the complete set of six parameters is rarely required. This led us to the conclusion that the dimension of the deviation space can be reduced when summing sets of geometric constraints.

We noticed that the same results could be obtained when summing constraints in 6D and then projecting the result to the subspace of the bounded displacements, as when projecting the operands to that subspace and then computing the sum. We proved this property for the general case, by means of a theorem (Theorem 4.3.1). We did this by introducing the concept of prismatic polyhedra in geometric tolerancing.

We found that performing kinematic analysis simultaneously with the tolerance analysis allowed us to determine the subspace in which the limits of the bounded displacements can be manipulated. This space reduction avoids not only the manipulation of unbounded sets of constraints, but also the calculation of fictitious bounds. The projection of the operands can be linked to the phenomenon of displacement absorption, which happens in reality with mechanisms.

We explored other advantages of representing sets of geometric constraints as prismatic polyhedra. For example, some special cases of the ISO geometrical specifications can be addressed. We presented and discussed how some orientation specifications and the case of features used as datum can be correctly treated in tolerance simulations with sets of constraints.

In short, by using the second method, which we called the projection-based method, the complexity of tolerance simulations with sets of constraints could be reduced even further. However, it requires an algorithm to project sets in $\mathbb{R}^n$. Additionally, a special case of a set of constraints poses problems when treated with this method, i.e. the case of unilateral contacts. In these cases, the set of geometric constraints is not centrally symmetrical and cannot be decomposed according to the way the projection-based method proposes.

The case of unilateral contacts can be dealt with perfectly well by means of the method based on caps. We therefore conclude that the methods described in this study are not exclusive; on the contrary, they are complementary.

When comparing the three methods addressed in this thesis, we concluded that, in terms of efficiency, the one based on projections is by far the most efficient since it allows manipulation of only the constraints we are interested in.

As the results obtained from the three different methods in the case study were equal, this indicates the accuracy of the methods in a qualitative sense. The accuracy of the simulation in quantitative terms does not depend on the choice of strategy used to treat the DOFs, it actually depends on the initial discretization of the nominal surfaces originally used to define the sets of constraints.

Future prospects

This study opens up many possibilities for further research. As shown in Figure 4.15, we propose four main specific points:

- The methods formalized in this study have been partially implemented in the computer-aided tolerancing software PolitoCAT, developed at the I2M laboratory. Therefore, the next step is to finish the implementation and continue testing the methods, simulating more case studies. This implementation implies the development of algorithms to project n-dimensional polytopes in $\mathcal{HV}$-description. Additionally, algorithms for sweeping polytopes are also required.

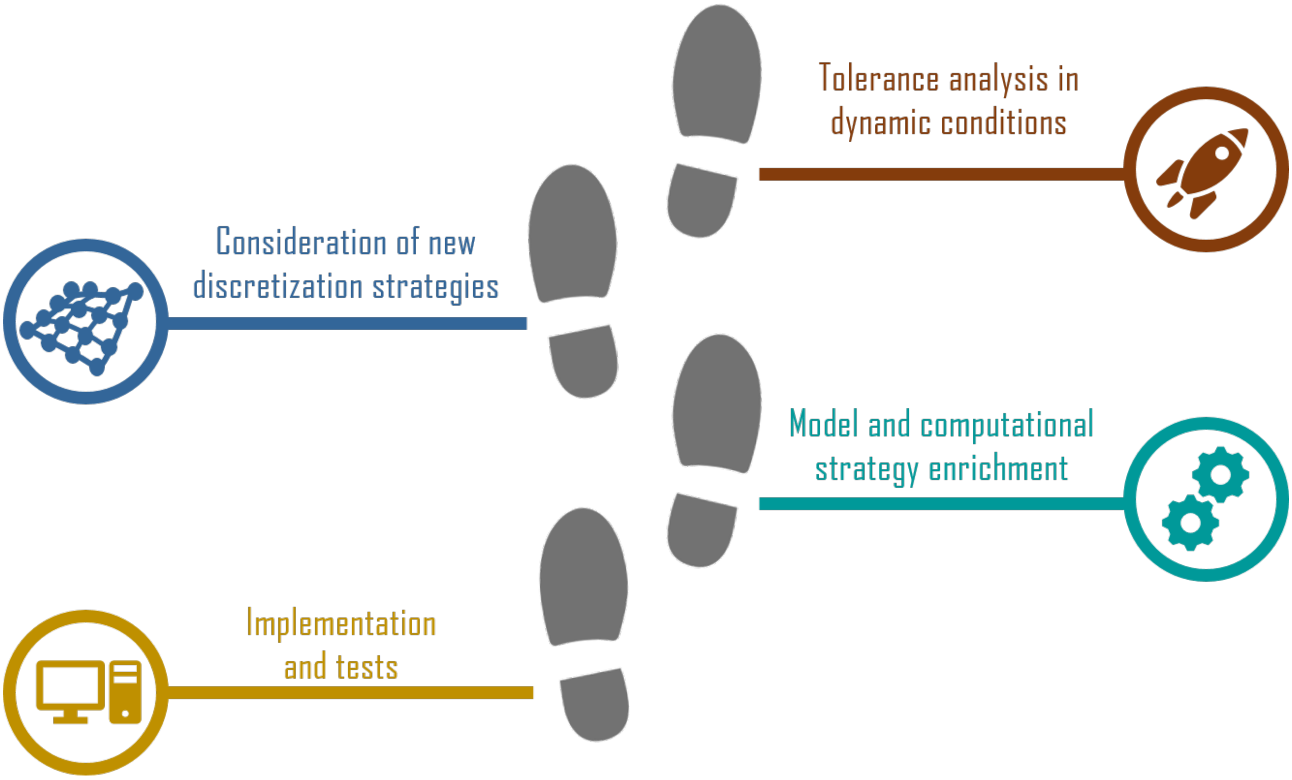

Tolerance analysis in dynamic conditions
Consideration of new discretization strategies
Model and computational strategy enrichment
Implementation and tests

Figure 4.15: Future work.

- As we stated at the beginning of this study, several assumptions were made regarding the physical phenomena considered when defining the model. Future work is required to enrich the model as far as possible, taking form defects, flexible parts and dynamic interaction between parts into consideration. At the I2M laboratory, tolerance analysis considering flexible assemblies started to be studied by Gouyou et al. (2017) and form defects by Yan and Ballu (2017). Regarding to the computational strategy, future work is also required to explore its advantages for the statistical treatment of tolerances as well as for tolerance synthesis.

- The way the nominal features are discretized has a high impact on the complexity of tolerance simulations with sets of constraints. Therefore, further research is required to study different discretization strategies and measure their impact in terms of accuracy and efficiency.

- More advantages of performing kinematic and tolerance analysis simultaneously can be explored. It would be interesting, for example:
  - to explore its advantages when computing intersections of sets of constraints,
  - to determine the level of influence of each toleranced feature or joint regarding each functional condition,
  - to determine the displacements of each joint that really influence the functional conditions of the assembly. This could lead to a strategy for the synthesis of manufacturing specifications.

Bibliography

J. Adams and D. Whitney. Application of screw theory to constraint analysis of assemblies of rigid parts. In *Assembly and Task Planning, 1999. (ISATP '99) Proceedings of the 1999 IEEE International Symposium on*, pages 69–74, 1999. doi: 10.1109/ISATP.1999.782937.

J. D. Adams. Feature based analysis of selective limited motion in assemblies. Master's thesis, Massachusetts Institute of Technology, 1998.

P.-A. Adragna, S. Samper, and H. Favreliere. *How Form Errors Impact on 2D Precision Assembly with Clearance?*, pages 50–59. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-11598-1. doi: 10.1007/978-3-642-11598-1_6. URL http://dx.doi.org/10.1007/978-3-642-11598-1_6.

G. Ameta, J. K. Davidson, and J. J. Shah. The effects of different specifications on the tolerance-maps for an angled face. In *ASME 2004 IDETC-CIE conference*, pages 303–311. American Society of Mechanical Engineers, 2004.

G. Ameta, S. Samper, 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.

B. Anselmetti. Generation of functional tolerancing based on positioning features. *Computer-Aided Design*, 38(8):902 – 919, 2006. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2006.05.005. URL http://www.sciencedirect.com/science/article/pii/S0010448506000959.

B. Anselmetti. *CLIC: A Method for Geometrical Specification of Products*, pages 207–239. John Wiley & Sons, Inc., 2013. ISBN 9781118587027. doi: 10.1002/9781118587027.ch9. URL http://dx.doi.org/10.1002/9781118587027.ch9.

B. Anselmetti, R. Chavanne, J.-X. Yang, and N. Anwer. Quick gps: A new {CAT} system for single-part tolerancing. *Computer-Aided Design*, 42(9):768 – 780, 2010. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2010.04.006. URL http://www.sciencedirect.com/science/article/pii/S0010448510000795.

N. Anwer, A. Ballu, and L. Mathieu. The skin model, a comprehensive geometric model for engineering design. *{CIRP} Annals - Manufacturing Technology*, 62(1):143 – 146, 2013. ISSN 0007-8506. doi: http://dx.doi.org/10.1016/j.cirp.2013.03.078. URL http://www.sciencedirect.com/science/article/pii/S0007850613000796.

S. Arroyave-Tobón, D. Teissandier, and V. Delos. Présentation première année de thèse. In *Séminaire IMC-I2M*, Bordeaux, France, 2015a.

S. Arroyave-Tobón, D. Teissandier, and V. Delos. politopix and PolitoCAT : Open source software for tolerance analysis of over-constrained mechanisms. In *1st Seminar of the European Group of Research in Tolerancing (E-GRT)*, Erlangen, Germany, May 2015b.

S. Arroyave-Tobón, D. Teissandier, and V. Delos. PolitoCAT : open source software for tolerance analysis. In *International Virtual Concept Workshop on INDUSTRIE 4.0*, San Sebastian, Spain, Nov. 2015c.

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.

S. Arroyave-Tobón, D. Teissandier, and V. Delos. Tolerance analysis with adaptive polytopes. In *Journée de l'école doctoral SPI, Université de Bordeaux*, Bordeaux, France, 2016a.

S. Arroyave-Tobón, D. Teissandier, and V. Delos. Présentation deuxième année de thèse. In *Séminaire IMC-I2M*, Bordeaux, France, 2016b.

S. Arroyave-Tobón, D. Teissandier, and V. Delos. Adapting polytopes dimension for managing degrees of freedom in tolerancing analysis. In *14th CIRP Conference on Computer Aided Tolerancing (CAT)*, Gothenburg, Sweden, May 2016c. doi: 10.1016/j.procir.2016.01.020. URL `https://hal.archives-ouvertes.fr/hal-01332381`.

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, 2017a. doi: 10.1016/j.mechmachtheory.2017.02.004. URL `https://hal.archives-ouvertes.fr/hal-01485338`.

S. Arroyave-Tobón, D. Teissandier, and V. Delos. Présentation troisième année de thèse. In *Séminaire IMC-I2M*, Bordeaux, France, 2017b.

S. Arroyave-Tobón, D. Teissandier, and V. Delos. Tolerance analysis with polytopes in HV-description . *Journal of Computing and Information Science in Engineering*, 17, 2017c. doi: 10.1115/1.4036558. URL `https://hal.archives-ouvertes.fr/hal-01518530`.

S. Arroyave-Tobón, D. Teissandier, and V. Delos. Models for representing and summing sets of geometric constraints. In *2st Seminar of the European Group of Research in Tolerancing (E-GRT)*, Metz, France, 2017d.

ASME-Y14.5. *Dimensioning and Tolerancing*. 2009.

R. Ball. *A Treatise on the Theory of Screws*. Cambridge University Press, Cambridge, UK, 1900.

E. Ballot, P. Bourdet, and F. Thiébaut. *Determination of relative situations of parts for tolerance computation*, pages 63–72. Springer Netherlands, Dordrecht, 2003. ISBN 978-94-017-1691-8. doi: 10.1007/978-94-017-1691-8_7. URL `http://dx.doi.org/10.1007/978-94-017-1691-8_7`.

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. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2013.06.011. URL `http://www.sciencedirect.com/science/article/pii/S001044851300119X`.

S. Benichou and B. Anselmetti. Thermal dilatation in functional tolerancing. *Mechanism and Machine Theory*, 46(11):1575 – 1587, 2011. ISSN 0094-114X. doi: http://doi.org/10.1016/j.mechmachtheory.2011.06.009. URL http://www.sciencedirect.com/science/article/pii/S0094114X11001200.

P. Bourdet and É. Ballot. *Geometrical Behavior Laws for Computer-aided Tolerancing*, pages 119–131. Springer Netherlands, Dordrecht, 1996. ISBN 978-94-009-1529-9. doi: 10.1007/978-94-009-1529-9_8. URL http://dx.doi.org/10.1007/978-94-009-1529-9_8.

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.

P. Bourdet, F. Thiébaut, and G. Cid. *Writing the 3D Chain of Dimensions (Tolerance Stack-Up) in Symbolic Expressions*, pages 123–149. John Wiley & Sons, Inc., 2013. ISBN 9781118587027. doi: 10.1002/9781118587027.ch6. URL http://dx.doi.org/10.1002/9781118587027.ch6.

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

F. Charpentier. *Handbook for the geometrical specification of products. The ISO-GPS standard.* Réseau Canopé, 2012. ISBN 978-2-240-03385-7.

K. W. Chase and A. R. Parkinson. A survey of research in the application of tolerance analysis to the design of mechanical assemblies. *Research in Engineering Design*, 3(1):23–37, 1991. ISSN 1435-6066. doi: 10.1007/BF01580066. URL http://dx.doi.org/10.1007/BF01580066.

K. W. Chase, J. Gao, and S. P. Magleby. General 2-d tolerance analysis of mechanical assemblies with small kinematic adjustments. *Journal of Design and Manufacturing*, 5:263–274, 1995.

H. Chen, S. Jin, Z. Li, and X. Lai. A comprehensive study of three dimensional tolerance analysis methods. *Computer-Aided Design*, 53:1 – 13, 2014. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2014.02.014. URL http://www.sciencedirect.com/science/article/pii/S0010448514000475.

W. Cheney and A. A. Goldstein. Proximity maps for convex sets. *Proceedings of the American Mathematical Society*, 10(3):448–450, 1959. ISSN 00029939, 10886826. URL http://www.jstor.org/stable/2032864.

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. ISBN 978-1-4615-5797-5. doi: 10.1007/978-1-4615-5797-5_9. URL http://dx.doi.org/10.1007/978-1-4615-5797-5_9.

A. Clément, A. Rivière, and M. Temmerman. *Cotation tridimensionnelle des systèmes mécaniques.* PYC, 1998. ISBN 978-2-85330-132-9.

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

A. Corrado, W. Polini, and G. Moroni. Manufacturing signature and operating conditions in a variational model for tolerance analysis of rigid assemblies. *Research in Engineering Design*, pages 1–16, 2017. ISSN 1435-6066. doi: 10.1007/s00163-017-0250-y. URL `http://dx.doi.org/10.1007/s00163-017-0250-y`.

H. Dai. *Robust multi-contact dynamical motion planning using contact wrench set.* PhD thesis, Massachusetts Institute of Technology, 2016.

J.-Y. Dantan. *Synthèse des spécifications géométriques: modélisation par Calibre à Mobilités Internes.* PhD thesis, Université de Bordeaux, 2000.

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. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2008.11.003. URL `http://www.sciencedirect.com/science/article/pii/S0010448508002078`.

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

J. K. Davidson and J. J. Shah. Geometric tolerances: A new application for line geometry and screws. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 216(1):95–103, 2002. doi: 10.1243/0954406021524837. URL `http://dx.doi.org/10.1243/0954406021524837`.

V. Delos and D. Teissandier. PolitoCAT and politopix. http://i2m.u-bordeaux.fr/politopix.html, 2015a.

V. Delos and D. Teissandier. Minkowski sum of HV-polytopes in Rn. In *4th Annual International Conference on Computational Mathematics, Computational Geometry and Statistics*, Singapore, Singapore, Jan. 2015b. URL `https://hal.archives-ouvertes.fr/hal-01092060`.

V. Delos and D. Teissandier. Minkowski sum of polytopes defined by their vertices. *Journal of Applied Mathematics and Physics (JAMP)*, 3(1):62–67, Jan. 2015c. doi: 10.4236/jamp.2015.31008. URL `https://hal.archives-ouvertes.fr/hal-01092040`.

V. Delos, S. Arroyave-Tobón, and D. Teissandier. Model reduction in geometric tolerancing by polytopes. *Computer-Aided Design*, a. Submitted in January 2017.

V. Delos, D. Teissandier, and S. Arroyave-Tobón. How to trace the significant information in tolerance analysis with polytopes. In *International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing (JCM 2016)*, Catania, Italy, b. URL `https://hal.archives-ouvertes.fr/hal-01373866`.

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. ISSN 1433-3015. doi: 10.1007/BF01748479. URL `http://dx.doi.org/10.1007/BF01748479`.

A. Desrochers and O. Delbart. Determination of part position uncertainty within mechanical assembly using screw parameters. In *Geometric Design Tolerancing: Theories, Standards and Applications*, pages 185–196. Springer US, 1998. ISBN 978-1-4613-7656-9. doi: 10.1007/978-1-4615-5797-5_14. URL `http://dx.doi.org/10.1007/978-1-4615-5797-5_14`.

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.

A. Dumas. *Development of probabilistic methods for the tolerance analysis of overconstrained mechanisms*. Theses, Ecole nationale supérieure d'arts et métiers - ENSAM, Dec. 2014. URL `https://pastel.archives-ouvertes.fr/tel-01177079`.

A. Dumas, J.-Y. Dantan, and N. Gayton. Impact of a behavior model linearization strategy on the tolerance analysis of over-constrained mechanisms. *Computer-Aided Design*, 62:152 – 163, 2015. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2014.11.002. URL `http://www.sciencedirect.com/science/article/pii/S0010448514002395`.

M. Dupac and D. G. Beale. Dynamic analysis of a flexible linkage mechanism with cracks and clearance. *Mechanism and Machine Theory*, 45(12):1909 – 1923, 2010. ISSN 0094-114X. doi: http://dx.doi.org/10.1016/j.mechmachtheory.2010.07.006. URL `http://www.sciencedirect.com/science/article/pii/S0094114X10001345`.

A. Etienne. *Intégration Produit / Process par les concepts d'activités et de caractéristiques clés - Application à l'optimisation de l'allocation des tolérances géométriques*. Theses, Université de Metz, Oct. 2007. URL `https://tel.archives-ouvertes.fr/tel-00224938`.

P. Fanghella. Kinematics of spatial linkages by group algebra: A structure-based approach. *Mechanism and Machine Theory*, 23(3):171–183, 1988. ISSN 0094-114X. doi: 10.1016/0094-114X(88)90102-4. URL `http://www.sciencedirect.com/science/article/pii/0094114X88901024`.

H. Favreliere. *Modal Tolerancing : From metrology to specifications*. Theses, Université de Savoie, Oct. 2009. URL `https://tel.archives-ouvertes.fr/tel-00426927`.

F. Firmani, A. Zibil, S. B. Nokleby, and R. P. Podhorodeski. Wrench capabilities of planar parallel manipulators. part i: Wrench polytopes and performance indices. *Robotica*, 26(6): 791–802, 2008. doi: 10.1017/S0263574708004384.

A. Fleming. *Analysis of geometric tolerances and uncertainties in assemblies of parts*. PhD thesis, University of Edinburgh, 1987.

A. Fleming. Geometric relationships between toleranced features. *Artificial Intelligence*, 37 (1-3):403–412, 1988. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/0004-3702(88)90062-8. URL `http://www.sciencedirect.com/science/article/pii/0004370288900628`.

E. Fogel and D. Halperin. Exact and efficient construction of minkowski sums of convex polyhedra with applications. *Computer-Aided Design*, 39(11):929 – 940, 2007. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2007.05.017. URL `http://www.sciencedirect.com/science/article/pii/S0010448507001492`.

K. Fukuda. From the zonotope construction to the minkowski addition of convex polytopes. *Journal of Symbolic Computation*, 38(4):1261 – 1272, 2004. ISSN 0747-7171. doi: http://dx.doi.org/10.1016/j.jsc.2003.08.007. URL `http://www.sciencedirect.com/science/article/pii/S0747717104000409`.

K. Fukuda and V. Rosta. Combinatorial face enumeration in convex polytopes. *Computational Geometry*, 4(4):191 – 198, 1994. ISSN 0925-7721. doi: http://dx.doi.org/10.1016/0925-7721(94)90017-5. URL `http://www.sciencedirect.com/science/article/pii/0925772194900175`.

J. Gao, K. W. Chase, and S. P. Magleby. Generalized 3-d tolerance analysis of mechanical assemblies with small kinematic adjustments. *IIE Transactions*, 30(4):367–377, 1998. ISSN 1573-9724. doi: 10.1023/A:1007451225222. URL `http://dx.doi.org/10.1023/A:1007451225222`.

K. Geetha, D. Ravindran, M. S. Kumar, and M. N. Islam. Multi-objective optimization for optimum tolerance synthesis with process and machine selection using a genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 67(9):2439–2457, 2013. ISSN 1433-3015. doi: 10.1007/s00170-012-4662-6. URL `http://dx.doi.org/10.1007/s00170-012-4662-6`.

S. Gerbino and F. Arrichiello. How to investigate constraints and motions in assemblies by screw theory. In *CIRP ICME'04*, volume 4, pages 331–336, 2004.

W. Ghie, L. Laperrière, and A. Desrochers. Statistical tolerance analysis using the unified jacobian-torsor model. *International Journal of Production Research*, 48(15):4609–4630, 2010. doi: 10.1080/00207540902824982. URL `http://dx.doi.org/10.1080/00207540902824982`.

M. Giordano, D. Duret, S. Tichadou, and R. Arrieux. Clearance space in volumic dimensioning. *CIRP Annals - Manufacturing Technology*, 41(1):565 – 568, 1992. ISSN 0007-8506. doi: http://dx.doi.org/10.1016/S0007-8506(07)61269-4. URL `http://www.sciencedirect.com/science/article/pii/S0007850607612694`.

M. Giordano, S. Samper, and J.-P. Petit. *Tolerance Analysis and Synthesis by Means of Deviation Domains, Axi-Symmetric Cases*, pages 85–94. Springer Netherlands, Dordrecht, 2007. ISBN 978-1-4020-5438-9. doi: 10.1007/1-4020-5438-6_10. URL `http://dx.doi.org/10.1007/1-4020-5438-6_10`.

D. Gouyou, D. Teissandier, and V. Delos. Tolerance analysis by polytopes: Application to assembly interferences diagnosis. *Procedia CIRP*, 43:52 – 57, 2016. ISSN 2212-8271. doi: http://dx.doi.org/10.1016/j.procir.2016.02.019. URL `http://www.sciencedirect.com/science/article/pii/S2212827116002365`.

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*, Apr 2017. ISSN 1435-6066. doi: 10.1007/s00163-017-0256-5. URL `https://doi.org/10.1007/s00163-017-0256-5`.

L. Grandguillaume, S. Lavernhe, and C. Tournier. Choix de l'orientation outil en usinage 5 axes positionnés à partir de polytopes de jerk. In *15e Colloque National AIP-Priméca*, volume 15, pages 1–6, 2017.

J. Hervé. The Lie group of rigid body displacements, a fundamental tool for mechanism design. *Mechanism and Machine Theory*, 34(5):719–730, July 1999. ISSN 0094-114X. doi: 10.1016/S0094-114X(98)00051-2. URL `http://www.sciencedirect.com/science/article/pii/S0094114X98000512`.

J. M. Hervé. Intrinsic formulation of problems of geometry and kinematics of mechanisms. *Mechanism and Machine Theory*, 17(3):179 – 184, 1982. ISSN 0094-114X.

J. M. Hervé. The mathematical group structure of the set of displacements. *Mechanism and Machine Theory*, 29(1):73–81, Jan. 1994. ISSN 0094-114X. doi: 10.1016/0094-114X(94)90021-3. URL http://www.sciencedirect.com/science/article/pii/0094114X94900213.

L. Homri. *Stratégies de mise en oeuvre des polytopes en analyse de tolérance.* PhD thesis, Université de Bordeaux, Nov. 2014. URL https://tel.archives-ouvertes.fr/tel-01133691.

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. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2014.11.005. URL http://www.sciencedirect.com/science/article/pii/S0010448514002425.

L. Homri, E. Goka, G. Levasseur, and J.-Y. Dantan. Tolerance analysis — form defects modeling and simulation by modal decomposition and optimization. *Computer-Aided Design*, 91:46 – 59, 2017. ISSN 0010-4485. doi: https://doi.org/10.1016/j.cad.2017.04.007. URL http://www.sciencedirect.com/science/article/pii/S0010448517300623.

C. Huang, K. Sugimoto, and I. Parkin. The correspondence between finite screw systems and projective spaces. *Mechanism and Machine Theory*, 43(1):50 – 56, 2008. ISSN 0094-114X. doi: http://dx.doi.org/10.1016/j.mechmachtheory.2007.01.005. URL http://www.sciencedirect.com/science/article/pii/S0094114X07000225.

M. Inui and A. Ohta. Using a gpu to accelerate die and mold fabrication. *IEEE Computer Graphics and Applications*, 27:82–88, 2007. ISSN 0272-1716. doi: doi.ieeecomputersociety.org/10.1109/MCG.2007.23.

ISO-1101. *Geometrical Product Specification (GPS) - Geometrical tolerancing - Tolerances of form, orientation, location and run-out.* 2012.

ISO-14405-1. *Geometrical Product Specification (GPS) - Geometrical tolerancing - Linear sizes.* 2011.

ISO-14405-2. *Geometrical Product Specification (GPS) - Geometrical tolerancing - Dimensions other linear sizes.* 2011.

ISO-3952-1. *Kinematic diagrams - Graphical symbols, Part 1.* 1981.

ISO-5459. *Geometrical Product Specification (GPS) - Geometrical tolerancing - Datums and datum systems.* 2011.

ISO-8015. *Geometrical Product Specification (GPS) - Geometrical tolerancing - Concepts, principles and rules.* 2011.

A. Jeang. Tolerance design: Choosing optimal tolerance specifications in the design of machined parts. *Quality and Reliability Engineering International*, 10(1):27–35, 1994. ISSN 1099-1638. doi: 10.1002/qre.4680100107. URL http://dx.doi.org/10.1002/qre.4680100107.

A. D. Jian, G. Ameta, J. K. Davidson, and J. J. Shah. *Tolerance Analysis and Allocation using Tolerance-Maps for a Power Saw Assembly*, pages 267–276. Springer Netherlands, Dordrecht, 2007. ISBN 978-1-4020-5438-9. doi: 10.1007/1-4020-5438-6_27. URL http://dx.doi.org/10.1007/1-4020-5438-6_27.

R. Konkar and M. Cutkosky. Incremental Kinematic Analysis of Mechanisms. *Journal of Mechanical Design*, 117(4):589–596, Dec. 1995. ISSN 1050-0472. doi: 10.1115/1.2826724. URL http://dx.doi.org/10.1115/1.2826724.

U. Kumaraswamy, M. Shunmugam, and S. Sujatha. A unified framework for tolerance analysis of planar and spatial mechanisms using screw theory. *Mechanism and Machine Theory*, 69:168 – 184, 2013. ISSN 0094-114X. doi: http://dx.doi.org/10.1016/j.mechmachtheory.2013.06.001. URL http://www.sciencedirect.com/science/article/pii/S0094114X13001146.

L. Laperrière and H. ElMaraghy. Tolerance analysis and synthesis using jacobian transforms. *CIRP Annals - Manufacturing Technology*, 49(1):359 – 362, 2000. ISSN 0007-8506. doi: http://dx.doi.org/10.1016/S0007-8506(07)62964-3. URL http://www.sciencedirect.com/science/article/pii/S0007850607629643.

Y. Ledoux and D. Teissandier. Tolerance analysis of a product coupling geometric and architectural specifications in a probabilistic approach. *Research in Engineering Design*, 24(3):297–311, 2013. ISSN 1435-6066. doi: 10.1007/s00163-012-0146-9. URL http://dx.doi.org/10.1007/s00163-012-0146-9.

Q. Li, Z. Huang, and J. M. Hervé. Displacement manifold method for type synthesis of lower-mobility parallel mechanisms. *Science in China Series E: Technological Sciences*, 47(6): 641–650, 2004. ISSN 1006-9321. doi: 10.1360/03ye0352. URL http://dx.doi.org/10.1360/03ye0352.

W. Li and S. McMains. A sweep and translate algorithm for computing voxelized 3d minkowski sums on the GPU. *Computer-Aided Design*, 46:90 – 100, 2014. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2013.08.021. URL http://www.sciencedirect.com/science/article/pii/S0010448513001619. 2013 {SIAM} Conference on Geometric and Physical Modeling.

J.-M. Lien. *A Simple Method for Computing Minkowski Sum Boundary in 3D Using Collision Detection*, pages 401–415. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-00312-7. doi: 10.1007/978-3-642-00312-7_25. URL http://dx.doi.org/10.1007/978-3-642-00312-7_25.

B. Lindau, S. Lorin, L. Lindkvist, and R. Söderberg. Efficient contact modeling in nonrigid variation simulation. *Journal of Computing and Information Science in Engineering*, 16(1): 011002, 2016.

L. Lindkvist and R. Söderberg. Computer-aided tolerance chain and stability analysis. *Journal of Engineering Design*, 14(1):17–39, 2003. doi: 10.1080/0954482031000078117. URL http://dx.doi.org/10.1080/0954482031000078117.

S. Lorin, L. Lindkvist, R. Söderberg, and R. Sandboge. Combining variation simulation with thermal expansion simulation for geometry assurance. *Journal of Computing and Information Science in Engineering*, 13(3):031007, 2013.

M. Mansuy. *Three-dimensional tolerancing assistance : domains model.* Theses, Université Grenoble Alpes, June 2012. URL https://tel.archives-ouvertes.fr/tel-00734713.

M. Mansuy, M. Giordano, and P. Hernandez. A new calculation method for the worst case tolerance analysis and synthesis in stack-type assemblies. *Comput. Aided Des.*, 43(9):1118–1125, Sept. 2011. ISSN 0010-4485. doi: 10.1016/j.cad.2011.04.010. URL http://dx.doi.org/10.1016/j.cad.2011.04.010.

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, 2013a.

M. Mansuy, M. Giordano, and P. Hernandez. A generic method for the worst case and statistical tridimensional tolerancing analysis. *Procedia CIRP*, 10:276 – 282, 2013b. ISSN 2212-8271. doi: http://dx.doi.org/10.1016/j.procir.2013.08.042. URL http://www.sciencedirect.com/science/article/pii/S221282711300591X.

M. Mazur, M. Leary, and A. Subic. Computer aided tolerancing (cat) platform for the design of assemblies under external and internal forces. *Computer-Aided Design*, 43(6):707 – 719, 2011. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2011.02.004. URL http://www.sciencedirect.com/science/article/pii/S0010448511000364.

H. Mejbri, B. Anselmetti, and K. Mawussi. Functional tolerancing of complex mechanisms: Identification and specification of key parts. *Computers & Industrial Engineering*, 49(2): 241 – 265, 2005. ISSN 0360-8352. doi: http://dx.doi.org/10.1016/j.cie.2005.04.002. URL http://www.sciencedirect.com/science/article/pii/S0360835205000641.

X. Ming and K. Mak. Intelligent approaches to tolerance allocation and manufacturing operations selection in process planning. *Journal of Materials Processing Technology*, 117(1-2): 75 – 83, 2001. ISSN 0924-0136. doi: http://doi.org/10.1016/S0924-0136(01)01154-2. URL http://www.sciencedirect.com/science/article/pii/S0924013601011542.

E. P. Morse. Statistical analysis of assemblies having dependent fitting conditions. In *ASME 2004 International Mechanical Engineering Congress and Exposition*, pages 91–95. American Society of Mechanical Engineers, 2004.

M. Mounaud, F. Thiébaut, P. Bourdet, H. Falgarone, and N. Chevassus. Assembly sequence influence on geometric deviations propagation of compliant parts. *International Journal of Production Research*, 49(4):1021–1043, 2011. doi: 10.1080/00207540903460240. URL http://dx.doi.org/10.1080/00207540903460240.

P. Muthu, V. Dhanalakshmi, and K. Sankaranarayanasamy. Optimal tolerance design of assembly for minimum quality loss and manufacturing cost using metaheuristic algorithms. *The International Journal of Advanced Manufacturing Technology*, 44(11):1154–1164, 2009. ISSN 1433-3015. doi: 10.1007/s00170-009-1930-1. URL http://dx.doi.org/10.1007/s00170-009-1930-1.

S. D. Nigam and J. U. Turner. Review of statistical approaches to tolerance analysis. *Computer-Aided Design*, 27(1):6 – 15, 1995. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/0010-4485(95)90748-5. URL http://www.sciencedirect.com/science/article/pii/0010448595907485.

M. Peternell and T. Steiner. Minkowski sum boundary surfaces of 3d-objects. *Graphical Models*, 69(3-4):180 – 190, 2007. ISSN 1524-0703. doi: http://dx.doi.org/10.1016/j.gmod.2007.01.001. URL http://www.sciencedirect.com/science/article/pii/S1524070307000021.

J.-P. Petit. *Spécification Géométrique des produits: méthode d'analyse de tolérances. Application en conception assistée par ordinateur*. PhD thesis, Université de Savoie, 2004.

J. Pey and F. J. Planes. Direct calculation of elementary flux modes satisfying several biological constraints in genome-scale metabolic networks. *Bioinformatics*, page btu193, 2014.

L. Pierre. *Integration of the thermomechanical behaviour of parts in tolerancing analysis: application to a helicopter turboshaft engine turbine*. PhD thesis, Arts et Métiers ParisTech, May 2011. URL `https://pastel.archives-ouvertes.fr/pastel-00594317`.

L. Pierre and B. Anselmetti. Comparison of analysis line and polytopes methods to determine the result of a tolerance chain. In *13th CIRP International Seminar on Computer Aided Tolerancing*, Hangzhou, China, May 2014. URL `https://hal.archives-ouvertes.fr/hal-01094151`.

U. Prisco and G. Giorleo. Overview of current CAT systems. *Integrated Computer-Aided Engineering*, 9(4):373–387, 2002.

A. J. Qureshi, J.-Y. Dantan, V. Sabri, P. Beaucaire, and N. Gayton. A statistical tolerance analysis approach for over-constrained mechanism based on optimization and monte carlo simulation. *Computer-Aided Design*, 44(2):132 – 142, 2012. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2011.10.004. URL `http://www.sciencedirect.com/science/article/pii/S0010448511002636`.

J. Rico, J. Gallardo, and J. Duffy. Screw theory and higher order kinematic analysis of open serial and closed chains. *Mechanism and Machine Theory*, 34(4):559 – 586, 1999. ISSN 0094-114X. doi: http://dx.doi.org/10.1016/S0094-114X(98)00029-9. URL `http://www.sciencedirect.com/science/article/pii/S0094114X98000299`.

O. Rique Garaizar, L. Qiao, N. ANWER, and L. Mathieu. Integration of thermal effects into Tolerancing using Skin Model Shapes. In *14th CIRP Conference on Computer Aided Tolerancing (CAT)*, Goteborg, Sweden, 2016. doi: 10.1016/j.procir.2016.02.079. URL `https://hal.archives-ouvertes.fr/hal-01363754`.

O. Rouetbi, L. Pierre, B. Anselmetti, and H. Denoix. *ISO Tolerancing of hyperstatic mechanical systems with deformation control*, pages 991–1000. Springer International Publishing, Cham, 2017. ISBN 978-3-319-45781-9. doi: 10.1007/978-3-319-45781-9_99. URL `http://dx.doi.org/10.1007/978-3-319-45781-9_99`.

S. Samper. *Tolerancing and structural analysis in order to study flexible assemblies and form errors*. Habilitation à diriger des recherches (HDR), Université de Savoie, Nov. 2007. URL `https://tel.archives-ouvertes.fr/tel-00359471`.

S. Samper and M. Giordano. *Simultaneous analysis method for tolerancing flexible mechanisms*, pages 127–134. Springer Netherlands, Dordrecht, 2003. ISBN 978-94-017-1691-8. doi: 10.1007/978-94-017-1691-8_13. URL `http://dx.doi.org/10.1007/978-94-017-1691-8_13`.

S. Samper, J.-P. Petit, and M. Giordano. *Computer Aided Tolerancing - Solver and Post Processor Analysis*, pages 487–497. Springer London, London, 2006. ISBN 978-1-84628-210-2. doi: 10.1007/1-84628-210-1_40. URL `http://dx.doi.org/10.1007/1-84628-210-1_40`.

B. Schleich and S. Wartzack. A discrete geometry approach for tolerance analysis of mechanism. *Mechanism and Machine Theory*, 77:148 – 163, 2014. ISSN 0094-114X. doi: http://dx.doi.org/10.1016/j.mechmachtheory.2014.02.013. URL http://www.sciencedirect.com/science/article/pii/S0094114X14000524.

B. Schleich and S. Wartzack. Approaches for the assembly simulation of skin model shapes. *Computer-Aided Design*, 65:18 – 33, 2015. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2015.03.004. URL http://www.sciencedirect.com/science/article/pii/S0010448515000366.

B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack. Contact and Mobility Simulation for Mechanical Assemblies based on Skin Model Shapes. *Journal of Computing and Information Science in Engineering*, page 7, 2015. doi: 10.1115/1.4029051. URL https://hal.archives-ouvertes.fr/hal-01094269.

P. Serré, N. Anwer, and J. Yang. *On the Use of Conformal Geometric Algebra in Geometric Constraint Solving*, pages 217–232. Springer London, London, 2011. ISBN 978-0-85729-811-9. doi: 10.1007/978-0-85729-811-9_11. URL https://doi.org/10.1007/978-0-85729-811-9_11.

J. J. Shah, G. Ameta, Z. Shen, and J. Davidson. Navigating the tolerance analysis maze. *Computer-Aided Design and Applications*, 4(5):705–718, 2007. doi: 10.1080/16864360.2007.10738504. URL http://dx.doi.org/10.1080/16864360.2007.10738504.

Z. Shen, G. Ameta, J. J. Shah, and J. K. Davidson. A comparative study of tolerance analysis methods. *Journal of Computing and Information Science in Engineering*, 5(3):247–256, 2005.

D. K. Smith. Constraint analysis of assemblies using screw theory and tolerance sensitivities. Master's thesis, Brigham Young University. Department of Mechanical Engineering, 2001.

R. Sodhi and J. U. Turner. Relative positioning of variational part models for design analysis. *Computer-Aided Design*, 26(5):366 – 378, 1994. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/0010-4485(94)90024-8. URL http://www.sciencedirect.com/science/article/pii/0010448594900248.

V. Srinivasan. The role of sweeps in tolerancing semantics. *Manufacturing Review*, 6:275–275, 1993.

A. Stricher. *Tolérancement flexible d'assemblages de grandes structures aéronautiques*. PhD thesis, École normale supérieure de Cachan-ENS Cachan, 2013. URL https://tel.archives-ouvertes.fr/tel-00824579.

H. Su and C. Yue. Type synthesis of freedom and constraint elements for design of flexure mechanisms. *Mechanical Sciences*, 4(2):263–277, 2013. doi: 10.5194/ms-4-263-2013. URL http://www.mech-sci.net/4/263/2013/.

D. Teissandier. *Contribution à l'analyse des tolérances géométriques d'un système mécanique par des polytopes*. Habilitation à diriger des recherches, Université Bordeaux 1, Dec. 2012. URL https://hal.archives-ouvertes.fr/tel-01202782.

D. Teissandier and V. Delos. Algorithm to calculate the Minkowski sums of 3-polytopes based on normal fans. *Computer-Aided Design*, 43:1567–1576, 2011. doi: https://doi.org/10.1016/j.cad.2011.06.016. URL http://www.sciencedirect.com/science/article/pii/S0010448511001588.

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. ISBN 0-7923-5654-3.

F. Wandebäck, R. Söderberg, J. S. Carlson, P.-J. Wahlborg, L. Lindkvist, and F. Ekstedt. Variation feedback and 3d visualization of geometrical inspection data. In *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 197–208. American Society of Mechanical Engineers, 2009.

K. Wärmefjord, J. S. Carlson, and R. Söderberg. Controlling geometrical variation caused by assembly fixtures. *Journal of Computing and Information Science in Engineering*, 16(1): 011007, 2016.

C. Weibel. *Minkowski sums of polytopes: combinatorics and computation.* PhD thesis, École Polytechnique Fédérale De Lausanne, 2007.

F. Wu, J.-Y. Dantan, A. Etienne, A. Siadat, and P. Martin. Improved algorithm for tolerance allocation based on monte carlo simulation and discrete optimization. *Computers & Industrial Engineering*, 56(4):1402 – 1413, 2009. ISSN 0360-8352. doi: https://doi.org/10.1016/j.cie.2008.09.005. URL `http://www.sciencedirect.com/science/article/pii/S0360835208001927`.

Y. Wu, J. J. Shah, and J. K. Davidson. Improvements to algorithms for computing the minkowski sum of 3-polytopes. *Computer-Aided Design*, 35(13):1181 – 1192, 2003. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/S0010-4485(03)00023-X. URL `http://www.sciencedirect.com/science/article/pii/S001044850300023X`.

K. Xie, L. Wells, J. A. Camelio, and B. D. Youn. Variation propagation analysis on compliant assemblies considering contact interaction. *Journal of Manufacturing Science and Engineering*, 129(5):934–942, 2007.

X. Yan and A. Ballu. Generation of consistent skin model shape based on fea method. *The International Journal of Advanced Manufacturing Technology*, pages 1–14, 2017. ISSN 1433-3015. doi: 10.1007/s00170-017-0177-5. URL `http://dx.doi.org/10.1007/s00170-017-0177-5`.

J. Yang, J. Wang, Z. Wu, and N. Anwer. Statistical tolerancing based on variation of point-set. *Procedia CIRP*, 10:9 – 16, 2013. ISSN 2212-8271. doi: http://dx.doi.org/10.1016/j.procir.2013.08.006. URL `http://www.sciencedirect.com/science/article/pii/S2212827113005556`.

W. Zeng, Y. Rao, and P. Wang. An effective strategy for improving the precision and computational efficiency of statistical tolerance optimization. *The International Journal of Advanced Manufacturing Technology*, pages 1–12, 2017a. ISSN 1433-3015. doi: 10.1007/s00170-017-0256-7. URL `http://dx.doi.org/10.1007/s00170-017-0256-7`.

W. Zeng, Y. Rao, P. Wang, and W. Yi. A solution of worst-case tolerance analysis for partial parallel chains based on the unified jacobian-torsor model. *Precision Engineering*, 47:276 – 291, 2017b. ISSN 0141-6359. doi: http://doi.org/10.1016/j.precisioneng.2016.09.002. URL `http://www.sciencedirect.com/science/article/pii/S0141635916302069`.

M. Zhang, N. Anwer, L. Mathieu, and H. Zhao. A discrete geometry framework for geometrical product specifications. In *Proceedings of the 21st CIRP Design Conference, Kaist, MK Thompson, ed., Paper*, number 20, 2011.

Z. Zhu, N. Anwer, and L. Mathieu. Deviation modeling and shape transformation in design for additive manufacturing. *Procedia CIRP*, 60:211 – 216, 2017. ISSN 2212-8271. doi: http://dx.doi.org/10.1016/j.procir.2017.01.023. URL `http://www.sciencedirect.com/science/article/pii/S2212827117300240`. Complex Systems Engineering and Development Proceedings of the 27th CIRP Design Conference Cranfield University, UK 10th – 12th May 2017.

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

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. ISSN 0951-8320. doi: http://dx.doi.org/10.1016/j.ress.2014.09.017. URL `http://www.sciencedirect.com/science/article/pii/S0951832014002336`.

Annexes

Annex A

Scientific communications

A.1 Journal articles

- V. Delos, S. Arroyave-Tobón, and D. Teissandier. Model reduction in geometric tolerancing by polytopes. *Computer-Aided Design*, a. Submitted in January 2017
- S. Arroyave-Tobón, D. Teissandier, and V. Delos. Tolerance analysis with polytopes in HV-description . *Journal of Computing and Information Science in Engineering*, 17, 2017c. doi: 10.1115/1.4036558. URL `https://hal.archives-ouvertes.fr/hal-01518530`
- 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, 2017a. doi: 10.1016/j.mechmachtheory.2017.02.004. URL `https://hal.archives-ouvertes.fr/hal-01485338`

A.2 Oral presentations

- S. Arroyave-Tobón, D. Teissandier, and V. Delos. Models for representing and summing sets of geometric constraints. In *2st Seminar of the European Group of Research in Tolerancing (E-GRT)*, Metz, France, 2017d
- S. Arroyave-Tobón, D. Teissandier, and V. Delos. Présentation troisième année de thèse. In *Séminaire IMC-I2M*, Bordeaux, France, 2017b
- V. Delos, D. Teissandier, and S. Arroyave-Tobón. How to trace the significant information in tolerance analysis with polytopes. In *International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing (JCM 2016)*, Catania, Italy, b. URL `https://hal.archives-ouvertes.fr/hal-01373866`
- 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
- S. Arroyave-Tobón, D. Teissandier, and V. Delos. Adapting polytopes dimension for managing degrees of freedom in tolerancing analysis. In *14th CIRP Conference on Computer*

Aided Tolerancing (CAT), Gothenburg, Sweden, May 2016c. doi: 10.1016/j.procir.2016.01.020. URL `https://hal.archives-ouvertes.fr/hal-01332381`

- S. Arroyave-Tobón, D. Teissandier, and V. Delos. Présentation deuxième année de thèse. In *Séminaire IMC-I2M*, Bordeaux, France, 2016b
- S. Arroyave-Tobón, D. Teissandier, and V. Delos. PolitoCAT : open source software for tolerance analysis. In *International Virtual Concept Workshop on INDUSTRIE 4.0*, San Sebastian, Spain, Nov. 2015c
- S. Arroyave-Tobón, D. Teissandier, and V. Delos. politopix and PolitoCAT : Open source software for tolerance analysis of over-constrained mechanisms. In *1st Seminar of the European Group of Research in Tolerancing (E-GRT)*, Erlangen, Germany, May 2015b
- S. Arroyave-Tobón, D. Teissandier, and V. Delos. Présentation première année de thèse. In *Séminaire IMC-I2M*, Bordeaux, France, 2015a

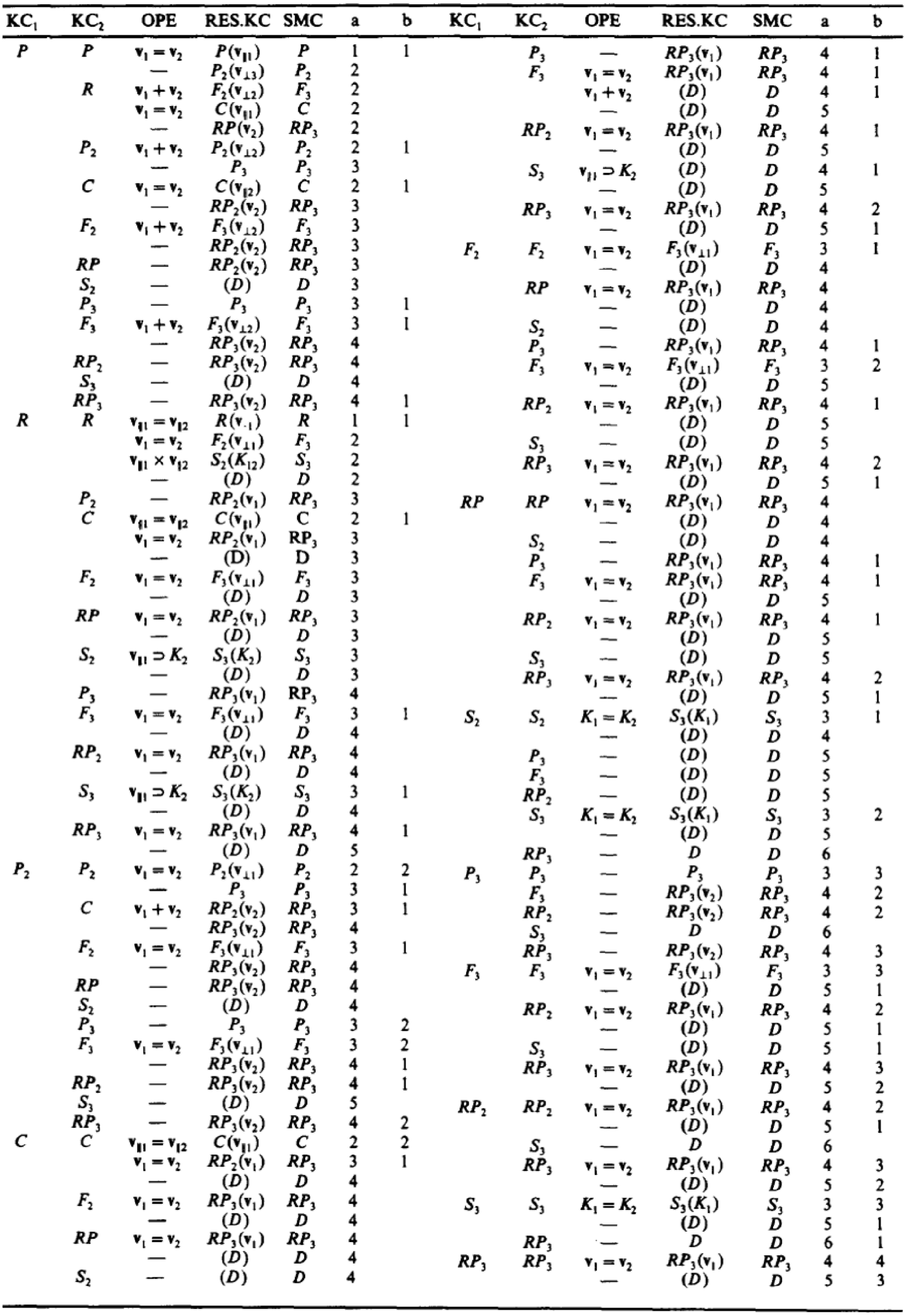
A.3 Posters

- S. Arroyave-Tobón, D. Teissandier, and V. Delos. Tolerance analysis with adaptive polytopes. In *Journée de l'école doctoral SPI, Université de Bordeaux*, Bordeaux, France, 2016a

Annex A

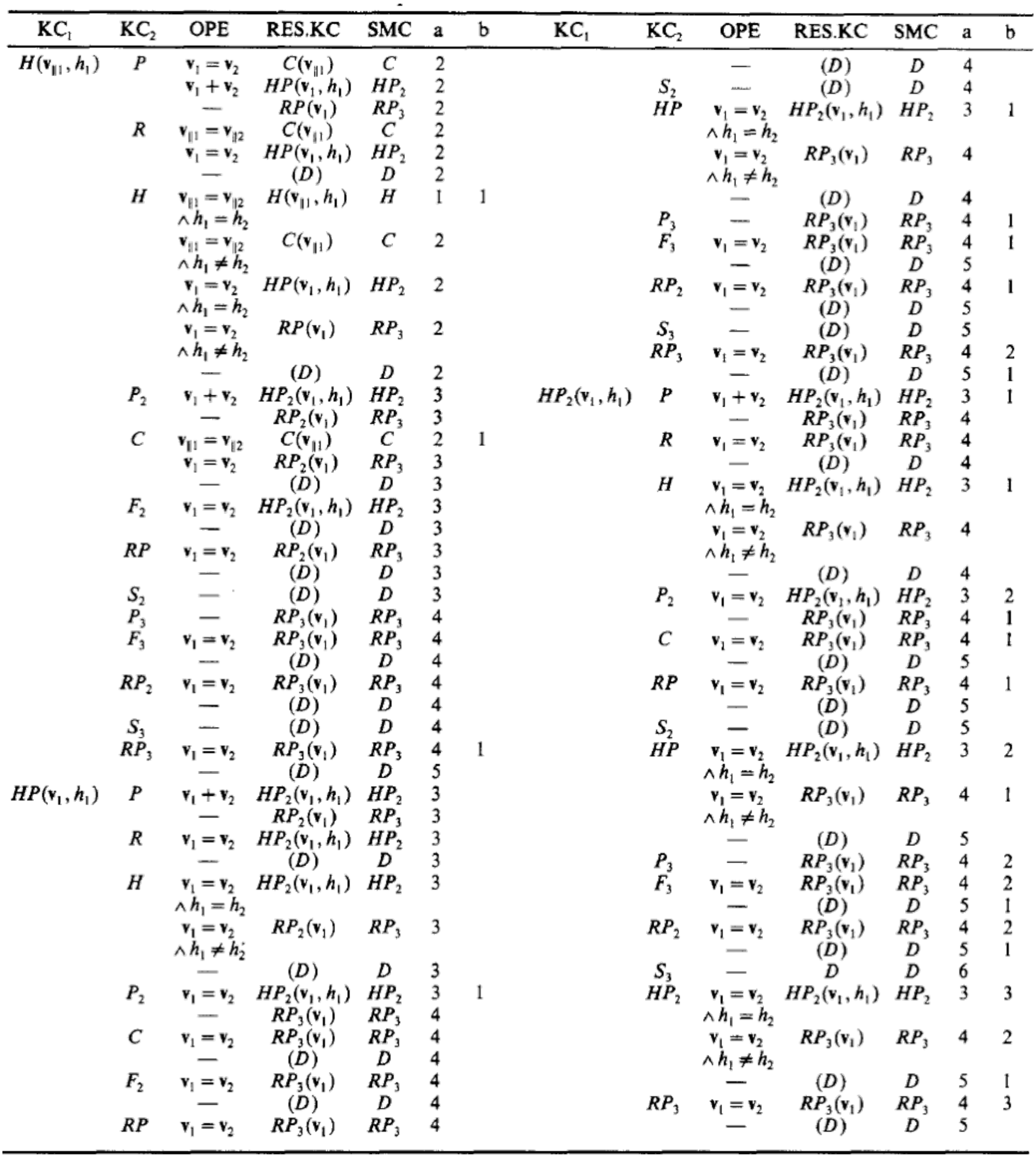
Product of displacements subgroups

A.1 Composition rules for algebraic constraints



KC_1	KC_2	OPE	RES.KC	SMC	a	b
P	P	$\mathbf{v}_1 = \mathbf{v}_2$	$P(\mathbf{v}_{\parallel 1})$	P	1	1
		—	$P_2(\mathbf{v}_{\perp 3})$	P_2	2	
	R	$\mathbf{v}_1 + \mathbf{v}_2$	$F_2(\mathbf{v}_{\perp 2})$	F_3	2	
		$\mathbf{v}_1 = \mathbf{v}_2$	$C(\mathbf{v}_{\parallel 1})$	C	2	
		—	$RP(\mathbf{v}_2)$	RP_3	2	
	P_2	$\mathbf{v}_1 + \mathbf{v}_2$	$P_2(\mathbf{v}_{\perp 2})$	P_2	2	1
		—	P_3	P_3	3	
	C	$\mathbf{v}_1 = \mathbf{v}_2$	$C(\mathbf{v}_{\parallel 2})$	C	2	1
		—	$RP_2(\mathbf{v}_2)$	RP_3	3	
	F_2	$\mathbf{v}_1 + \mathbf{v}_2$	$F_3(\mathbf{v}_{\perp 2})$	F_3	3	
		—	$RP_2(\mathbf{v}_2)$	RP_3	3	
	RP	—	$RP_2(\mathbf{v}_2)$	RP_3	3	
	S_2	—	(D)	D	3	
	P_3	—	P_3	P_3	3	1
	F_3	$\mathbf{v}_1 + \mathbf{v}_2$	$F_3(\mathbf{v}_{\perp 2})$	F_3	3	1
		—	$RP_3(\mathbf{v}_2)$	RP_3	4	
	RP_2	—	$RP_3(\mathbf{v}_2)$	RP_3	4	
	S_3	—	(D)	D	4	
	RP_3	—	$RP_3(\mathbf{v}_2)$	RP_3	4	1
R	R	$\mathbf{v}_{\parallel 1} = \mathbf{v}_{\parallel 2}$	$R(\mathbf{v}_{\cdot 1})$	R	1	1
		$\mathbf{v}_1 = \mathbf{v}_2$	$F_2(\mathbf{v}_{\perp 1})$	F_3	2	
		$\mathbf{v}_{\parallel 1} \times \mathbf{v}_{\parallel 2}$	$S_2(K_{12})$	S_3	2	
		—	(D)	D	2	
	P_2	—	$RP_2(\mathbf{v}_1)$	RP_3	3	
	C	$\mathbf{v}_{\parallel 1} = \mathbf{v}_{\parallel 2}$	$C(\mathbf{v}_{\parallel 1})$	C	2	1
		$\mathbf{v}_1 = \mathbf{v}_2$	$RP_2(\mathbf{v}_1)$	RP_3	3	
		—	(D)	D	3	
	F_2	$\mathbf{v}_1 = \mathbf{v}_2$	$F_3(\mathbf{v}_{\perp 1})$	F_3	3	
		—	(D)	D	3	
	RP	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_2(\mathbf{v}_1)$	RP_3	3	
		—	(D)	D	3	
	S_2	$\mathbf{v}_{\parallel 1} \supset K_2$	$S_3(K_2)$	S_3	3	
		—	(D)	D	3	
	P_3	—	$RP_3(\mathbf{v}_1)$	RP_3	4	
	F_3	$\mathbf{v}_1 = \mathbf{v}_2$	$F_3(\mathbf{v}_{\perp 1})$	F_3	3	1
		—	(D)	D	4	
	RP_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	S_3	$\mathbf{v}_{\parallel 1} \supset K_2$	$S_3(K_2)$	S_3	3	1
		—	(D)	D	4	
	RP_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
P_2	P_2	$\mathbf{v}_1 = \mathbf{v}_2$	$P_2(\mathbf{v}_{\perp 1})$	P_2	2	2
		—	P_3	P_3	3	1
	C	$\mathbf{v}_1 + \mathbf{v}_2$	$RP_2(\mathbf{v}_2)$	RP_3	3	1
		—	$RP_3(\mathbf{v}_2)$	RP_3	4	
	F_2	$\mathbf{v}_1 = \mathbf{v}_2$	$F_3(\mathbf{v}_{\perp 1})$	F_3	3	1
		—	$RP_3(\mathbf{v}_2)$	RP_3	4	
	RP	—	$RP_3(\mathbf{v}_2)$	RP_3	4	
	S_2	—	(D)	D	4	
	P_3	—	P_3	P_3	3	2
	F_3	$\mathbf{v}_1 = \mathbf{v}_2$	$F_3(\mathbf{v}_{\perp 1})$	F_3	3	2
		—	$RP_3(\mathbf{v}_2)$	RP_3	4	1
	RP_2	—	$RP_3(\mathbf{v}_2)$	RP_3	4	1
	S_3	—	(D)	D	5	
	RP_3	—	$RP_3(\mathbf{v}_2)$	RP_3	4	2
C	C	$\mathbf{v}_{\parallel 1} = \mathbf{v}_{\parallel 2}$	$C(\mathbf{v}_{\parallel 1})$	C	2	2
		$\mathbf{v}_1 = \mathbf{v}_2$	$RP_2(\mathbf{v}_1)$	RP_3	3	1
		—	(D)	D	4	
	F_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	RP	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	S_2	—	(D)	D	4	
	P_3	—	$RP_3(\mathbf{v}_1)$	RP_3	4	1
	F_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		$\mathbf{v}_1 + \mathbf{v}_2$	(D)	D	4	1
		—	(D)	D	5	
	RP_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
	S_3	$\mathbf{v}_{\parallel 1} \supset K_2$	(D)	D	4	1
		—	(D)	D	5	
	RP_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	2
		—	(D)	D	5	1
F_2	F_2	$\mathbf{v}_1 = \mathbf{v}_2$	$F_3(\mathbf{v}_{\perp 1})$	F_3	3	1
		—	(D)	D	4	
	RP	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	S_2	—	(D)	D	4	
	P_3	—	$RP_3(\mathbf{v}_1)$	RP_3	4	1
	F_3	$\mathbf{v}_1 = \mathbf{v}_2$	$F_3(\mathbf{v}_{\perp 1})$	F_3	3	2
		—	(D)	D	5	
	RP_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
	S_3	—	(D)	D	5	
	RP_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	2
		—	(D)	D	5	1
RP	RP	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	S_2	—	(D)	D	4	
	P_3	—	$RP_3(\mathbf{v}_1)$	RP_3	4	1
	F_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
	RP_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
	S_3	—	(D)	D	5	
	RP_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	2
		—	(D)	D	5	1
S_2	S_2	$K_1 = K_2$	$S_3(K_1)$	S_3	3	1
		—	(D)	D	4	
	P_3	—	(D)	D	5	
	F_3	—	(D)	D	5	
	RP_2	—	(D)	D	5	
	S_3	$K_1 = K_2$	$S_3(K_1)$	S_3	3	2
		—	(D)	D	5	
	RP_3	—	D	D	6	
P_3	P_3	—	P_3	P_3	3	3
	F_3	—	$RP_3(\mathbf{v}_2)$	RP_3	4	2
	RP_2	—	$RP_3(\mathbf{v}_2)$	RP_3	4	2
	S_3	—	D	D	6	
	RP_3	—	$RP_3(\mathbf{v}_2)$	RP_3	4	3
F_3	F_3	$\mathbf{v}_1 = \mathbf{v}_2$	$F_3(\mathbf{v}_{\perp 1})$	F_3	3	3
		—	(D)	D	5	1
	RP_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	2
		—	(D)	D	5	1
	S_3	—	(D)	D	5	1
	RP_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	3
		—	(D)	D	5	2
RP_2	RP_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	2
		—	(D)	D	5	1
	S_3	—	D	D	6	
	RP_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	3
		—	(D)	D	5	2
S_3	S_3	$K_1 = K_2$	$S_3(K_1)$	S_3	3	3
		—	(D)	D	5	1
	RP_3	—	D	D	6	1
RP_3	RP_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	4
		—	(D)	D	5	3

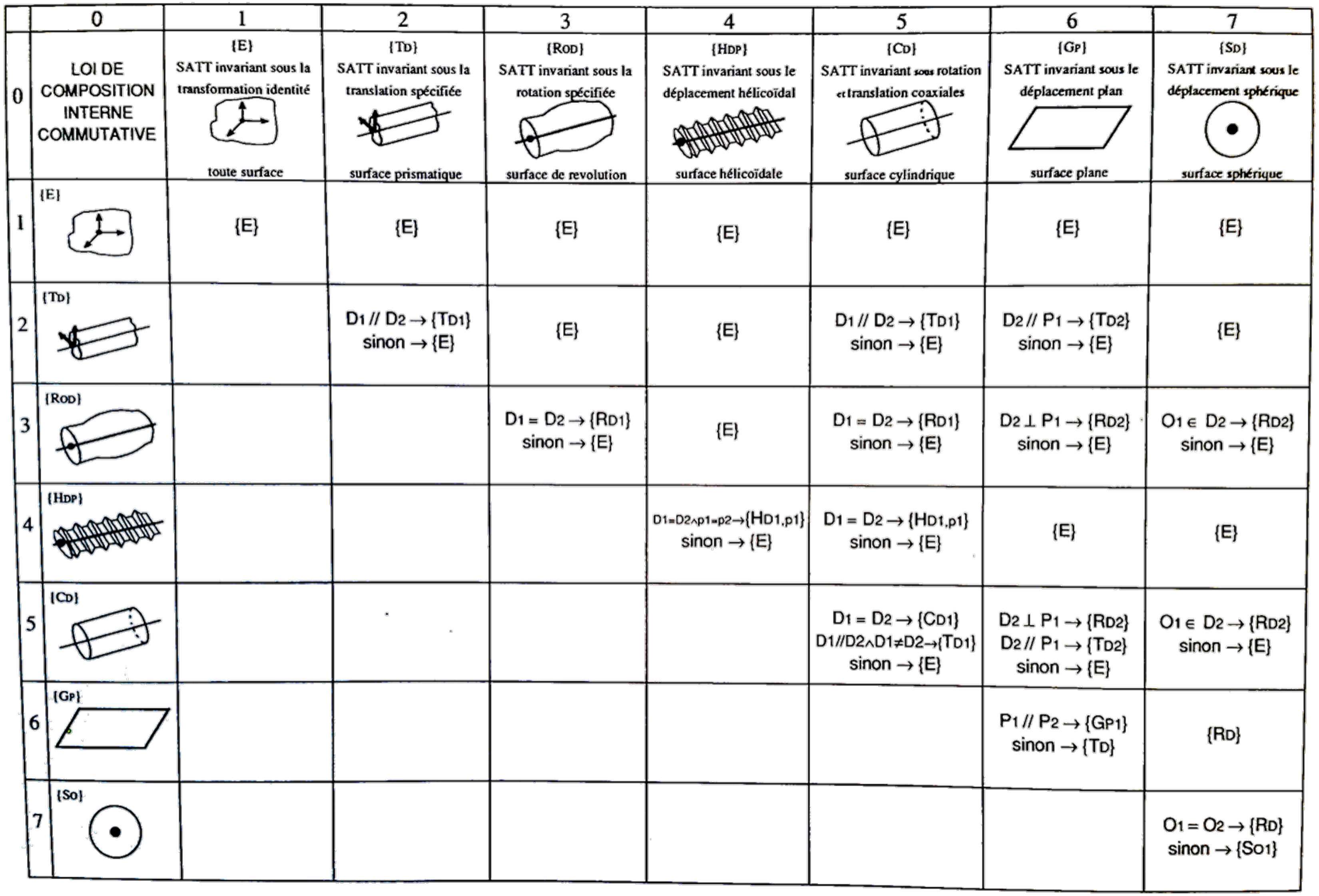
Figure A.1: Composition rules for algebraic constraints (Fanghella, 1988).



KC_1	KC_2	OPE	RES.KC	SMC	a	b
$H(\mathbf{v}_{\parallel 1}, h_1)$	P	$\mathbf{v}_1 = \mathbf{v}_2$	$C(\mathbf{v}_{\parallel 1})$	C	2	
		$\mathbf{v}_1 + \mathbf{v}_2$	$HP(\mathbf{v}_1, h_1)$	HP_2	2	
		—	$RP(\mathbf{v}_1)$	RP_3	2	
	R	$\mathbf{v}_{\parallel 1} = \mathbf{v}_{\parallel 2}$	$C(\mathbf{v}_{\parallel 1})$	C	2	
		$\mathbf{v}_1 = \mathbf{v}_2$	$HP(\mathbf{v}_1, h_1)$	HP_2	2	
		—	(D)	D	2	
	H	$\mathbf{v}_{\parallel 1} = \mathbf{v}_{\parallel 2}$ $\wedge h_1 = h_2$	$H(\mathbf{v}_{\parallel 1}, h_1)$	H	1	1
		$\mathbf{v}_{\parallel 1} = \mathbf{v}_{\parallel 2}$ $\wedge h_1 \neq h_2$	$C(\mathbf{v}_{\parallel 1})$	C	2	
		$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 = h_2$	$HP(\mathbf{v}_1, h_1)$	HP_2	2	
		$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 \neq h_2$	$RP(\mathbf{v}_1)$	RP_3	2	
		—	(D)	D	2	
	P_2	$\mathbf{v}_1 + \mathbf{v}_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	
		—	$RP_2(\mathbf{v}_1)$	RP_3	3	
	C	$\mathbf{v}_{\parallel 1} = \mathbf{v}_{\parallel 2}$	$C(\mathbf{v}_{\parallel 1})$	C	2	1
		$\mathbf{v}_1 = \mathbf{v}_2$	$RP_2(\mathbf{v}_1)$	RP_3	3	
		—	(D)	D	3	
	F_2	$\mathbf{v}_1 = \mathbf{v}_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	
		—	(D)	D	3	
	RP	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_2(\mathbf{v}_1)$	RP_3	3	
		—	(D)	D	3	
	S_2	—	(D)	D	3	
	P_3	—	$RP_3(\mathbf{v}_1)$	RP_3	4	
	F_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	RP_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	S_3	—	(D)	D	4	
	RP_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
$HP(\mathbf{v}_1, h_1)$	P	$\mathbf{v}_1 + \mathbf{v}_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	
		—	$RP_2(\mathbf{v}_1)$	RP_3	3	
	R	$\mathbf{v}_1 = \mathbf{v}_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	
		—	(D)	D	3	
	H	$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 = h_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	
		$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 \neq h_2;$	$RP_2(\mathbf{v}_1)$	RP_3	3	
		—	(D)	D	3	
	P_2	$\mathbf{v}_1 = \mathbf{v}_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	1
		—	$RP_3(\mathbf{v}_1)$	RP_3	4	
	C	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	F_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	RP	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	S_2	—	(D)	D	4	
	HP	$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 = h_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	1
		$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 \neq h_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	P_3	—	$RP_3(\mathbf{v}_1)$	RP_3	4	1
	F_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
	RP_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
	S_3	—	(D)	D	5	
	RP_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	2
		—	(D)	D	5	1
$HP_2(\mathbf{v}_1, h_1)$	P	$\mathbf{v}_1 + \mathbf{v}_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	1
		—	$RP_3(\mathbf{v}_1)$	RP_3	4	
	R	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	H	$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 = h_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	1
		$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 \neq h_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	
		—	(D)	D	4	
	P_2	$\mathbf{v}_1 = \mathbf{v}_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	2
		—	$RP_3(\mathbf{v}_1)$	RP_3	4	1
	C	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
	RP	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
	S_2	—	(D)	D	5	
	HP	$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 = h_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	2
		$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 \neq h_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	1
		—	(D)	D	5	
	P_3	—	$RP_3(\mathbf{v}_1)$	RP_3	4	2
	F_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	2
		—	(D)	D	5	1
	RP_2	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	2
		—	(D)	D	5	1
	S_3	—	D	D	6	
	HP_2	$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 = h_2$	$HP_2(\mathbf{v}_1, h_1)$	HP_2	3	3
		$\mathbf{v}_1 = \mathbf{v}_2$ $\wedge h_1 \neq h_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	2
		—	(D)	D	5	1
	RP_3	$\mathbf{v}_1 = \mathbf{v}_2$	$RP_3(\mathbf{v}_1)$	RP_3	4	3
		—	(D)	D	5	

Figure A.2: Composition rules for trascendental constraints (Fanghella, 1988).

A.2 TTRS association cases



	0	1	2	3	4	5	6	7
0	LOI DE COMPOSITION INTERNE COMMUTATIVE	{E} SATT invariant sous la transformation identité toute surface	{TD} SATT invariant sous la translation spécifiée surface prismatique	{ROD} SATT invariant sous la rotation spécifiée surface de revolution	{HDP} SATT invariant sous le déplacement hélicoïdal surface hélicoïdale	{CD} SATT invariant sous rotation et translation coaxiales surface cylindrique	{GP} SATT invariant sous le déplacement plan surface plane	{SD} SATT invariant sous le déplacement sphérique surface sphérique
1	{E}	{E}	{E}	{E}	{E}	{E}	{E}	{E}
2	{TD}		$D_1 // D_2 \rightarrow \{TD_1\}$ sinon $\rightarrow$ {E}	{E}	{E}	$D_1 // D_2 \rightarrow \{TD_1\}$ sinon $\rightarrow$ {E}	$D_2 // P_1 \rightarrow \{TD_2\}$ sinon $\rightarrow$ {E}	{E}
3	{ROD}			$D_1 = D_2 \rightarrow \{RD_1\}$ sinon $\rightarrow$ {E}	{E}	$D_1 = D_2 \rightarrow \{RD_1\}$ sinon $\rightarrow$ {E}	$D_2 \perp P_1 \rightarrow \{RD_2\}$ sinon $\rightarrow$ {E}	$O_1 \in D_2 \rightarrow \{RD_2\}$ sinon $\rightarrow$ {E}
4	{HDP}				$D_1 = D_2 \wedge p_1 = p_2 \rightarrow \{HD_{1,p1}\}$ sinon $\rightarrow$ {E}	$D_1 = D_2 \rightarrow \{HD_{1,p1}\}$ sinon $\rightarrow$ {E}	{E}	{E}
5	{CD}					$D_1 = D_2 \rightarrow \{CD_1\}$ $D_1 // D_2 \wedge D_1 \neq D_2 \rightarrow \{TD_1\}$ sinon $\rightarrow$ {E}	$D_2 \perp P_1 \rightarrow \{RD_2\}$ $D_2 // P_1 \rightarrow \{TD_2\}$ sinon $\rightarrow$ {E}	$O_1 \in D_2 \rightarrow \{RD_2\}$ sinon $\rightarrow$ {E}
6	{GP}						$P_1 // P_2 \rightarrow \{GP_1\}$ sinon $\rightarrow$ {TD}	{RD}
7	{SO}							$O_1 = O_2 \rightarrow \{RD\}$ sinon $\rightarrow$ {SO1}

Figure A.3: TTRS association cases (Desrochers and Clément, 1994).

Annex B

Simulation details

B.1 Mobility analysis of the brake system

$$\mathcal{T}_{1,1/1,0} = \begin{bmatrix} \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{11}M} \\ \boldsymbol{0} \mid \boldsymbol{x} \\ \boldsymbol{0} \mid \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{B.1}$$

$$\mathcal{T}_{1,0/3,0-a} = \begin{bmatrix} \boldsymbol{x} \mid \boldsymbol{x} \times \boldsymbol{O_{12}M} \\ \boldsymbol{y} \mid \boldsymbol{y} \times \boldsymbol{O_{12}M} \\ \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{12}M} \\ \boldsymbol{0} \mid \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 15,5 & 17 \\ 0 & 1 & 0 & -15,5 & 0 & -30,34 \\ 0 & 0 & 1 & -17 & 30,34 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.2}$$

$$\mathcal{T}_{1,0/3,0-b} = \begin{bmatrix} \boldsymbol{x} \mid \boldsymbol{x} \times \boldsymbol{O_{13}M} \\ \boldsymbol{y} \mid \boldsymbol{y} \times \boldsymbol{O_{13}M} \\ \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{13}M} \\ \boldsymbol{0} \mid \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 15,5 & 17 \\ 0 & 1 & 0 & -15,5 & 0 & 30,34 \\ 0 & 0 & 1 & -17 & -30,34 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.3}$$

$$\mathcal{T}_{1,0/2,0} = \begin{bmatrix} \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{14}M} \\ \boldsymbol{0} \mid \boldsymbol{x} \\ \boldsymbol{0} \mid \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{B.4}$$

$$\mathcal{T}_{1,0/3,0-c} = \begin{bmatrix} \mathcal{T}_{1,0/2,0} \\ \mathcal{T}_{2,0/3,0} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.5}$$

$$\mathcal{T}_{1,0/3,0} = \begin{bmatrix} (\mathcal{T}_{1,0/3,0-a})^{\perp} \\ (\mathcal{T}_{1,0/3,0-b})^{\perp} \\ (\mathcal{T}_{1,0/3,0-c})^{\perp} \end{bmatrix}^{\perp} \tag{B.6}$$

$$\mathcal{T}_{3,0/4,0-a} = \begin{bmatrix} \boldsymbol{x} \mid \boldsymbol{x} \times \boldsymbol{O_{34}M} \\ \boldsymbol{y} \mid \boldsymbol{y} \times \boldsymbol{O_{34}M} \\ \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{34}M} \\ \boldsymbol{0} \mid \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -60 \\ 0 & 0 & 0 & 0 & 60 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.7}$$

$$\mathcal{T}_{3,0/4,0-b} = \begin{bmatrix} \boldsymbol{x} \mid \boldsymbol{x} \times \boldsymbol{O_{35}M} \\ \boldsymbol{y} \mid \boldsymbol{y} \times \boldsymbol{O_{35}M} \\ \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{35}M} \\ \boldsymbol{0} \mid \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 60 \\ 0 & 0 & 0 & 0 & -60 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.8}$$

$$\mathcal{T}_{3,0/4,0-c} = \begin{bmatrix} \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{36}M} \\ \boldsymbol{0} \mid \boldsymbol{x} \\ \boldsymbol{0} \mid \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{B.9}$$

$$\mathcal{T}_{3,0/4,0} = \begin{bmatrix} (\mathcal{T}_{3,0/4,0-a})^{\perp} \\ (\mathcal{T}_{3,0/4,0-b})^{\perp} \\ (\mathcal{T}_{3,0/4,0-c})^{\perp} \end{bmatrix}^{\perp} \tag{B.10}$$

$$\mathcal{T}_{4,0/6,0-a} = \begin{bmatrix} \boldsymbol{x} \mid \boldsymbol{x} \times \boldsymbol{O_{62}M} \\ \boldsymbol{y} \mid \boldsymbol{y} \times \boldsymbol{O_{62}M} \\ \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{62}M} \\ \boldsymbol{0} \mid \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 15,5 & 17 \\ 0 & 1 & 0 & -15,5 & 0 & -30,34 \\ 0 & 0 & 1 & -17 & 30,34 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.11}$$

$$\mathcal{T}_{4,0/6,0-b} = \begin{bmatrix} \boldsymbol{x} \mid \boldsymbol{x} \times \boldsymbol{O_{63}M} \\ \boldsymbol{y} \mid \boldsymbol{y} \times \boldsymbol{O_{63}M} \\ \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{63}M} \\ \boldsymbol{0} \mid \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 15,5 & 17 \\ 0 & 1 & 0 & -15,5 & 0 & 30,34 \\ 0 & 0 & 1 & -17 & -30,34 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.12}$$

$$\mathcal{T}_{5,0/6,0} = \begin{bmatrix} \boldsymbol{z} \mid \boldsymbol{z} \times \boldsymbol{O_{64}M} \\ \boldsymbol{0} \mid \boldsymbol{x} \\ \boldsymbol{0} \mid \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{B.13}$$

$$\mathcal{T}_{4,0/5,0} = \begin{bmatrix} \boldsymbol{z} \mid \boldsymbol{z} \times \mathbf{0} \\ \mathbf{0} \mid \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{B.14}$$

$$\mathcal{T}_{4,0/6,0-c} = \begin{bmatrix} \mathcal{T}_{4,0/5,0} \\ \mathcal{T}_{5,0/6,0} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.15}$$

$$\mathcal{T}_{4,0/6,0} = \begin{bmatrix} (\mathcal{T}_{4,0/6,0-a})^{\perp} \\ (\mathcal{T}_{4,0/6,0-b})^{\perp} \\ (\mathcal{T}_{4,0/6,0-c})^{\perp} \end{bmatrix}^{\perp} \tag{B.16}$$

$$\mathcal{T}_{R} = \begin{bmatrix} \mathcal{T}_{1,1/1,0} \\ \mathcal{T}_{1,0/2,0} \\ \mathcal{T}_{2,0/3,0} \\ \mathcal{T}_{4,0/6,0} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.17}$$

$$\mathcal{W}_{R} = \mathcal{T}_{R}^{\perp} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.18}$$

B.2 Final calculated polytope

```
# HALFSPACES : a0 + a1.x1 + ... + an.xn >= 0.
2 28  28
# HALFSPACES : a0 + a1.x1 + ... + an.xn >= 0.
1 −0,310217 0,310217
1 −0,055907 0,440788
1 −0,074977 0,434133
1 −0,096103 0,426593
1 −0,256120 0,354627
1 −0,129377 0,414379
1 0,310217  −0,310217
1 0,074977  −0,434133
1 0,096103  −0,426593
1 0,256120  −0,354627
1 0,129377  −0,414379
1 0,055907  −0,440788
1 0,401563  0,000000
1 0,129377  0,414379
1 0,096103  0,426593
1 0,074977  0,434133
1 0,000000  0,459630
1 0,055907  0,440788
1 0,310217  0,310217
1 0,256120  0,354627
1 −0,401563 0,000000
1 0,000000  −0,459630
1 −0,129377 −0,414379
1 −0,096103 −0,426593
1 −0,055907 −0,440788
1 −0,074977 −0,434133
1 −0,310217 −0,310217
1 −0,256120 −0,354627
# GENERATORS : V = (v1, ..., vn)
0,758101  −2,172514
0,774344  −2,169709
1,453266  −1,770280
0,794768  −2,165107
0,991728  −2,103613
−0,774344 2,169709
−1,453266 1,770280
−0,794768 2,165107
−0,991728 2,103613
−0,758101 2,172514
−2,490272 0,733274
−0,794768 −2,165107
−0,774344 −2,169709
```

```
0,733274   -2,175663
-0,758101  -2,172514
-0,733274  -2,175663
-2,490272  -0,733274
-0,991728  -2,103613
-1,453266  -1,770280
2,490272   -0,733274
-0,733274  2,175663
0,794768   2,165107
0,733274   2,175663
0,774344   2,169709
0,758101   2,172514
2,490272   0,733274
0,991728   2,103613
1,453266   1,770280
# HALFSPACES : a0 + a1.x1 + ... + an.xn >= 0.
1 2
2 3
0 4
3 5
4 5
7 8
6 9
8 10
9 10
7 11
6 12
13  14
14  15
1 16
15  17
16  17
12  18
13  19
18  19
0 20
11  21
22  23
21  24
23  25
24  25
20  26
22  27
26  27
```