Product robustness management : formalization by formal logic, application to set based design, and tolerancing

Ahmed Jawad Qureshi

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

ARTS ET MÉTIERS ParisTech

2011-ENAM-0020

École doctorale n° 432:SMI- Sciences des Métiers de l'Ingénieur

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure d'Arts et Métiers

Spécialité " Génie Mécanique "

présentée et soutenue publiquement par

Ahmed Jawad QURESHI

Le 23 juin 2011

Contributions à la maîtrise de la robustesse des produits: Formalisation par logique formelle, applications à la conception ensembliste et au tolérancement

Directeur de thèse : **Jean-Yves DANTAN**
Co-encadrement de la thèse : **Jérôme BRUYERE, Régis BIGOT**

Jury

M. Alain DESROCHERS, Professeur, L'Université de Sherbrooke, Québec, Canada	Président
M. Luc LAPERRIERE, Professeur, Université du Québec à Trois-Rivières, Québec, Canada	Rapporteur
M. Thierry TISON, Professeur, LAMIH, Université de Valenciennes, Valenciennes, France	Rapporteur
M. François VILLENEUVE, Professeur, G-Scop, Université Joseph Fourier, Grenoble, France	Rapporteur
M. Régis BIGOT, Professeur, Arts et Métiers ParisTech, Metz, France	Examinateur
M. Jérôme BRUYERE, Maitre de conférences, INSA Lyon, Lyon, France	Examinateur
M. Jean-Yves DANTAN, Professeur, Arts et Métiers ParisTech, Metz, France	Examinateur

T H È S E

Arts et Métiers ParisTech - Centre de Metz
Laboratoire de Conception, Fabrication Commande (LCFC)-EA4495

Avant Propos

Ce mémoire de thèse représente une synthèse de mes activités de recherche menées depuis Septembre 2007. Ces activités ont été effectuées sous les statuts :

- d'allocataire d'une bourse ≪ *Higher Education Commision (HEC) Pakistan* ≫ de Septembre 2007 à Septembre 2010. Cette bourse a été obtenue suite à un concours national, pour le développement des ressources humaines.
- d'A.T.E.R. à Arts et Métiers ParisTech Metz avec des enseignements en informatique (algorithmique), génie industriel (optimisation des tolérances) et mathématiques (méthodes numériques) de Septembre 2010 à ce jour.

Ces travaux s'inscrivent dans la problématique de la gestion des variations pendant la phase de conception des produits mécaniques.

Etant donné l'aspect international de cette thèse et suite à la décision N° DG2009-46 du 1er Octobre 2009, le Directeur Général d'Arts et Métiers ParisTech autorise la rédaction en deux langues : anglaise et française. De ce fait, ce document se divise en deux parties : un résumé étendu en français sans figure et le descriptif des travaux en anglais avec les figures.

Dedicated to Inaya, my daughter.

Acknowledgements

First and foremost, I would to thank the members of jury of the thesis for their time, effort and consideration. I would like to thank Mr. Luc LAPERRIERE, Mr. Thierry TISON and Mr.François VILLENEUVE for accepting to be the referees, for their attention and for their detailed review and critique of my thesis. I would like to thank Mr. Alain DEROCHERS for presiding over the thesis jury and for his comments on the thesis.

I am indebted to my PhD director, supervisor and advisor, Professor Jean-Yves DANTAN, without whom, this thesis would not have been possible. Being my Master Thesis supervisor, He ignited the spark and gave me the impetus to start research in this domain. His guidance, determination and insight provided me with the direction whenever I needed it. His continuous support and encouragement both in curricular matters as well as in life was an immense source of motivation for me.

I am grateful to the rest of my doctoral committee for the amount of time that they invested in guiding me in their respective domains. I am thankful to Professor Jérôme BRUYERE who, in spite of being geographically away, provided a steady and intensive insight and guidance in modeling and other issues related to mechanical engineering. His commitment to the detail and rigor in my research has greatly improved the quality of this work. Every research work needs to be bounded by the realms of industrial demands and practicalities and Professor Régis BIGOT made sure that these were taken into account.

A special mention to other faculty members of our research group Professor Patrick MARTIN, Professor Ali SIADAT and Professor Alain

Etienne needs to be made, who were always there for any type of assistance and were instrumental in exposing me to the teaching profession.

I would like to thank Jean-Paul for initiating me in usage of *Mathematica*®.

Arts et Métiers has been my home for the past five years and it is the amazing people here that have made the stay a wonderful experience, most notably, Alaa, Amir, Armaghan, Cyril, Dong, Jinna, Jim, Liaqat, Marius, Mathieu, Sajid, Shirin, Somasunder, Xavier, the people at the administration and at the library.

Apart from the institution, there were friends who made life off the campus wonderful and more worthwhile. Thank you Fayez, Hetav, Indrava, Muneeb, Praveena, Saumya and Zine.

I am grateful to my parents, Shafiuddin QURESHI and Razia GUL, a constant role model for me in life. They instilled and cultivated in me the curiosity and imagination to dream and taught me the courage, perseverance, persistence and ambition to follow and realize it.

Lastly, I owe my deepest gratitude to my loving wife Saba and my daughter Inaya, who devotedly supported me through each and every moment during the last five years. A constant source of motivation, encouragement and help, Saba was and is my greatest source of support and the ray that kept me going through the stress and the hard times. I also wish to thank my entire extended family for providing a loving and supporting environment for me.

In the end, I gratefully acknowledge the financial support for this research from the Higher Education Commission of Pakistan (HEC), Arts et Métiers ParisTech France, Government of France, and the Lorraine Region.

Contents

List of Figures

List of Algorithms

Première partie

Résumé Étendu En Français

Chapitre 1

Introduction

1.1 Objectifs

L'objet de cette thèse est de définir et développer une nouvelle formalisation et des outils pour le processus de conception d'un produit mécanique permettant une approche globale de définition du produit en tenant compte des variations et des incertitudes. Il s'agit d'intégrer au plus tôt dans la définition du produit la notion de variabilité et l'impact des variations inhérentes aux processus de fabrication (ces deux notions variabilité et variation seront détaillées dans la suite).

1.2 Situation

La conception d'un produit peut être simplement décomposée en deux étapes : la recherche de concepts de solution et l'optimisation de la solution. Néanmoins, le niveau de performance optimale du produit peut éventuellement varier en fonction de perturbations (dispersions de fabrication, vieillissement, variations de l'environnement), ce qui peut s'avérer pénalisant.

L'objet de la conception robuste est d'optimiser en même temps les performances du produit et de minimiser la sensibilité aux perturbations. Afin de pouvoir intégrer la robustesse au cours de la conception, Taguchi (1987) a proposé une approche permettant de mesurer et prévoir la robustesse d'une solution. Un produit est dit robuste si sa réponse est peu sensible aux perturbations (paramètres non maîtrisés appelés bruits). Un produit optimisé mais qui ne fonctionne que dans des conditions particulières ne sera pas robuste.

Afin d'optimiser la définition de nouveaux produits, de réduire les délais d'ingénierie, il est utile de capitaliser les connaissances métier relatives aux différents intervenants (concepteur, fabricant) et d'intégrer ces connaissances dans une démarche globale de définition du produit. La notion d'intégration de la conception consiste à prendre en compte l'ensemble des contraintes du cycle de vie du produit dès la phase de conception.

1.3 Problématique

Ces travaux se focalisent sur la prise en compte des contraintes portant sur la robustesse du produit qui se traduiront en contraintes sur la variabilité admissible des paramètres du produit (espace des solutions) et sur les variations de ceux-ci (tolérances).

Deux notions apparaissent : variabilité et variation. La variabilité admissible des paramètres du produit représente l'ensemble des solutions viables du produit respectant les exigences identifiées lors du dimensionnement.

Les variations des paramètres sont les écarts entre la cible et la valeur réelle des paramètres, qui sont inhérentes aux imperfections des processus de fabrication.

Dans notre étude, la conception robuste vise à définir la variabilité admissible des paramètres du produit (espace de solution) afin de garantir la capacité du produit à maintenir ses performances, malgré des changements dans les conditions d'utilisation ou la présence de variations liées à ses paramètres ou à ses composants (Chen, 2007; Taguchi, 1987). D'autres définitions ont été proposées dans la littérature, la plus répandue est : « La conception robuste d'un mécanisme a pour objectif de rendre ses performances optimales et insensibles aux variations. »; celle-ci diffère légèrement de notre problématique de définition de la variabilité admissible.

Le tolérancement vise à définir les limites acceptables des variations des pièces permettant d'assurer un certain niveau de respect des performances du produit. En ingénierie concourante, le tolérancement a lieu le plus souvent à la fin de la phase de conception (Detail design). Pourtant, les effets des tolérances sont propagés à d'autres étapes de vie du produit (planification, fabrication, contrôle de qualité, ...) et ne sont pas intégrés lors du dimensionnement du celui-ci. Une approche intégrant la conception robuste et le tolérancement permettra la maîtrise de l'ensemble des variations et variabilités.

Donc, ces travaux de recherche visent à définir, des modèles, des méthodes et des outils permettant une démarche globale de la définition du produit et de ses tolérances. Il est nécessaire d'intégrer la spécification du produit dans le processus de définition et de dimensionnement du produit dans le cas des mécanismes de système mécaniques.

1.4 Positionnement De Travaux

Le travail présenté dans cette thèse se base sur les notions de propagation de contraintes quantifiées. Cette approche est basée sur un espace de description commun du produit via un ensemble de paramètres, une formalisation des différentes expertises sous forme de contraintes sur ces paramètres et le traitement de ces contraintes par propagation de celles-ci. La grande diversité des paramètres et l'extrême imbrication des contraintes d'un produit font que le concepteur, dans sa tentative d'optimiser son produit, ne fait que choisir des modes de résolution partiels (Le concepteur est confronté à tout moment à des alternatives sur les choix fonctionnels, structurels, technologiques et dimensionnels. Ces choix sont validés ou invalidés par des simulations. Les invalidations dues à des intégrations d'expertise métier génèrent des itérations à une étape amont de la conception, une remise en question d'un choix antérieur). En effet, le concepteur tente de procéder à une résolution séquentielle et itérative de chacune des contraintes. L'informatique apporte des outils d'aide à la conception permettant la gestion de la cohérence des paramètres et des contraintes du produit : les méthodes de Programmation Par Contraintes qui permettent d'explorer l'espace des solutions et de réduire le domaine de définition des paramètres représentant l'ensemble des alternatives de conception qui paraissent a priori satisfaisantes. Ces alternatives de conception correspondent aux variabilités admissibles dans notre étude.

Donc, le travail proposé s'appuie sur les domaines de la conception mécanique, l'informatique, et les mathématiques appliqués pour proposer une solution globale de gestion des variations pendant le processus de la conception d'un produit afin de rechercher l'ensemble des solutions robustes.

Ainsi, ces travaux se focalisent sur la recherche des solutions robustes : la variabilité admissible des paramètres du produit (espace des solutions), et sur le tolérancement de celles-ci : les variations admissibles (tolérances). Une approche sera développée permettant :

- Exploration de l'espace des solutions robustes
- Exploration de l'espace des variations

Le but est d'identifier les domaines des variables du produit qui satisfont les exigences fonctionnelles malgré les incertitudes, et d'analyser l'impact de l'ensemble des configurations des variations sur la performance du produit.

Le manuscrit est ainsi divisé en une introduction, 4 chapitres et une conclusion : Le premier est une étude bibliographique visant à expliciter les différentes briques nécessaires à la réalisation de l'étude : un état de l'art sur les processus de conception, sur la conception robuste et sur le tolérancement. Le second presente le formalisme unifié développé permettant la prise en compte des incertitudes et variations lors de la définition d'un produit. Le troisième expose les outils développés permettant une exploration de l'espace des solutions robustes. De la même manière, le quatrième expose les outils développés permettant l'analyse des tolérances.

Chapitre 2

État de L'Art

Il existe de nombreux travaux sur la formalisation des processus de conception et démarches de conception, sur les outils d'analyse des solutions, ...; par contre, il en existe peu sur la formalisation du problème de conception intégrant la prise en compte des incertitudes. Donc l'étude bibliographique a été divisée en trois grandes parties.

- Processus de la conception
- La conception robuste
- Tolérancement de système mécanique

2.1 Processus De La Conception

Concevoir un produit c'est passer de l'expression d'un besoin à la définition des caractéristiques d'un objet permettant de le satisfaire et à la détermination de ses modalités de fabrication. Le futur produit passe par une série d'état. La conception en tant qu'activité est une transformation provoquant un changement d'état du produit. Pour Mistree *et al.* (1990) , il s'agit d'un procédé de conversion d'information qui caractérise les besoins et exigences par un artefact, en connaissance sur le produit. Suh (2001) définit la conception comme l'interaction entre ce que nous voulons réaliser, et comment nous voulons le réaliser ; c'est-à-dire la transformation d'exigences fonctionnelles en paramètres de conception. Cette transformation peut être formalisée sous la forme d'un processus qui permet de définir une solution à un problème formalisé. Ce processus est structuré sous la forme d'une suite de prises de décision par le concepteur en tenant compte l'ensemble de contraintes technologiques, économiques, environnementales, ... Plusieurs autres définitions ou points de vue de processus existent (McDowell *et al.*, 2010), transformations paramétrique (Suh, 2001), la conception basée sur la propagation des contraintes (Ullman, 2003). Une définition générique de la conception est donnée par Gero (1990) : « *a goal-oriented, constrained, decision-making, exploration and learning activity which operates within a context which depends on the designer's perception of the context* ». L'objectif principal du processus de la conception est de « *to transform requirements - generally termed "functions", which embody the expectations of the purposes of the resulting artifact, into design descriptions* ».

Un modèle élémentaire de la conception peut être décrit :

$$F \rightarrow D$$

Où F sont les fonctions souhaitées, D est une description de la solution et $\rightarrow$ est une transformation (Gero, 1990).

Selon l'approche systèmatique décrite par Pahl et Beitz, le but de la conception est de mettre en place une méthodologie compréhensive, pour toutes les phases de conception et développement de systèmes techniques (Pahl & Beitz, 1996). Ce processus est décomposé en quatre phases :

- Clarification de la tâche : phase de spécification des informations dans une liste d'exigences,
- « Conceptual design », recherche de concepts : phase de recherche de la structure fonctionnelle et de solutions de principe ; puis définition du concept, exploration, évaluation et sélection. Cette phase détermine une solution de principe.
- « Embodiment design », conception architecturale : Les concepts sont traduits en architectures. Pendant cette phase, sont déterminées les choix structuraux, les choix de composants et de leurs paramètres pertinents, ainsi que les principales dimensions du système.

- « Detail design », conception détaillée : phase de production de plans et spécifications détaillés, de mise en place du processus de fabrication et de contrôle.

Une analyse comparative et critique de différents processus de conception est présentée dans les figures 2.2 et 2.3 par Scaravetti (2004). Cette analyse révèle que tous les processus de conception présentés ont des méta étapes communes (souligné dans les figures 2.2 et 2.3) :

- *Product planning and clarifying the task*
- *Conceptual Design*
- *Embodiment Design*
- *Detail Design*

En plus des processus détaillés dans les figures 2.2 et 2.3, il existe des processus de conception dont les étapes ne sont pas similaires et classés comme des démarches de conception. Ces démarches incluent la conception axiomatique (Benhabib, 2003; Suh, 2001), la conception robuste (Méthodes de Taguchi) (Benhabib, 2003; Phadke, 1989; Taguchi, 1978, 1987) et la conception ensembliste (Malak *et al.*, 2009).

Le travail présenté dans cette thèse se concentre sur les phases de conception préliminaire et détaillée. Il vise à fournir une approche unifiée et structurée permettant la maîtrise des variations lors de ces phases de conception.

Le risque de conception correspond au fait que le produit conçu ne réponde pas à l'ensemble des exigences. Le risque durant le développement de produit, résulte de l'exposition à une chance d'insuccès dû aux effets de l'ambigüité et/ou de l'incertitude :

L'ambigüité est un manque de compréhension sur les paramètres critiques d'un problème décisionnel ou sur la nature des relations entre les paramètres. Cela se traduit par l'incapacité pour le concepteur à construire un modèle adéquat du problème de conception ou inapproprié pour aider à la décision.

L'incertitude provient d'un manque d'information sur certaines variables pertinentes pour le problème de conception. L'incertitude peut découler du caractère stochastique de la variable. Elle peut aller jusqu'à un manque complet d'information (méconnaissance).

Par rapport aux risques de conception, les travaux de recherche présentés dans cette thèse ne se focalisent que sur l'incertitude et ne considère pas l'ambigüité. Après avoir situé nos travaux dans le processus de conception, nous allons nous intéresser aux démarches de conception et plus particulièrement aux processus décisionnels : « Point based design » et « Set based design ».

Durant la conception, plusieurs choix sont à valider, afin de passer aux phases suivantes. Chaque décision change l'état de la conception et la description du produit s'enrichie. Finch & Ward (1997); Ward & Seering (1993a,b) dans la méthodologie SBCE et Liker *et al.* (1996); Sobek *et al.* (1999); Ward *et al.* (1995) dans la philosophie SBCE illustrent les degrès de liberté de conception. L'espace des solutions s'élargi de manière pyramidale et il est exploré pour chaque concept, jusqu'à ce qu'une ou plusieurs solution de conception soit trouvée. Durant l'exploration des alternatives de conception, le concepteur prend des décisions comme la détermination de valeurs pour des paramètres de conception. Afin d'obtenir une ou plusieurs solution, il est nécessaire pour le concepteur d'explorer les alternatives générées par les différentes options de décision et de les comparer. Ces démarches dites ensemblistes ont été enrichies des outils d'exploration des espaces de solution par propagation de contraintes (Sébastian *et al.*, 2007; Yannou *et al.*, 2009; Yvars *et al.*, 2009; Yvars, 2008).

La démarche de conception de produit est itérative. A chaque niveau de la conception, le processus est itératif et récursif, et fourni un progrès incrémental du problème. Ces démarches sont dites « Point based design ». La conception du produit est au cours du projet une série d'aller-retour. Avec les outils traditionnels d'analyse assistée par ordinateur, les calculs et simulations se font avec des définitions avancées du produit ; les concepteurs emploient souvent une approche d'essai erreur. Elle produit quelques valeurs solutions sans donner les relations fondamentales entre les paramètres de conception. Dans ce contexte, notre étude se focalise sur les démarches « Set based design » pour éviter les décisions arbitraires, nous suggérons de définir les espaces admissibles des paramètres garantissant la robustesse des solutions.

2.2 Conception Robuste

Le concept de la robustesse a été introduit pour la première fois par Taguchi. Il a proposé dans les années cinquante de transformer le contrôle de qualité en ligne en contrôle de qualité hors ligne. Cette étape est à l'origine du concept de qualité robuste en ingénierie. Il fallut cependant attendre le début des années quatre-vingt-dix pour trouver les premières contributions dans la littérature qui intègrent ce concept dans le cadre de la conception Taguchi (1987) et Phadke (1989).La philosophie de Taguchi repose sur deux concepts : la fonction perte et le ratio signal/ bruit.

La fraction de produits en dehors des limites spécifiées est couramment utilisée comme une mesure de la qualité. Bien que ce soit une bonne mesure de la perte due au rebut, ce n'est pas un bon indicateur de la satisfaction du client. C'est la raison pour laquelle G.Taguchi a défini la fonction perte comme une mesure de la perte de la qualité subie par un client, due à une mauvaise conception du produit acheté. Le respect des tolérances ne garantit pas nécessairement une bonne qualité du produit. Phadke (1989) présente ainsi une fonction perte quadratique à la place de la fonction perte double échelons.

La robustesse d'une conception varie avec les valeurs des variables de conception. Le ratio signal/bruit est une mesure de la sensibilité de la conception aux changements des conditions environnementales et peut être utilisé pour calculer les valeurs optimales des variables de conception.

Dans tout problème de conception robuste, trois ensembles sont dissociés : l'ensemble des variables de conception, l'ensemble des paramètres de conception environnementaux et l'ensemble des fonctions performances. Les fonctions performances d'un système mécanique peuvent être multiples, elles sont impactées par les variations des variables de conception et des paramètres environnementaux. Afin de rendre le système robuste, le concepteur doit ainsi calculer les valeurs nominales des variables de conception afin de minimiser la variation des fonctions performances. L'approche de Taguchi est certainement la plus connue à l'heure actuelle en conception robuste. L'une des raisons est la simplicité d'utilisation du ratio signal/bruit comme critère d'optimisation. Une variante de l'approche de Taguchi à l'égard de la conception robuste par Chen *et al.* (1996) définit une conception robuste comme l'amélioration de la qualité d'un produit en minimisant les effets des variations sans éliminer les causes. Les problèmes sont dissociées en deux grandes catégories selon le type d'incertitude et les variations, elles portent sur :

- Type I - Minimiser les variations en performance causées par les facteurs des variations (paramètres incontrôlables).
- Type II - Minimiser les variations en performance causées par les variations dans les facteurs de contrôles (paramètres de conception).

La notion de robustesse de ce travail de recherche repose sur la définition de « Type I et Type II » fournie par Chen *et al.* (1996). Par conséquent, dans cette étude, la notion de robustesse s'applique aux variations des valeurs des paramètres de conception ainsi qu'aux variations des facteurs incontrôlables et imprévisibles tels que les variations des propriétés du matériau.

De nouvelles méthodes sont naturellement apparues dans la littérature, tant sur la modélisation des incertitudes (modèle probabiliste, modèle possibiliste, modèle ensembliste), que sur les techniques de propagation des incertitudes (Simulation de Monte Carlo, FORM SORM, ...).

Ayant positionné notre étude dans une démarche ensembliste, les outils proposés s'écartent des approches statistiques et probabilistes généralement employées.

2.3 Tolérancement

La prise en compte des variations est essentielle dans la phase de développement d'un produit. En effet, tout produit manufacturier est soumis à des variations de ses caractéristiques nominales ou cibles qui peuvent être inhérentes aux procédés et processus de fabrication, aux incertitudes sur les caractéristiques des matériaux et à l'environnement d'utilisation de celui-ci. Ces variations impactent les performances du produit. Les imperfections inhérentes aux procédés et aux processus de fabrication entraînent une dégradation des caractéristiques du produit, et donc de la qualité du produit. L'objectif de l'activité de tolérancement est de définir les limites acceptables des variations des caractéristiques « pièce » permettant d'assurer un certain niveau de qualité ou de robustesse à un coût optimal. L'allocation ou la synthèse des tolérances fonctionnelles est une importante étape du processus de conception qui, dans les pratiques courantes, se situe généralement durant les dernières étapes de la conception détaillée et impacte énormément la conception du processus de fabrication, la fabrication et le contrôle du produit.

Pour l'analyse des tolérances, il existe un grand corps de recherche sur les différentes approches (Chase & Parkinson, 1991; Hong & Chang, 2002; Nigam & Turner, 1995; Roy *et al.*, 1991). L'analyse des tolérances consiste à évaluer l'effet des variations géométriques des pièces par rapport aux conditions fonctionnelles ou aux conditions d'assemblage.

De nombreux travaux (figure 2.1) ont été réalisés sur l'analyse des tolérances d'assemblage mécanique.

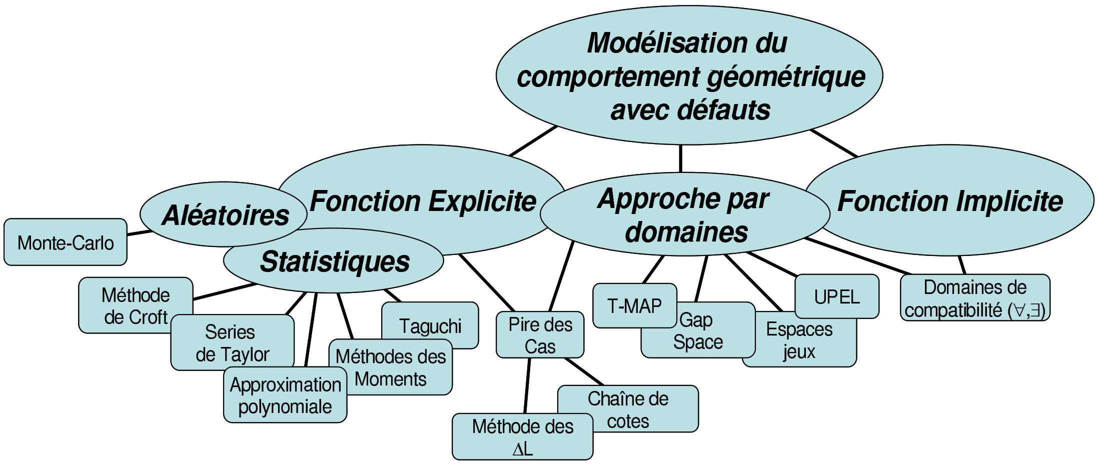

Figure 2.1: Les différentes approches d'analyse des tolérances

L'analyse des tolérances peut être effectuée au pire des cas ou statistiquement. Pour l'analyse des tolérances au pire des cas (analyse déterministe), l'analyse considère les plus mauvaises combinaisons possibles des différentes valeurs de tolérances et évalue les jeux et les caractéristiques géométriques fonctionnelles afin d'assurer la montabilité, le respect des exigences géométriques et l'interchangeabilité de 100% des mécanismes (la probabilité que les exigences géométriques soient respectées est égale à 1). Cette condition de 100% a généralement tendance à réduire les intervalles de tolérances et donc à augmenter le coût de fabrication. L'analyse statistique est une approche plus pratique et économique d'analyser les tolérances. On accepte un petit pourcentage de non-conformité. Mais les approches statistiques sont généralement limitées au mécanisme pour lesquels les ≪ response function ≫ sont définies explicitement ($Y = f(X)$) et continues dans un intervalle donné. Par contre, ces réponses dépendant de beaucoup des paramètres, il n'est pas toujours possible de trouver une fonction explicite dans un intervalle donné. La façon ≪ la plus simple ≫ pour analyser les tolérances est de simuler les influences des déviations sur le comportement géométrique du mécanisme. Pour la formulation mathématique de l'analyse des tolérances, le comportement géométrique est décrit en utilisant les différents concepts comme :≪ *Varitional geometry, Geometrical behavior law, clearence space and deviation space, gap space, response surface, kinematic models, . . .* ≫

Un des points essentiels de l'allocation de tolérance consiste à vérifier que la quantification des spécifications et son impact sur les paramètres du produit respectent les exigences spécifiées par le client. Cette vérification est réalisée par l'analyse des tolérances. Il existe à ce jour, deux grandes approches permettant de réaliser cette analyse des tolérances : l'approche par composition des déplacements et l'approche variationnelle. La première approche est basée sur la composition des déplacements, qui suppose que les causes de variation sont indépendantes, propose de les composer (les additionner) afin d'évaluer la variation résultante. On peut ainsi distinguer le cas 1D du cas 3D. Cette analyse s'effectue usuellement à l'aide des chaînes de cotes. Cette approche est basée sur une modélisation de la géométrie réelle par une géométrie de substitution, sur la modélisation d'un déplacement entre deux éléments géométriques. Elle est également basée sur la composition des déplacements dans les cycles topologiques. La seconde approche permettant l'analyse des tolérances est basée sur le concept de domaine. Il existe deux types de domaines : les domaines écarts qui représentent l'espace dans lequel peut évoluer la surface réelle considérant les tolérances, et les domaines jeux qui modélise les déplacements possibles dans une liaison cinématique. Contrairement à l'analyse par composition des déplacements qui se focalise sur l'expression des relations entre les jeux et les écarts des composants, l'approche par domaines s'appuie sur le mécanisme spécifié et tolérancé ce qui permet une réelle analyse des tolérances. L'objectif est alors de vérifier si la latitude offerte par les tolérances n'engendre pas des incompatibilités avec les exigences du produit. Cette méthode, basée sur les domaines effectue la somme Minkowski de ces domaines. Par cette somme, on peut qualifier l'approche par domaines comment étant une approche au pire des cas.

L'étape de tolérancement fait partie de l'étape de vérification et validation dimensionnelle d'un produit donc elle a un effet prononcé sur la robustesse du produit. Alors, il est nécessaire d'intégrer cette étape dans le cadre de la recherche de cette thèse. Donc les orientations de ces travaux sont en termes d'étape de tolérancement :

- Définition d'une syntaxe commune et harmonisée d'analyse des tolérances dans la phase de conception du produit
- Une prise en compte de la notion des quantificateurs proposé par Dantan & Ballu (2002)
- L'application d'une méthode de paramétrage géométrique pour les assemblages dont les « response function » sont implicites ($0 = f(X, Y)$),
- La modélisation du comportement géométrique des assemblages en 1D et 3D,
- Analyse des tolérances d'assemblage en utilisant un modèle statistique

2.4 Bilan

La gestion de la variation est une étape importante pendant la phase de la conception du produit pour assurer la performance de celui-ci. Pourtant, cette étape est souvent placée comme une activité avale du processus de conception.

La plus part des processus de conception traitent la gestion de la variation par deux phases : la phase de la conception robuste et la phase d'analyse des tolérances. Il existe des outils spécifiques pour chaqu'une de ces phases. Il n'existe pas actuellement de stratégie pour intégrer ces phases directement en amont de phase de conception. C'est-à-dire une méthodologie unifiée qui permet de prendre en compte les variations tout au long de la phase de conception. Donc, une syntaxe qui nous permet la prise en compte et l'expression des paramètres clés de la conception d'un produit et les variations associées avec les contraintes imposées est nécessaire pour assurer un processus concourant. Le chapitre suivant introduit les notions de base de la logique et propose un modèle basé sur la logique formelle pour exprimer un problème de conception du produit intégrant la maîtrise de l'impact des variations.

Chapitre 3

Formalisation pour la maîtrise des variations

L'utilisation des approches classiques d'exploration (CSP, optimisation, Approche probabiliste, ...) ne permettent pas une co-exploration des espaces de solutions et de variations.

Dans ces travaux, nous proposons une approche basée sur la logique formelle (Russell & Norvig, 2003) et la notion des quantificateurs (Bordeaux & Monfroy, 2006) et sur l'analyse par intervalles.

Le chapitre est divisé en trois grandes parties. La première partie concerne la formalisation théorique de l'approche. Dans cette partie, l'introduction de l'idée de la conception du produit en termes de problème de satisfaction des contraintes généralisé est présenté suivi par le développement de la syntaxe générale de celle-ci dans le contexte de la logique formelle. La deuxième partie se focalise sur la formalisation du problème de conception robuste dans le contexte théorique développé. La troisième partie de ce chapitre est consacré au tolérancement sous la forme d'un problème de satisfaction de contraintes quantifié utilisant la syntaxe théorique développée.

3.1 Formalisation théorique

Comme discuté dans le chapitre 2, nous pouvons considérer le processus de conception comme un processus séquentiel de prise de décisions. Ces décisions sont prises en tenant compte des contraintes qui traduisent les besoins du clients. Le processus de conception peut-être alors imaginé comme un modèle constitué des contraintes imposées par les clients en termes de conditions fonctionnelles ainsi que les variables clés qui contrôlent les performance du produit.

Suivant cette approche, Yvars (2008) définit le processus de la conception comme une problématique de satisfaction des contraintes sous la forme $\{V, D, C\}$, où V est l'ensemble des variables, D est l'ensemble des domaines de ces variables, et C est l'ensemble des contraintes imposées. S est défini comme l'ensemble des solutions tel que pour chaque membre s_i des solutions, toutes les variables de l'ensemble V ont valeurs dans leur domaine D et satisfont l'ensemble des contraintes C.

Plusieurs travaux de recherche avec différentes améliorations existent dans le domaine de la conception de systèmes mécaniques par cette approche (Chenouard, 2007; Thornton, 1996; Yannou *et al.*, 2007; Yvars, 2008).

Nous basons notre approche sur les travaux dans le domaine de la quantification des variables faites par (Dantan & Ballu, 2002) pour la spécification des tolérances d'assemblage. Ces travaux introduisent la notion des quantificateurs dans le tolérancement.

En utilisant cette approche, nous l'étendons à un cadre général qui traite du problème de la maîtrise de la variation dans la conception du produit en général avec l'aide de la logique formelle. De ce fait, nous présentons les notions de base en logique.

3.2 Notions de base en logique

Hurley (2008) définit la logique comme la science d'évaluer les arguments. En ingénierie, l'application de la logique est répandue dans les domaines du génie électronique, de la mécatronique, de la recherche opérationnelle ... Dans le domaine de génie industriel et du génie mécanique, l'utilisation de différents outils basés sur la logique pour la résolution et/ou l'optimisation des différentes problèmes est courante. Cependant, il n'existe pas une activité de recherche importante sur la formalisation et l'expression des problèmes mécaniques dans le domaine du génie mécanique. Souvent, les recherches portent sur le domaine appliqué des outils-informatiques.

Nous pouvons diviser le domaine de la logique en deux branches principales en termes d'évaluation d'expression soit : la logique classique bivalente ou, la logique polyvalente. La logique bivalente énonce que toute proposition doit être soit vraie soit fausse. Les travaux antérieurs dans le domaines de la logique bivalente appliquée au domaine du génie mécanique incluent les travaux faits par Ward (1989). D'autre part, la logique polyvalente ou les logiques multivaluées font partie du domaine des logiques non classiques et ajoutent un degré de la vérité à une évaluation. La logique floue est un exemple de la logique polyvalente. Des exemples de travaux appliqués, dans le domaine du génie mécanique, de la logique floue peuvent être trouvés dans les travaux de Massa *et al.* (2009).

Dans ce travail de recherche, nous développons une syntaxe structurée utilisant la logique formelle classique bivalente et nous présentons l'expression de la conception d'un produit dans ce paradigme. La base de la logique formelle est un langage qui a une syntaxe pré-définie contenant les symboles, connecteurs, prédicats et quantificateurs. Ils sont montrés dans le tableau 3.1.

Le langage de la logique formelle nous permet d'intégrer et d'harmoniser la notion des quantificateurs facilement pour le processus de conception d'un produit. L'idée générale d'un quantificateur est de généraliser un concept ou une relation qui existe pour une condition spécifique à une classe de variables et de conditions. Les deux types de quantificateurs définis sont : le quantificateur « il existe » $\exists$ et « quelque soit » $\forall$. La définition des quantificateurs utilisée dans cette thèse est montrée dans le tableau 3.2.

Les expressions contenant les variables quantifiées ne peuvent être résolues par les méthodes classiques des problèmes de satisfaction de contraintes. Ils tombent dans le domaine de résolution de problèmes de satisfaction de contraintes quantifiées (QCSP) et donc doivent être résolus par les méthodes de (QCSP)(Bordeaux & Monfroy, 2006).

3.3 Approche générale

Nous proposons une extension et une généralisation de l'expression de la conception du produit en CSP en utilisant le langage de la logique formelle. Pour un modèle $\{V, D, C\}$, pour trouver un ensemble de solutions s_i qui appartiennent à l'ensemble des solutions globales S, nous nous basons sur les définitions élémentaires de la logique formelle proposée par Peters & Westerstahl (2006); Russell & Norvig (2003). Les relations sémantiques (Annexe A.1 : Description de la sémantiques) pour un modèle en termes de relations quantifiées deviennent :

$$
\begin{aligned}
\mathcal{D} &\models (\varphi \wedge \psi)(\bar{a}) : \bar{a} \in D \Rightarrow s_i \models (\varphi \wedge \psi)(\bar{a}) \\
\mathcal{D} &\models \exists V(\varphi \wedge \psi)(V_{(\varphi \wedge \psi)}, \bar{a}) : \bar{a} \in D \Rightarrow s_i \models \exists V(\varphi \wedge \psi)(\bar{a}) \\
\mathcal{D} &\models \forall V(\varphi \wedge \psi)(V_{(\varphi \wedge \psi)}, \bar{a}) : \bar{a} \in D \Rightarrow s_i \models \forall V(\varphi \wedge \psi)(\bar{a})
\end{aligned}
$$

Ou $\mathcal{D}$ est un modèle pour V, φ et ψ sont deux contraintes symboliques contenant les variables de l'ensemble V, s_i est une solution, $\bar{a}$ sont les affectations pour les variables dans V pour la solution s_i.

Nous formulons ces expressions, maintenant, pour développer les formulations pour la conception robuste basée sur l'ensemble « *set based robust design* » et pour l'analyse de tolérances dans les sections suivantes.

3.4 Formalisation de l'approche ensembliste de la conception robuste

La conception robuste vise à définir la variabilité admissible des paramètres du produit (espace de solution) afin de garantir la capacité du produit à maintenir ses performances, malgré des changements dans les conditions d'utilisation ou la présence de variations liées à ses paramètres ou à ses composants. La définition générale de la robustesse, prise dans cette thèse, est celle proposée par Chen *et al.* (1996). Suivant cette définition, une solution est considérée robuste si elle intègre la gestion de variation de type I et type II simultanément dans la conception du produit.

3.4 Formalisation de l'approche ensembliste de la conception robuste

Pour définir la formalisation de l'approche ensembliste de la conception robuste, nous fournissons les définitions fondamentales pour le modèle basées sur la logique formelle et la conception basée sur les ensembles. V représente l'ensemble de toutes les variables dans la problématique de la conception. V contient l'ensemble des variables de conception DV ainsi que des variables de bruit Δ. Donc V peut être représenté comme $V = DV \cup \Delta$. D représente l'espace initial de la recherche de V. D est représenté par $D = D_{DV} \cup D_{\Delta}$. C représente l'ensemble des contraintes du modèle. Les contraintes pourront être de type continu, discret, complexe ou relationnel, si un modèle contient n contraintes, l'ensemble C pourra être écrit comme $C = \{c_1, ...c_n\}$.

Une solution s_i est considérée valide si il existe des affectations pour toutes les variables de conception en V telle que toutes les contraintes soient satisfaites. De la même façon, une solution est considérée robuste, si et seulement si, les contraintes sont satisfaites pour toutes les affectations sous forme d'ensembles pour toutes les variables de conception ainsi que pour les ensembles d'écarts pour toutes les variables de bruit.

Les définitions ci-dessus oblige l'utilisation des quantificateurs et alors le modèle mathématique général du système devient $\{QV, D, C\}$, où QV est l'ensemble des variables quantifiées. Le système émergeant est un problème de satisfaction des contraintes quantifiées. Les définitions des solutions pourront être traduites par les expressions logiques qui exigent l'existence d'une solution et d'existence d'une solution robuste :

Existence d'une solution La condition peut être écrite comme :

≪ **Il existe une solution appartenant à l'ensemble des solutions telle que pour au moins une affectation des variables de conception appartenant à leurs domaines, les exigences fonctionnelles sont satisfaites** ≫

Cela se traduit par :

$$\begin{aligned} \exists s_i \in S : s_i &= \mathcal{D}_i \models (D, C) \\ s_i &\models \exists \bar{v} C(\bar{v}, \bar{a}) \end{aligned}$$

Existence d'une solution robuste La condition peut être écrite comme :

« **Il existe une solution appartenant à l'ensemble des solutions telle que pour toutes les affectations des variables de conception et toutes les affectations des variables de bruit, appartenant à leur propres domaines, les exigences fonctionnelles sont satisfaites** »

Cela se traduit par :

$$\begin{aligned} \exists s_i \in S : s_i &= \mathcal{D}_i \models (D, C) \\ s_i &\models \forall \bar{v} \forall \bar{\delta v} C(\bar{v}, \bar{\delta v}, \bar{a}) \end{aligned}$$

Une solution qui remplie les deux conditions est une solution robuste et elle est insensible aux variations des paramètres de conception ainsi qu'aux variables des bruits.

3.5 Formalisation de l'analyse des tolérances

Le travail sur le tolérancement dans cette thèse est basé sur les travaux de recherche dans le domaine de spécification des tolérances suivant : Dantan *et al.* (2003a); Dantan & Ballu (2002). Ces travaux proposent les expressions quantifiées pour simuler l'influence des écarts géométriques sur le comportement d'un mécanisme en prenant en compte les écarts géométriques ainsi que l'influence des types de contacts. Leur approche stipule que la condition fonctionnelle doit être respectée dans au moins une configuration de jeux (le quantificateur « il existe ») ou que, la condition fonctionnelle doit être répétée dans toutes les configurations acceptables des jeux (le quantificateur « quelque soit »). Cette thèse harmonise et généralise cette expression dans le paradigme de la logique formelle et présente une formalisation de l'analyse de tolérances pour les mécanismes. Pour ça, d'une manière similaire à la formalisation de la conception robuste, nous proposons dans un premier temps, les définitions de base.

V représente l'ensemble de toutes les variables de la problématique de l'analyse de tolérances. Les définitions géométriques données par Dantan & Ballu (2002) sont retenues pour cette formalisation. A partir de ces définitions, le modèle géométrique doit modéliser les écarts des surfaces de chaque pièce (écart de situation et écart intrinsèque) et les déplacements relatifs entre pièces selon les variations des jeux (les jeux et conditions fonctionnelles). Le comportement géométrique est modélisé dans quatre espaces affines :

- l'espace des écarts de situation décrit les déviations angulaires et linéaires des éléments de la géométrie de substitution, par rapport à la géométrie nominale. Le vecteur Sd, exprimé dans l'espace « Situation », est défini par tous les paramètres des écarts de situation : un composant de Sd est donc un écart de situation.

- l'espace des écarts intrinsèques caractérise les variations de forme des éléments géométriques. Ils représentent les variations des paramètres intrinsèques des éléments de substitution, comme la variation du diamètre d'un cylindre, par exemple. Le vecteur I, exprimé dans l'espace « Intrinsèque », est défini par l'ensemble des écarts intrinsèques.

- l'espace des jeux décrit les déplacements relatifs entre les éléments de situation des surfaces de substitution qui sont nominalement en contact. Le vecteur G, exprimé dans l'espace « Jeu », est défini par l'ensemble des jeux.

- l'espace des jeux fonctionnels caractérise les déplacements relatifs entre les éléments de situation des surfaces de substitution fonctionnellement en relation. Le vecteur Fc, exprimé dans l'espace « Jeu Fonctionnel », est défini par l'ensemble des jeux fonctionnels.

Alors l'ensemble V est écrit comme :

$$V \quad = \quad DV \cup Sd \cup I \cup G \cup Fc$$

D pour la formalisation de l'analyse de tolérances contient les possibles affectations pour les variables dans V. L'ensemble des contraintes C contient les contraintes qui modélisent le comportement géométrique d'un assemblage en présence d'écarts. Ces contraintes peuvent être groupées en trois type :

- Hull de compatibilité ($H_{Compatibility}$). Il contient des équations de compatibilité entre les écarts et les jeux. L'ensemble des équations de compatibilité, obtenues en appliquant la relation de composition des déplacements aux diverses ≪ chaînes ≫ (chaînes de cotes - cycles topologiques), réalise un système d'équations linéaires. Afin que le système d'équations linéaires admette une solution, il est essentiel que les équations de compatibilité soient vérifiées. Ces équations décrivent un domaine de compatibilité défini dans l'espace de description (Sd, G, Fc).

- Hull d'interface ($H_{Interface}$). Il contient des inéquations et équations d'interface qui contraignent les jeux. Les contraintes de non interpénétration entre les différents composants se traduisent par l'expression d'inégalités entre les déplacements des surfaces de substitution en contact. Les contraintes d'association, qui modélisent le comportement attendu par une liaison cinématique (fixe ou glissante par exemple), se traduisent par des relations de dépendance entre les déplacements des surfaces de substitution. Ces équations et inéquations décrivent un domaine d'interface défini dans l'espace de description (G, I).

- Hull fonctionnel ($H_{Functional}$). Il contient les inéquations qui contraignent les jeux. Les contraintes fonctionnelles limitent l'orientation et la position relative des surfaces en relation fonctionnelle. Cette condition peut être exprimée par des contraintes, qui sont des inéquations. Ces dernières décrivent un domaine fonctionnel défini dans l'espace (Fc, I)

La formulation mathématique de l'analyse des tolérances est alors basée sur une expression du comportement géométrique du mécanisme ; plusieurs équations et inéquations qui modélisent le comportement géométrique du mécanisme sont définies. L'ensemble des contraintes C pourra être, alors, écrit comme :
$C = \{H_{Compatibility}, H_{interface}, H_{functional}\}$.

Afin d'exprimer l'analyse des tolérances à partir des expressions développées au dessus avec la syntaxe de la logique formelle, les relations relatives aux contraintes d'assemblage et aux conditions fonctionnelles peuvent être exprimées de la façon suivante :

Contrainte d'assemblage : La condition peut être écrite comme :

« **Pour tous les écarts admissibles (dans le domaine spécifié), il doit exister une configuration acceptable des jeux, telles que les contraintes de compatibilité et d'interface soient respectées.** »

Ce qui se traduit en termes de la logique par :

$$\begin{aligned} \exists s_g \in S : s_g &= \mathcal{D}_{assembly} \models (D, H_{Compatibility} \cap H_{Interface}) \\ s_g &\models \exists G \; H_{Compatibility} \cap H_{Interface}(V, \bar{a}) : \bar{a} \in D \end{aligned}$$

Contraintes de respect des conditions fonctionnelles : La condition peut être écrite comme :

« **Pour tous les écarts admissibles (dans le domaine spécifié) et pour toutes les configurations acceptables des jeux, il doit exister une configuration acceptable des caractéristiques telles que les contraintes de compatibilité, d'interface et fonctionnelles soient respectées.** »

Ce qui se traduit en termes de la logique par :

$$\begin{aligned} \exists s_{Fc} \in S : s_{Fc} &= \mathcal{D}_{Fc} \models (D, C) \\ s_{Fc} &\models \forall G \exists Fc \; C(V, \bar{a}) : \bar{a} \in D \end{aligned}$$

Une solution qui est validée pour les deux conditions (conditions d'assemblage et conditions fonctionnelles) satisfait les contraintes imposées. A partir de ces nouvelles expressions, nous pouvons envisager une approche appliquée qui repose sur ces deux conditions pour évaluer un assemblage à partir d'un domaine des écarts spécifié pour l'analyse de tolérances.

3.6 Synthèse

Ce chapitre développe et présente une formalisation harmonisée pour l'expression des deux étapes de maîtrise des variations dans le processus de conception du produit. Le travail est basé sur les travaux de recherche de Dantan & Ballu

(2002), qui ont proposés l'idée des quantificateurs pour l'interaction des écarts et des jeux dans un mécanisme. Ce travail étend et généralise cette approche avec le paradigme de la logique et développe son application pour la conception robuste et l'analyse des tolérances d'une manière structurée et syntaxique.

Cette formalisation nous permet d'exprimer les problèmes de conception non seulement en prenant en compte les variables de conception et des variables de bruit (variations et incertitudes) simultanément, mais aussi de les quantifier et ainsi de définir les relations et interactions entre eux. Cela nous permet de réaliser la conception robuste intégrée ensembliste.

Les deux chapitres suivants sont consacrés à l'élaboration de la méthode qui permet d'appliquer ce formalisme à différents exemples et de valider et évaluer la mise en œuvre de cette approche.

Chapitre 4

Application à l'approche ensembliste de la conception robuste

La formalisation pour la conception robuste du produit basée sur l'approche ensembliste est présentée dans le chapitre précédent. Ce chapitre a pour but de mettre en œuvre la méthode développée pour la conception robuste de systèmes mécaniques.

Ce chapitre est divisé en deux grandes parties. La première partie présente les outils disponibles pour transformer et appliquer la formalisation développée et présente un algorithme pour rechercher les ensembles des solutions robustes pour un modèle donné. La deuxième partie est dédiée à montrer l'application de la formalisation développée appliquée à quatre exemples choisis dans différents domaines du génie mécanique. L'approche développée est également validée par l'évaluation des solutions par d'autres techniques plus traditionnelles.

4.1 Considérations sur l'application de la formalisation de la conception robuste

Les expressions pour la conception robuste développées dans le chapitre précédent stipulent la condition d'existence d'une solution et d'une solution robuste. Afin de les appliquer, il est nécessaire de les transformer sous une forme algorithmique appliquée. Cet algorithme doit être capable de :

- Représenter, gérer et traiter l'espace global de la recherche comme défini dans le chapitre précédent.
- Transformer les expressions quantifiées dans le format appliqué et de traiter un modèle analytique pour la validation des conditions
- Pouvoir gérer le processus global sous forme algorithmique et cohérente. L'objectif de cette action est de filtrer l'espace de départ en appliquant les conditions transformées et de décomposer l'espace afin de trouver un résultat final sous forme d'ensembles de solutions robustes.

Une méthode de visualisation est également développée qui permet de visualiser et de manipuler les résultats trouvés par l'algorithme dans l'espace 2D et 3D afin d'aider les prise des décisions finales.

4.2 Représentation de l'espace de conception

L'espace de conception initial est l'espace de départ pour la recherche des solutions. La représentation propre de l'espace est nécessaire pour une mise en marche efficace de l'approche. En plus, pour implémenter le processus de la conception basée sur les ensembles, il est nécessaire de traiter l'espace et les valeurs de domaines sous forme d'ensembles au lieu des valeurs uniques. C'est pour ces raisons que nous choisissons la représentation et traitement de l'espace de la conception sous formes d'ensembles.

Les dimensions de l'espace de recherche dépendent du nombre de variables en V. Il peut être considéré comme un hyper-cube de dimensions n où n est la taille de l'ensemble V.

4.3 Expression de la consistance

Les conditions d'existence d'une solution et d'existence des solutions robustes forment les bases de l'évaluation pour la validation de l'espace de recherche. Une partie donnée de l'espace de recherche peut être une solution robuste seulement si cet espace est validé par ces deux conditions.

Afin d'appliquer ces deux conditions il est nécessaire de les transformer en un format adapté qui peut être mis en œuvre pour la validation des conditions. Pour cela, il faut définir le moyen de transformer et d'évaluer les expressions quantifiées.

La Programmation par Contraintes (PPC ou *Constraint Satisfaction Problem* soit CSP en anglais) est un ensemble de techniques pour résoudre des problèmes mathématiques, c'est-à-dire pour apporter des solutions à des variables mathématiques « contraintes » entre elles. A la différence du calcul formel qui propose de transformer les contraintes (usuellement sous la forme d'inéquations et équations) afin d'obtenir formellement les valeurs des variables, la PPC agit non pas sur les contraintes mais opère une réduction du domaine de définition des variables. En effet, les techniques de PPC visent à la réduction des domaines d'un ensemble de paramètres ou variables d'un problème qui sont contraintes afin de trouver les domaines finaux qui satisfont l'ensemble des contraintes et ainsi forment l'ensemble des solutions. La PPC permet de résoudre des problèmes fortement combinatoires et est adaptée particulièrement bien aux besoins de la conception mécanique.

La littérature sur l'utilisation de la PPC dans le domaine de la conception robuste et du tolérancement est très restreinte, voir inexistante. Par contre, nous pouvons citer les travaux sur les UCSP et les QCSP qui intègrent la notion d'incertitude.

Les problèmes de satisfaction de contraintes quantifiées (QCSP) (Quantified constraint satisfaction problem) (Bordeaux & Monfroy, 2006) sont une extension des problèmes de satisfaction de contraintes (CSP). La différence est l'utilisation de quantificateurs sur les variables, ce qui permet une plus grande expressivité des problèmes. Évidemment, le coût de cette expressivité est une augmentation de la complexité. Du coté des QCSP, la recherche n'en est encore qu'à ses débuts. Bordeaux et Montfroy ont étendu la notion d'arc-consistance (AC) des CSP aux QCSP. Mamoulis & Stergiou (2004) ont défini un algorithme d'AC pour les QCSP à contraintes binaires. Benhamou *et al.* (1999) ont proposé les techniques des « *Box Consistency* » et « *Hull Consistency* ». Quelques solveurs basés sur ces techniques ont été récemment développés pour les QCSP, dont QCSP-Solve, par Gent *et al.* (2005).

Dans ce travail, nous utilisons l'outil de « *Box Consistency* » (Benhamou *et al.*, 1999) pour transformer les expressions quantifiées puis évaluer la consistance d'un sous ensemble du domaine de recherche pour valider ou non les exigences.

Afin de transformer les conditions en utilisant « *Box Consistency* », nous transformons l'espace de départ ainsi que les contraintes sous la forme mathématique d'intervalles. Cette transformation est faite en suivant les travaux antérieurs (Parsons & Dohnal, 1992; Vareilles, 2005). Suivant cette transformation l'espace de recherche de départ sous forme d'intervalles est désormais nommés « D-Box ». Les sous ensemble d'espace de recherche deviennent « BSD ». Les expressions des contraintes, fonctions sur des réels, sont transformées en fonctions sur des intervalles.

Les notions de consistance, adaptées à la conception robuste peuvent maintenant être décrites de la façon suivante :

La consistance pour existence d'une solution L'expression de l'existence d'une solution s_i :

$$\begin{aligned} DV &= \{v_k | k \in [1,n], v_k \in d_{v_k}, \} \\ \mathbf{A} &= \{\mathbf{a}_{v_k}, | k \in [1,n], \mathbf{a}_{v_i} \in d_{v_k}\} \\ s_i &\models (\exists DV \min \mathbf{C}(V,\mathbf{A}) \vee \exists DV \max \mathbf{C}(V,\mathbf{A})) \end{aligned}$$

La consistance pour existence d'une solution robuste L'expression de la existence d'une solution robuste s_i :

$$\begin{aligned} V &= \{v_k, \delta_{v_k} | k \in [1,n], v_k \in d_{v_k}, \delta_{v_k} \in d_{\delta_{v_k}}\} \\ \mathbf{A} &= \{\mathbf{a}_{v_k}, \mathbf{a}_{\delta_{v_k}} | k \in [1,n], \mathbf{a}_{v_k} \in d_{v_k}, \mathbf{a}_{\delta_{v_k}} \in d_{\delta_{v_k}}\} \\ s_i &\models (\forall V \min \mathbf{C}(V,\mathbf{A}) \wedge \forall V \max \mathbf{C}(V,\mathbf{A})) \end{aligned}$$

Un sous ensemble BSD validé pour les deux expressions ci-dessus est une solution robuste pour toutes les valeurs unique des BSD

Ces expressions étant définis, nous pouvons maintenant les utiliser dans un algorithme pour réduire et filtrer l'espace de départ, à la recherche des solutions robustes.

4.4 Outils pour l'exploration de l'espace de conception

Pour la mise en œuvre des expressions explicitées ci-dessus, nous avons développé un algorithme qui est basé sur le principe de la recherche directe de l'espace solution. L'algorithme est décrit en algorithme 4.1. Il est divisé en :

- Découpage d'espace de recherche.
- Évaluation de l'existence de solution (consistance)
- Évaluation de l'existence de solution robuste (consistance)
- Triage de l'espace
- Présentation graphique des résultats

La démarche suivie par l'algorithme est la suivante. L'algorithme commence avec un espace de recherche de départ $D - Box$ donné par le concepteur. Pour démarrer la recherche la première partie de l'algorithme découpe l'espace $D - Box$ de recherche en divisant les intervalles de départs en sous-espace BSD et en analysant chaque BSD pour l'existence d'une solution. S'il en existe une, le BSD est en suite testé pour existence d'une solution robuste. Ce processus est répété jusqu'à la profondeur mentionnée ou jusqu'à fin de la recherche. A la fin, l'algorithme présente l'espace de recherche en trois parties (1) L'espace dans laquelle il n'existe pas de solution, (2) L'espace dans laquelle il existe au moins une solution (3) L'espace dans laquelle la solution est robuste sur tout l'intervalle. L'algorithme a été développé et testé sous Mathematica.

4.5 Exemple d'illustration

Cette section présente l'application et sa transformation appliquée à la conception robuste, sur un exemple simple.

Le système à concevoir est constitué d'une poutre simple. Le chargement extérieur ainsi que son poids propre engendrent uniquement un état de traction dans la poutre (figure 4.2. La poutre FG est paramétrée par sa longueur l, sa largeur a et son épaisseur b. Elle doit supporter une masse M. Le problème est de trouver les valeurs de a et b qui assurent que la masse m de la poutre demeure inférieure à la masse maximale admissible m_{adm} et que la contrainte maximale admissible de traction σ_{adm} ainsi que l'allongement maximal admissible dans la longueur de la poutre ∂_{adm} ne sont pas atteints. Ces contraintes sont représentées mathématiquement par les inéquations 4.18-4.20. Ces relations sont exprimées en termes des variables de la conception ainsi que les variables de bruits liées aux variables de la conception. La liste des variables, leurs domaines et les contraintes maximales sont données dans le tableau 4.2.

Toutes les contraintes, dans cet exemple, sont de type inéquations. Le but est de rechercher les ensembles pour les variables a et b tels que toutes les contraintes soient satisfaites. Dans un premier temps, les expressions logiques correspondantes pour la conception robuste sont données par les équations 4.21-4.22.

Cette formalisation est alors transformée par « *Box Consistency* » en utilisant l'arithmétique des intervalles. Les expressions finales pour la consistance de l"existence d'une solution et de la consistance de l'existence d'une solution robuste deviennent équation 4.26 et équation 4.27 respectivement.

L'exemple est traité par l'algorithme développé en 6 itérations et, à chaque itération, le domaine de chaque variable est divisé en 2. Le résultat est affiché en 2D en termes des variable a et b sur la figure 4.3. Chaque carré de la figure représente un espace de conception basé sur les ensembles pour a et b. Les carrés rouges (gris noir en NB) représentent l'espace sans solution. Les carrés violet (gris clair en NB) représentent les espaces des solutions robustes et les carrés jaunes (gris noir en NB) représentent l'espace à explorer.

Les résultats obtenus pour cet exemple ont été validés en les comparant avec les limites admissibles des contraintes imposées (inégalités remplacées par des égalités) ; (figure 4.4). Nous pouvons remarquer que les solutions trouvées par l'algorithme en utilisant les expressions logiques se trouvent bien entre les limites fixées par les contraintes.

4.6 Application à d'autres exemples

Afin de valider l'approche développée avec des exemples de conception de produits mécaniques, trois exemples sont présentés et traités. Chacun apporte un éclairage spécifique sur l'approche développée.

4.6.1 Structure à deux poutres

Un assemblage simple comprenant deux poutres est considéré. L'assemblage est donnée en figure 4.5

Dans l'exemple, le poids W est supporté par l'assemblage au point D. Les paramètres de conception sont la longueur de la poutre CD (L_1), son épaisseur et largeur (w_1, t_1), la longueur de la poutre AB (L_2), son épaisseur et sa largeur (w_2, t_2) et le positionnement de la liaison entre les deux poutres (L_3). Les contraintes de conception sont : le facteur de la sûreté d'assemblage basé sur des valeurs des contraintes mécaniques admissible, de la force maximale à la compression (limite de flambage), et le poids maximal de l'assemblage. Les variables de bruits sont associées aux variables de la conception ainsi que les caractéristiques de matériaux. Le tableau 4.3 présente les variables de l'exemple avec leur domaine de départ.

Le modèle analytique de base du système est représenté par l'équation 4.6.1 où, s_σ est le facteur de sécurité du système basé sur les contraintes, s_f est le facteur de sécurité du système basé sur la force de compression (F_{AB}, F_b) et W_{sys} est le poids maximal du mécanisme.

L'expression quantifiée pour la consistance de l'existence d'une solution pour ce modèle est donnée par l'inéquation 4.6.1. De la même façon, l'expression quantifiée pour la consistance de l'existence d'une solution robuste est donnée par l'inéquation 4.6.1.

Le résultat pour cet exemple est présenté en projection 3D, permettant d'observer les dimensions de poutre CD (w_1, t, l_1) pour trois itérations (figures 4.7-4.10). L'espace de départ est le grand cube transparent. Les parties bleues marquent l'espace de solutions robustes. L'espace vide marque l'espace sans solutions, et les parties vertes marquent les espaces avec solutions possibles.

Afin de valider cette approche, un balayage de l'espace de solutions a été fait. Ce balayage est montré en figure 4.12. Une comparaison des espaces de solutions robustes retourné par l'algorithme avec le domaine numérisé (balayé) nous montre la validité des solutions proposées par l'algorithme.

4.6.2 Accouplement à plateaux vissés

Cet exemple traite du problème de la conception d'un accouplement à plateaux vissés. Les critères de performance choisis sont la transmission de couple et la fiabilité de l'assemblage. Les paramètres fondamentaux de la conception de cet accouplement sont des paramètres dimensionnels continus et des paramètres discrets, liés à choix du nombre de vis (normes ISO), et au choix des matériaux. Une liste des variables considérées dans l'exemple est donnée en tableau 4.4.

Les contraintes principales prises en compte sont :

- Contraintes mécaniques liées à la capacité de transmission du couple
- Contraintes de sûreté et de la prise en compte d'incertitude et des variations dans les variables de conception
- Contraintes liées à l'assemblage et dimensionnement des composants

Les contraintes issues de ces trois catégories sont représentés par les équations A.9-A.21. Les expressions quantifiées pour cet exemple sont données par le système 4.42 pour la consistance de l'existence d'une solution et par le système 4.43 pour la consistance de l'existence des solutions robustes.
Le résultat obtenu est donnée en figure 4.19. Les espaces gris clairs montrent à chaque itération, les sous espaces de solutions possibles. Les espaces noirs sont les ensembles des solutions robustes. La mise en place de cette application nous a permis de valider le traitement de problème où les variables de conception

sont mixtes (variables discrètes et continus). Les résultats de cet exemple ont été validés par les méthodes classiques.

4.6.3 La presse à six barres

Le dernier exemple développé est un exemple qui montre l'application de notre approche à un problème géométrique et cinématique et qui met en œuvre des fonctions trigonométriques ainsi que des dérivées. Il s'agit de la conception d'une presse constituée d'un mécanisme à six barres. La figure 4.21 montre le mécanisme de la presse utilisé pour la simulation. Le tableau 4.6 présente les variables ainsi que leur domaine. Les objectifs principaux de ce problème de conception sont le dimensionnement de chaque barres de la presse pour :

- Assurer un assemblage qui permette la rotation de l'élément R_2
- Obtenir la fonction désirée de la position de l'élément G en fonction de la rotation de l'élément R_2
- Être d'un encombrement limité

Les contraintes mécaniques qui forment le modèle analytique de cet exemple sont données par les équations A.22-A.64 qui décrivent les contraintes d'assemblage, de positionnement et de montage. Les expressions quantifiées sont données par les équations 4.6.3.3-4.6.3.3.

Les travaux sur cette application ont mis en évidence la nécessité du traitement de contraintes temporelles par une méthode de bissection ainsi qu'un risque d'explosion combinatoire.

Afin de traiter ce type de problème, l'approche développée a été complétée par l'inclusion de fonctions spéciales qui permettent de traiter l'analyse d'intervalle et de résoudre des systèmes complexes trigonométriques et par l'inclusion de technique comme « *Bisection* » et « *Contraint Hierarchy* ».

Cet exemple a permis aussi d'identifier certaines limites de l'approche. Pendant le traitement de cet exemple, nous avons pu remarquer qu'avec les équations trigonométriques contenant des occurrences multiples d'une variable, l'approche souffre de problème de dépendance. A cause de ce problème les résultats retournés ne sont pas suffisamment précis et ils ne permettent pas de poursuivre la méthode de conception.

4.7 Conclusion.

Dans ce chapitre, nous présentons une discussion sur les outils, les méthodes, les transformations et les algorithmes nécessaires pour appliquer la formalisation de la conception robuste développée dans le chapitre 4.

Un processus complet pour transformer le problème de conception en expressions logiques et applicables est présenté. Ce processus est basé sur l'adaptation et utilisation d'outils du domaines de l'informatique et des mathématiques appliquées. Les conditions d'existence sont transformées à l'aide des mathématiques sur les intervalles et la notion de ≪ Box Consistency ≫. Aussi, un algorithme est développé qui est capable de faire une recherche directe dans l'espace de recherche par le principe de ≪ Branch and Bound ≫/ des solutions robustes.

Quatre exemples sont présentés dans ce chapitre qui montrent l'applicabilité de l'approche développées, tirés de différents domaines du génie mécanique. Ces exemples mettent en œuvre des variables continus et/ou discrètes. Deux exemples permettent de valider la démarche. Aussi d'un point de vu de génie mécanique, les problèmes de dimensionnement, les problèmes de sélections en utilisant des normes ainsi que les problèmes de positionnement ont été traités.

Le dernier exemple a exposé la limite de cette approche à cause du problème de dépendance des intervalles. En raison d'occurrence multiples des variables dans les expressions mathématiques de conditions, le résultat retourné par l'approche basée sur l'arithmétique des intervalles contient de fausses solutions. A cause de ce problèmes, les solutions ne sont pas précises et donc ne peuvent être utilisées pour un calcul précis. L'algorithme a été amélioré par l'addition de la technique de la bissection pour traiter ce problème.

Chapitre 5

Application à l'analyse des tolérances

La formalisation de l'analyse des tolérances dans la phase de la conception du produit a été discutée dans le chapitre 3. Pour implémenter cette formalisation et la rendre exécutable, il faut fournir des outils et des méthodes. Il faut aussi une adaptation de ces outils et méthodes qui permettent l'application de cette formulation aux assemblages mécaniques. Ce chapitre présente un algorithme constitué de différentes méthodes et outils utilisables pour l'analyse des tolérances des assemblages mécaniques.

5.1 Considérations pour l'application de la formalisation d'analyse des tolérances

Le problème de l'analyse des tolérances peut être divisé en trois étapes principales : Le modèle qui permet de représenter la géométrie du produit d'une manière paramétrique ; Les relations mathématiques pour modéliser le comportement du système incluant les variations et donc permettre la modélisation analytique de la fonction « Assembly response » ; Le développement d' une solution technique ou d'une méthode d'analyse qui éventuellement prend en considération l'analyse des tolérances d'un assemblage. Il est donc nécessaire de considérer les outils appropriés et les méthodes pour chacune des étapes.

L'objectif principal de la représentation géométrique du produit est de modéliser les écarts des surfaces de chaque pièce (écart de situation et écart intrinsèque) et les déplacements relatifs entre pièces selon les variations des jeux (les jeux et conditions fonctionnelles). Il s'agit d'une paramétrisation des défauts géométriques du mécanisme qui aboutit à une représentation paramétrique c'est-à-dire à un espace à n dimensions décrivant ces défauts.

La seconde étape concerne le développement d'un modèle analytique qui permet la modélisation de la fonction « assembly response » . Le but de cette étape est de générer un modèle analytique qui permet la capture de l'interaction géométrique entre les composants d'un assemblage et ses entités. Pour la formalisation de l'analyse des tolérances, cette étape doit finalement générer un ensemble des contraintes définies dans Rn (espace paramétrique de description des défauts).

La dernière étape de l'analyse des tolérances, vise à vérifier le respect des exigences d'assemblage et fonctionnelle au regard des tolérances de chacune des pièces. Différentes techniques peuvent être utilisés dans se sens, les deux méthodes connues sont : L'analyse des tolérances en pire des cas et l'analyse statistique des tolérances. Cette étape est importante en termes de formalisation de l'analyse des tolérances. Dans le contexte de notre étude, cette étape vise à vérifier la consistance des expressions quantifiées proposées dans le chapitre 3.

De plus, nous nous intéressons particulièrement aux mécanismes hyperstatiques dont la détermination de la fonction « Assembly response » sous une forme explicite est complexe.

5.2 Représentation des défauts géométriques

Dans un premier temps, il est important de spécifier que le formalisme explicité dans le chapitre 3 ne restreint pas l'utilisation de tel ou tel modèle de paramétrisation des défauts : torseur des petits déplacements, matrice homogène, … L'approche prise dans ce travail est la « parametrisation » des déviations par rapport à la géométrie nominale. La modélisation des déplacements s'appuie sur une description variationnelle par des matrices ou des torseurs des petits déplacements, la géométrie de substitution de la pièce est appréhendée par une variation de la géométrie nominale. Ces formalisations mathématiques caractérisent le déplacement d'un élément géométrique par rapport à un autre. La faible amplitude des défauts par rapport aux dimensions nominales des pièces permet de modéliser le déplacement qui associe deux éléments géométriques idéaux entre eux, par des petits déplacements. De ce fait, l'outil choisi pour modéliser les déviations géométriques est le torseur de petit déplacement (SDT) (Bourdet *et al.*, 1996) (Dantan & Ballu, 2002). Les composantes d'un torseur peuvent également être considérées comme des paramètres. Ceci nous permet de définir l'espace paramétrique des variations, dans lequel chaque axe de coordonnée de système représente une variable paramétrique. Un torseur $\{d\}$ constituée d'une partie de rotation (r) et une partie de translation (t) est donnée par l'équation 5.1. Pour ce faire, deux types principaux de torseurs sont utilisés, c'est-à-dire, torseurs des déviations et torseurs des jeux. Cette paramétrisation est détaillée dans la thèse de Dantan (2000)

5.3 Description du comportement - expression des contraintes par « Hulls »

La formulation mathématique de l'analyse de tolérance est basée sur une expression du comportement géométrique du mécanisme ; plusieurs équations et inéquations qui modélisent le comportement géométrique du mécanisme sont définies :

- Equations de compatibilité entre les écarts et les jeux. L'ensemble des équations de compatibilité, obtenues en appliquant la relation de composition des déplacements aux diverses « chaînes » (chaînes de cotes - cycles topologiques), réalise un système d'équations linéaires. Afin que le système d'équations linéaires admette une solution, il est essentiel que les équations de compatibilité soient vérifiées. Ces équations décrivent un domaine de compatibilité défini dans l'espace de description.

- Inéquations et équations d'interface qui contraignent les jeux. Les contraintes de non interpénétration entre les différents composants se traduisent par l'expression d'inégalités entre les déplacements des surfaces de substitution en contact. Les contraintes d'association, qui modélisent le comportement attendu par une liaison cinématique (fixe ou glissante par exemple), se traduisent par des relations de dépendance entre les déplacements des surfaces de substitution. Ces équations et inéquations décrivent un domaine d'interface défini dans l'espace de description (G, I).

- Inéquations qui contraignent les jeux. Les contraintes fonctionnelles limitent l'orientation et la position relative des surfaces en relation fonctionnelle. Cette condition peut être exprimée par des contraintes, qui sont des inéquations. Ces dernières décrivent un domaine fonctionnel défini dans l'espace.

En synthèse, les contraintes sont divisés en trois catégories : Le groupe des contraintes qui proviennent de « Compatibility Hull » ; Le groupe des contraintes qui proviennent de « Interface Hull », et le groupe des contraintes qui proviennent de « Functional Hull ». L'ensemble de ces contraintes forment le modèle de contrainte globale pour le problème de l'analyse des tolérances. Le degré et la complexité des contraintes peuvent être divisés en deux grandes catégories selon le degré de liberté : les contraintes pour les mécanismes isostatiques et les contraintes pour les mécanismes hyperstatiques. Dans le cas de la plupart des mécanismes isostatique, la fonction « assembly response » est explicite. Mais dans le cas des mécanismes hyperstatiques, la fonction « Assembly response » est plus complexe et peut-être implicite.

5.4 Développement des méthodes d'analyse

La formalisation du problème d'analyse des tolérances et les paramétrisations s'appuient sur les travaux de thèse de Dantan (2000). L'opérationnalisation de ceux-ci est partie intégrante de cette étude. L'objectif de la méthode d'analyse est de fournir des techniques pour estimer l'effet des tolérances attribuées sur les exigences. Un état de l'art des techniques utilisées pour l'analyse des tolérances a été discuté dans la section 2.3 ; deux types de techniques existent des techniques au pire des cas ou des approches probabilistes. La formalisation proposée dans le chapitre 3 est assimilable à une formulation au pire de cas. Pour l'analyse des tolérances au pire des cas (analyse déterministe), l'analyse considère les plus mauvaises combinaisons possibles des différentes valeurs de tolérances et évalue les jeux et les caractéristiques géométriques fonctionnelles afin d'assurer la montabilité, le respect des exigences géométriques et l'interchangeabilité de 100% des mécanismes (la probabilité que les exigences géométriques soient respectées est égale à 1). Cette condition de 100% a généralement tendance à réduire les intervalles de tolérances et donc à augmenter le coût de fabrication. L'analyse statistique est une approche plus pratique et économique d'analyser les tolérances, pour lequel on accepte un petit pourcentage de non-conformité. De ce fait, il est important d'étudier une opérationnalisation de la formalisation via une approche au pire des cas et une approche statistique.

5.4.1 L'approche d'analyse des tolérances par pire de cas

Le champ d'application de la formalisation de l'analyse des tolérances mentionné dans ce travail est limité à l'expression du problème de l'analyse des tolérances via l'expression logique proposée dans le chapitre 3. Dans ce chapitre, nous avons déjà discuté des similarités entre la formalisation du problème d'analyse des tolérances et de la formalisation d'un QCSP. Afin d'opérationnaliser l'analyse des tolérances à partir des expressions avec quantificateurs, les relations relatives aux contraintes d'assemblage et aux conditions fonctionnelles ont été modifiées afin de tendre vers un formalisme QCSP de manière suivante :

Contrainte d'assemblage : Pour tous les écarts admissibles (dans le domaine spécifié), il doit exister une configuration acceptable des jeux, telles que les contraintes de composition et d'interface soient respectées.

Conditions Fonctionnelles : Pour tous les écarts admissibles (dans le domaine spécifié) et pour toutes les configurations acceptables des jeux, il doit exister une configuration acceptable des caractéristiques telles que les contraintes de composition, d'interface et fonctionnelles soient respectées.

L'algorithme permettant d'opérationnaliser ces expressions se base sur les techniques utilisées en « Quantified Constraint Satisfaction Problem ».

Les problèmes de satisfaction de contraintes quantifiées (QCSP) sont une généralisation des problèmes de satisfaction de contraintes (CSP), dans lesquels chaque variable est quantifiée soit existentiellement, soit universellement. La différence est l'utilisation de quantificateurs sur les variables, ce qui permet une plus grande expressivité de problèmes. Un réseau de contraintes quantifiées est une formule de la forme $\{Q, D, C\}$dans lequel :

- Q est une séquence de variables quantifiées : $\{Q_1 x_1, ...Q_n, x_n\}$ avec $Q_i \in \{\forall, \exists\}$ et x_i une variable au sens des CSP
- D est un ensemble de domaines :$\{D_1, ..., D_n\}$ avec $\forall i \in \{1, ..., n\}$ $x_i \in D_i$
- C est une conjonction de contraintes : $\{C_1, ..., C_m\}$

Ainsi, nous pouvons expliciter un problème d'assemblage via le formalisme $QCSP$:
$\forall$écarts $\in$ **Tolérances**,

$\exists$jeux $\in$ **Cont.non_interpénétration**,

: (écarts,jeux)soient**compatibles**

Nous pouvons formuler ce problème via le formalisme QCSP :

$$\underbrace{\{\forall x_1, \forall x_2, ..., \forall x_n,}_{\text{pour tous les écarts}} \underbrace{\exists x_{n+1}, \forall x_2, .., \exists x_m;}_{\text{il doit exister une configuration des jeux}} \underbrace{D(x_1), D(x_2), .., D(x_n),}_{\text{tel que chaque écart appartienne à cet intervalle}} \underbrace{D(x_{n+1}), D(x_{n+2}), ..., D(x_m);}_{\text{,que chaque jeux appartienne à cet intervalle}} \underbrace{c_1, .., c_p\}}_{\text{, et que l'ensemble de ces contraintes soit vérifié}}$$

L'expressivité des QCSP ouvre de nombreuses perspectives pour l'analyse des tolérances. Évidemment, le coût de cette expressivité est une augmentation de la complexité.

L'algorithme QCSP vérifie la consistance des contraintes quantifiées : dans le cas où le problème est « arc-consistent » il renvoie VRAI, il renvoie FAUX dans le cas où le problème n'est pas « arc consistent ».

Le pouvoir expressif de QCSP intègre la notion de quantificateur dans l'expression mathématique et nous permet de modéliser la formulation de l'analyse des tolérances pour les exigences d'assemblage « assembly requirement » avec un modèle de tolérancement vectoriel.

5.4.2 L'analyse statistique des tolérances basée sur la formalisation et la simulation Monte Carlo

Dans cette section, nous proposons de transformer les conditions logiques présenté dans les sections 3.5.4.1 et 3.5.4.2 pour le problème de l'analyse des tolérances tel que l'analyse soit probabiliste. A partir de cette nouvelle formulation, différentes techniques seront testées comme l'optimisation, la simulation Monte Carlo et la propagation de contraintes pour opérationnaliser l'analyse statistiques des tolérances.

Respect de la condition d'assemblabilité du mécanisme La condition d'assemblabilité du mécanisme dans la section 3.5.4.1 décrit la première condition essentielle à remplir par l'assemblage afin d'assurer que les composants s'assemblent. Afin d'appliquer la condition ci-dessus dans une forme implémentable par simulation Monte Carlo, la formulation 3.46 doit être transformée dans l'expression suivante :

« Pour chaque échantillon (simulation Monte Carlo) des écarts acceptables (écarts qui sont à l'intérieur des limites des tolérances), il existe un configuration des jeux telle que l'exigence d'assemblabilité (contraintes d'interface) et les équations de compatibilité soient vérifiées »

Respect des Conditions fonctionnelles (Solution Robuste) La condition de respect des conditions fonctionnelles est donnée dans la section 3.5.4.2. Cette condition décrit la deuxième condition à remplir par les configurations validées par la condition de l'assemblabilité. L'évaluation de la deuxième condition vérifie que pour toutes les configurations assemblables, l'assemblage se conformera aux conditions de fonctionnement. Afin d'appliquer la condition ci-dessus dans une forme implémentable par simulation Monte Carlo, la formulation 3.47 doit être transformée dans l'expression suivante :

≪ Pour chaque échantillon (simulation Monte Carlo) des écarts acceptables (écarts qui sont à l'intérieur des tolérances), le pire des cas des caractéristiques fonctionnelles doit respecter les besoins fonctionnels tels que les contraintes d'interface et les équations de compatibilité soient vérifiées ≫

Un organigramme général décrivant l'algorithme pour l'analyse des tolérances est représenté en figure 5.6. Les principes de celui-ci sont explicités dans la suite :

Les écarts des composants (S, I) sont évalués de manière aléatoire à chaque itération de la simulation de Monte Carlo. La simulation Monte Carlo est utilisée pour générer des variables aléatoires simulant les écarts des composants. Un tirage des écarts générés par la simulation Monte Carlo est noté à $S' = \{s'_1, s'_2, ..., s'_n\}$ et $I' = \{i'_1, i'_2 i'_o, ..., \}$. Pour chaque tirage, les écarts générées doivent satisfaire l'ensemble des contraintes $(S', I') \in H_{specification}$, il s'agit de vérifier le respect des tolérances à analyser.

Implémentation de la vérification du respect de la condition d'assemblabilité Pour évaluer l'assemblablité de chaque tirage, nous vérifions qu'il existe une configuration admissible des jeux du mécanisme tel que les obligations d'assemblabilité (contraintes d'interface) et les équations de compatibilité soient vérifiées. Cette vérification est effectuée via la transformation de la relation 3.46 en la formule 5.10. La vérification d'existence peut être faite par propagation de contraintes (filtrage) et " Elementary Box consistency " ou par optimisation sous contraintes. Les deux solutions ont été validées. La seconde solution a été préférée, elle est formalisée par l'équation 5.11, et son implémentation est présentée via le pseudo-code 5.2.

Implémentation de la vérification du respect des conditions fonctionnelles La même démarche a été faite pour étudier l'implémentation de la vérification du respect des conditions fonctionnelles : Pour évaluer le respect des conditions fonctionnelles de chaque tirage, nous vérifions si, pour toutes les configurations admissibles des jeux du mécanisme, les conditions fonctionnelles, les équations de compatibilité et les conditions d'interface sont vérifiées. Cette vérification est effectuée via la transformation de la relation 3.47 en la formule

5.12. De même, cette formulation a été transformée en un problème d'optimisation sous contraintes 5.13 dont l'implémentation est détaillée via le pseudo code 5.3.

Ainsi, les probabilités de respect de la condition d'assemblabilité et de respect des conditions fonctionnelles peuvent être estimées par simulation de Monte Carlo.

5.5 L'application

L'approche 3D proposée ci dessus pour des assemblages mécaniques a été appliquée et validée sur différents modèles avec et sans spécifications GD& T. Pour des raisons de brièveté et de clarté de l'application, un mécanisme simple et hyperstatique est présenté comme exemple 5.7- 5.9. Les deux plateaux sont assemblés par trois axes de guidage. Cet exemple inclut 32 variables « écart » , 24 variables « jeux », 6 variables de condition fonctionnelle, 3 boucles topologiques des exigence d'assemblabilité donc 18 équations de compatibilité (équations linéaires), 3 boucles topologiques des exigences fonctionnelles donc 18 équations des exigences fonctionnelles (équations linéaires), 6 contraintes de non-interférence (6 non inéquations linéaires), 6 contraintes de contact fixe (12 équations linéaires), et 2 contraintes des exigences fonctionnelles (2 non inéquations linéaires).

L'analyse de tolérance a été réalisée en utilisant la distribution normale, pour chaque déplacement des points extrêmes de chaque trou, avec une moyenne à zéro et écart-type spécifique à chaque dimension dérivée des tolérances spécifiées. Les résultats sont présentés dans le tableau 5.1 pour différentes valeurs de tolérances. Cet exemple intègre des contraintes non linéaires (certaines équations, inéquations, ou formulations sont détaillées ...) qui ont nécessité l'utilisation de méthodes numériques d'optimisation sous contraintes. Afin de vérifier l'absence de violation de contraintes, deux méthodes de contrôle d'erreur ont été programmés.

5.6 Conclusion

Suite aux propositions de formalisation du problème d'analyse des tolérances faites dans le chapitre 3, ce chapitre développe la partie opérationnalisation de celles-ci. Afin d'appliquer le cadre logique, il est nécessaire de déterminer les méthodes, techniques et outils qui permettent la transformation des expressions en forme implémentables. Ce chapitre donne un aperçu du processus de choix, d'utilisation et de développement des algorithmes nécessaires qui permettent cette application.

Deux transformations ont été envisagées afin de développer une approche d'analyse des tolérances au pire des cas et une approche d'analyse statistique des tolérances.

Pour l'analyse des tolérances au pire des cas, le déploiement d'un QCSP a été effectué. Malgré la puissance expressive des QCSP, l'utilisation de ceux-ci dans ce contexte a montré des limites :

- Utilisation possible que dans le cas des exigences d'assemblage ; dans le cas des conditions fonctionnelles la définition du domaine de chaque jeu est fonction des valeurs prises par les écarts des pièces, cette dépendance sur la définition des domaines n'est pas modélisable via un QCSP.

- Utilisation possible uniquement dans le cas d'un tolérancement vectoriel où chaque écart est limité indépendamment ; la définition du domaine de chaque écart via un intervalle ne permet pas de modéliser des spécifications géométriques qui limitent plusieurs écarts.

Pour l'analyse statistique des tolérances, le déploiement de la simulation de Monte Carlo couplée avec des optimisations a été effectué. L'utilisation de ceux-ci a montré beaucoup moins de limites que le déploiement précédent : utilisation possible dans le cas des exigences d'assemblage et des conditions fonctionnelles, utilisation possible pour différents types de tolérancement, utilisation possible pour des mécanismes fortement hyperstatiques et au comportement non linéaire. La principale limite est le coût de calcul, la simulation de Monte Carlo nécessite un nombre important de tirages. De ce fait, l'une des principales perspectives est le déploiement d'autres approches probabilistes couplées avec des algorithmes d'optimisation ou d'autres techniques ensemblistes afin de réduire le coût de calcul.

Chapitre 6

Conclusion et perspectives

Cette thèse présente, une formalisation généralisée pour définir, modéliser et exprimer la problème de la maîtrise des variations dans le processus de la conception du produit utilisant la philosophie de la conception basée sur les ensembles ≪ Set Based Design ≫.

La formalisation intègre la notion de l'ingénierie concurrente en permettant l'exploration simultanée de l'espace de conception ainsi que l'espace de la variation. Ceci nous permet de chercher l'espace global pour les solutions robustes pour une problématique de conception. Le travail développé propose ainsi :

- Une formalisation abstraite pour exprimer le problème de la conception d'un produit mécanique en prenant compte des variables de la conception, les variables de la variation, l'espace global de recherche et les contraintes qui modélisent le système avec le pouvoir d'expression de la logique formelle.
- La capacité de quantifier les interactions des différents variables entre elles.
- Des expressions harmonisées pour exprimer la conception robuste intégrée dans la phase de la conception utilisant la formalisation développée.
- L'expression harmonisée pour exprimer la méthode d'analyse des tolérances dans la phase de la conception utilisant la formalisation développée.
- Une méthode pour appliquer la formalisation développée pour la conception robuste, illustrée par plusieurs exemples de conception de produit

mécanique. Cette méthode implique le choix des outils nécessaires et a conduit au développement d'un algorithme permettant la recherche d'espace de solutions robustes.

- Une méthode pour appliquer la formalisation développée pour l'analyse de tolérances, appliquée à un assemblage hyperstatique, avec le choix des outils nécessaires et développement d'un algorithme permettant de faire un analyse statistique des tolérances avec optimisation.

Pour atteindre ces objectifs, le travail unifie les avancements dans les domaines de la conception, le génie mécanique, la logique et l'informatique.

L'approche de la conception « Set based design » intégrée avec la notion de variation nous donne le pouvoir de trouver les ensembles de solution qui sont insensibles aux variations dans l'espace de la conception ainsi que l'espace des variations. Afin de permettre cette intégration, le paradigme de la logique a été utilisé qui nous permet de quantifier les expressions et leur relations dans l'espace de la recherche. Cette formalisation a été appliquée pour étendre le travail existant dans le domaine de l'analyse de tolérances par Dantan & Ballu (2002) et donc nous a permi d'exprimer la deuxième étape importante de la maîtrise de variations dans un cadre uniforme.

Enfin, nous avons montré l'application de la formalisation de la conception robuste en traitant quatre exemples de différents niveaux de complexité. Pour cela, nous avons transformé l'expression des contraintes via l'arithmétique des intervalles et « Box consistency » en utilisant un algorithme développé pour la conception robuste. Les résultats obtenus sont validés par les approches classiques. Nous avons démontré aussi que l'approche souffre de problèmes de dépendances pour les problèmes qui ont des expressions avec multiples occurrences d'une variable et aussi avec les problèmes contenant les expressions trigonométriques.

De la même manière, nous avons montré l'application de la formalisation à l'analyse des tolérances par un exemple d'assemblage hyperstatique avec une fonction de réponse implicite. La formalisation a été appliquée pour l'analyse statistique des tolérances par transformation avec l'aide de modélisation par torseurs des petits déplacements, la notion de « Compatibility Hull, Interface Hull et Functional Hull », simulation Monte Carlo et optimisation. L'algorithme et les résultats ont été validés.

Par conséquence, nous pouvons conclure que la formalisation de la maîtrise de la variation est bien harmonisée pour être intégrée dans la phase de la conception globale. Utilisant cette approche, nous pouvons traiter les problèmes de la conception robuste et l'analyse de tolérances. Cette approche peut être étendue vers d'autres phases dans le processus de conception. Pourtant, il existe des limites dans les outils utilisés pour les transformations des approches. Pour les outils d'arithmétique d'intervalles utilisés dans la transformation de la formalisation de la conception robuste, l'application a été limité à cause de problème de dépendance. De la même façon, le développement et l'amélioration dans la maîtrise des variables en termes de découpage et de méthodes de recherche aura un impact important sur efficacité de l'approche en termes de temps de calcul.

Part II

English Version

Chapter 1

Introduction

"In particular, an engineering design method must integrate the many different aspects of designing in such a way that the whole process becomes logical and comprehensible. To that end, the design process must be broken down, first into phases and then into distinct steps, each with its own working methods"-(Pahl & Beitz, 1996)

Product design is a core step in the product life cycle. It is also one of the most important phase in the life of the product as the decisions taken in this phase define different attributes of the product such as its performance, reliability, cost, aesthetics etc. A good product design can translate into the commercial success for the product and vice versa. This in turn can translate into profit or loss for the company producing it (Abouel Nasr & Kamrani, 2007).

Designing a mechanical product is a complex process which consists of many different steps starting from identifying the basic functional requirements to be fulfilled from the client specifications followed by an identification of the design parameters and the embodiment design. This phase is an important part of the design activity as in this phase the key design parameters as well as different possible design architectures of the product are researched and analyzed for design feasibility with respect of different viewpoints such as functional requirements satisfaction, economic feasibility and product reliability. The functional requirements are then translated into the product functional conditions and parameters. A preliminary product model is formed on the basis of this information. This model contains the key design variables whose values are to be calculated in such a way that they ensure the product performance according to the functional requirements. An initial model allows the designer to decide upon the feasible values of the design variables. These values are then further refined in the detailed design phase to ensure the product functionality, quality and robustness. The values for the design variables are usually based on the optimization of these variables with respect to the criteria set down by the functional requirements (Hans-Georg & Bemhard, 2007). This optimized solution may be able to ensure product performance under the circumstances pertaining to the optimization conditions. However, in reality there are variations and uncertainties present throughout the product life cycle which affect the product and therefore, are a cause of deviation in the product performance from the requirements set down by the functional requirements (Schuëllera & Jensenb, 2008).

A product that performs according to the functional requirements in spite of variations and uncertainties is called a robust product. The practice of designing a product with respect to variation is called robust design. Robust design step is normally placed as a downstream activity of embodiment design, where the parameters of the product design already decided, are tested and evaluated for their impact on the performance of the product. These evaluations then lead to the adjustment of the parameters to ensure a design with a required degree of robustness. This process instead of being a concurrent simultaneous process is more of an asynchronous quasi sequential process.

The management of the variation is an important step in the early design stage (during the embodiment or parameter design) as well as during the downstream design process. Mechanical products consisting of numerous sub-systems, with each sub-system consisting of many components, is a common thing in today's highly competitive, technologically advanced industry. A strict limit in terms of precision as well as adherence to the performance range is required. In order to meet these goals and to ensure the systems that assemble and work as required, an important verification and qualification step of tolerance design and analysis is integrated in the product design process. This step forms a core downstream variation management pillar in product design.

This thesis deals with the management of variation during the product design phase. The variation may be in terms of key design parameters from a point of view of product core design (falling within the paradigm of embodiment design and the associated variations) or, in the case of product dimensional variation, in terms of multi composite assembly and thus falling within the paradigm of tolerance design and analysis of complex systems.

The work presented in the scope of this thesis integrates the management of these two variation sources with the help of a common structure. This structure forms the basis of formalization of variation management within product design and is supplemented by detailed work in the analysis of the developed syntax and formalization. The developed formalization is then transformed into applications to general engineering design problem with the help of the selection and development of methodologies and tools.

This thesis is divided into three main parts. The first part deals with the development of a formal syntax that allows a designer to specify the needs and requirements of the clients. This is done with the help of abstract quantifiable expressions that outline the interaction of the design and variation space while keeping into account the functional conditions imposed by the clients. The second part deals with development of a theory based on this syntax, for the embodiment design of mechanical systems with simultaneous integration of variation management. The last part discusses a theory of integrated variation management in the downstream phase of tolerance analysis using the syntax developed.

1.1 Placement and Importance of Variation Management in Design Process

Variation is an unavoidable reality of engineering. Variation management during product design is a crucial part of any engineering activity and it plays a fundamental role in its success. Variation management includes the management of variation and uncertainty in the design process. Variation and uncertainty may arise in experiments, processing methods, and model parameters that support concurrent design of materials and products/systems. Potential sources of uncertainty in a system also include human errors, manufacturing or processing variations, variations of operating conditions, inaccurate or insufficient data, model assumptions and idealizations, microstructure variability, and lack of knowledge. Manufacturing variations are manifested as tolerances in part dimensions, as well as material and structure defects such as: pores, inclusions, cracks, or, variations of microstructure attributes such as phase distributions or volume fractions, grain sizes, and so forth. Operating conditions, such as the ambient temperature, environment, loading history, magnetic field, etc., may also vary (McDowell *et al.*, 2010).

The term uncertainty is used to describe incomplete information. Uncertainty maybe of two types, aleatory uncertainty or epistemic uncertainty (Rekuc, 2005). The sources of uncertainty in a design can come from uncertainty in (1) the parameters that control the design or the inputs to the design (aleatoric) and (2) the models used to relate these inputs to system performance measurements (epistemic). Parametric uncertainties can include the uncertainty in a size in the system, the cost of an item in the system, or the requirement on a performance measurement of the system (Martin & Simpson, 2006). Integrating uncertainty in the product design process ensures that the product function will not be adversely affected by an unforeseen change due to an uncertainty in one of the product parameter value. Taking uncertainty into account early on in the embodiment design is therefore essential to the reliability and robustness of the product performance and its insensitivity to the presence of uncertainty.

In the embodiment design phase, the search for the parameter design for a product takes place. This means that the detailed parameters of the product are decided upon and the values for these parameters are searched. These values are normally the nominal values of the product parameters and the product is expected to perform according to the functional requirements if the required value of these parameters is equal to the researched nominal values. In the real world however, variation and uncertainty both exist in parallel. It is not possible to attain an exact value for a given parameter due to limitations in the technology; uncertainty propagation exists through out the different phases of the design and manufacturing. The variation in the design parameters may cause the product performance to vary from the expected. This may have serious consequences on the product performance or the process in which the product is performing. The management of variation in the design parameters is therefore important to ensure that with variation in design parameters, the product performance still remains satisfactory.

Manufacturing or dimensional variation is a direct result of the manufacturing process and testifies to the fact that no part or assembly can be manufactured to an exact dimension. Two same parts produced on different machines or on the same machine will differ from each other, and in a similar manner, the assemblies resulting from such parts will differ from each other. It is therefore clear that a product that performs according to the functional requirements with nominal values of the parameter, may or may not performs in presence of variations. The margin of variation from the nominal values which can be tolerated while achieving the required functional characteristics is called *tolerance* (Whitney, 2004).

Managing the manufacturing variation is therefore an essential part of the design process. It should be addressed, from the outset of the product design process in order to ensure that the product has been designed while taking into account the variations of the manufacturing process. The manufacturing processes that are selected for manufactured product should be such that the product performance is as per the functional requirements.

In most of the design processes, management of uncertainty takes place in the downstream design phase after the parameter design has taken place. Once the values of the key parameters have been decided, the integration of uncertainty and variation is carried out in the process. This may be implemented via different optimization techniques such as sensitivity analysis, Pareto front or via statistical techniques such as Taguchi's design of experiments etc. Manufacturing variation management is integrated via tolerancing activity in the product design process. The tolerancing activity may consist of an iterative process consisting of two main steps of tolerance allocation and tolerance analysis.

1.2 Positioning of Work

Robust design and tolerance analysis are two important steps in the verification of a product design. According to Maropoulos & Ceglarek (2010), "currently, the verification and validation activities are usually executed when the design process is almost complete. This results in deviations from the required form, dimensions or function, extending development times and increasing the compliance cost. The problem is temporal as well as theoretical. The aim should be to execute verification and validation as early as possible during the design process, by developing new generation digital or virtual testing methods."

This thesis is oriented towards the variation management within the embodiment design phase of a product. It focuses on management and integration of design parameter variation and manufacturing variations in design phase. For this purpose, existing design processes are analyzed and extended with the use of the tools and methodologies from the domain of logic, constraint propagation, tolerance analysis and robust design to propose a solution that:

- Provides a formal framework for expressing mechanical design problems in terms of key design parameters and associated variations early on in the design phase.

- Permits variation management within the product design process through integration of design and variation space simultaneously in the product design model to deliver robust solutions.

- Takes into account the impact of the manufacturing variations on the product with respect to performance, assembly and quality according to the functional requirements.

- Provides the necessary tools and methods to transform the developed formalization to search the design and variation space for robust solutions.

- Permits horizontal flexibility in the design phase of product while integrating uncertainty and variation management.

The research in the scope of this thesis has been carried out in the "*Laboratoire Conception Fabrication Commande*" laboratory (LCFC). It extends and integrates the earlier research carried out in the same laboratory based on quantifier notion in the domain of tolerance specification, analysis and design by Dantan & Ballu (2002); Dantan *et al.* (2003b, 2005). Building upon this base, the quantifier notion developed earlier is generalized with the help of a logical expression developed to encompass a more generic notion of variation to integrate and model the variation behavior through out product design process.

This work also proposes a formalization to search for robust solutions during the embodiment design phase of a product. The robustness of the solutions proposed in this work are formed from the inherent nature of the design process selected in conjunction with the tools and methods developed for the integration of the variation management. These solutions, therefore, are inherently robust without the need of a downstream conventional robustness analysis tool. These tools will be developed and discussed in the later chapters.

The work is therefore positioned at the conjunction of the product design process theory, tools and methods of variation management in product design, and computational and mathematical tools necessary to realize the objective of integration of the above two, resulting into the final solution.

1.3 Layout of the Thesis

This research work is divided into three main parts: the formal framework for formalization description for the variation management; the expression and syntax for robust design and its applications; expression and applications for tolerance analysis. In order to present the research carried out, the thesis is divided into six distinct chapters.

The chapter no. 1 provides an introduction to the research work under the scope of this thesis.

Chapter no. 2 is dedicated to a detailed state of the art related to the main areas in the research work. The chapter is divided into three main parts. The first part provides an overview of the research work in the product design processes and the general process with respect to different design stages. This is followed by a discussion on different design processes from a decision based perspective in order to choose the basis for an appropriate design process for the presented variation management approach. The second part of the chapter is dedicated to the state of the art in the variation management within the product design process. This is further subdivided into a section dealing with the variation management through robust design and variation management through tolerance analysis. Both of these sections provide an overview of the research works in the fields of robust design and tolerancing during the product design. A brief and concise synthesis is provided at the end of the chapter to allow a perspective towards the state of the art and the needs arising, that form the basis for the choice of the necessary tools and methods developed and presented in this research work.

Chapter no. 3 presents the standard formalization and syntax for the variation management approach that is developed and presented in this research work. A structured formalization is presented that enables the variation management expressions in a uniform, expressive, mathematical and logical form. In order to develop this formalization, a logic based paradigm is used. Chapter three is divided in three parts. The first part discusses the modeling of product design and the variation management process as a constraint satisfaction problem and its general mathematical and logical expression. This is then translated into a formal structure that describes product design model in terms of constraints, variables and design domain. This formal structure is then refined in the next two sections to integrate the variation management in the design process through a formulation of set based robust design, and variation management through tolerancing by formulation of tolerance analysis.

Chapter no. 4 is dedicated to the discussion regarding the issues and modalities in applying the developed approach to the problem of variation management via robust design. This chapter explores the necessary methods, tools, and transformations to use the approach in an applicable manner to the generally occurring design problems to search for the sets of robust solutions. Eventually, this chapter also presents an algorithm developed to perform the application of the chosen tools and methods for robust design of systems. Finally the chapter presents different examples from engineering design to illustrate the application of the developed approach followed by discussion on the results and performance of these applications.

Chapter no. 5 is similar in its layout to chapter no. 4 and deals with the main task of the application of the developed formalization of the problem of variation management through tolerancing. In order to do so, the chapter discusses the methods and tools needed to transform the developed formalization in applicable form to the problem of tolerance analysis. Different tools to perform the tolerance modeling of assemblies are discussed in order to choose the appropriate tool. In addition to that, a strategy to allow the application of the developed formalization to the tolerancing problems is also discussed. Finally, the chapter presents an algorithm that has been developed for the application of the approach to an over constrained assembly for evaluation of applicability, performance and results of the developed approach.

Chapter no. 6 is the last chapter of the thesis. It presents a general discourse on the research work carried out with discussions on the capability and limitations of the approach as well as the future avenues of development, research and improvement in the research work presented.

Chapter 2

State of the Art

This chapter provides an overview of the product design process, different concepts pertaining to the design of the product, the design process from the operational and strategic viewpoints. The main focus of the thesis being the management of variation in the form of robustness and tolerance management, this chapter also discusses the state of the art in robust design of systems followed by the discussion of the tolerancing activity in the mechanical product design phase. The chapter concludes with a discussion on the pertinence of the existing tools and methods pertaining to the research objectives of the thesis and the need analysis for further development work.

2.1 Product Design Process

The product design process in general according to the different prevailing definitions can be envisaged as a goal oriented process that aims to provide a solution to a problem stated in terms of clients requirements that are achieved through decision making on behalf of the designer under a given set of constraints, stipulated by the clients requirements translated in terms of functional conditions of the products, availability of resources, capabilities at hand and the temporal and financial constraints. Many other similar definitions from a view point of process (McDowell *et al.*, 2010), parametric transformation (Suh, 2001), constraint propagation based design (Ullman, 2003) exist. Perhaps the most generic design statement comes from Gero who defines design activity as "a goal-oriented, constrained, decision-making, exploration and learning activity which operates within a context which depends on the designer's perception of the context". The fundamental objective in a design process is " to transform requirements - generally termed "functions", which embody the expectations of the purposes of the resulting artifact, into design descriptions " (Gero, 1990).

According to (McDowell *et al.*, 2010) "*Research in engineering design is categorized into design philosophies, models, and methods. Design theory is a collection of principles that are useful for explaining a design process and provide a foundation for basic understanding required to propose useful methodologies. Design theory explains what design is, whereas design methodology is a collection of procedures, tools, and techniques for designers to use. Design methodology is prescriptive, while design theory is descriptive. Design methods have been developed from different viewpoints that emphasize various facets of the overall design process. Some of these views, as summarized by (Evbuomwan* et al.*, 1996), include (1) design as a top-down and bottom-up process, (2) design as an incremental (evolutionary) activity, (3) design as a knowledge-based exploratory activity, (4) design as an investigative (research) process, (5) design as a creative (art) process, (6) design as a rational process, (7) design as a decision making process, (8) design as an iterative process, and (9) design as an interactive process. Although design methods are generally developed with a few of these viewpoints in mind, an ideal design method should support all of these.*"

Design process may be of sequential or concurrent nature (Abouel Nasr & Kamrani, 2007). All the design processes can however be divided into the following main stages: acquisition of the clients requirement; decomposition of the clients abstract requirements into a set of discernable functional requirements; search for initial solutions through initial design space exploration and discovery of key parameters; analysis and evaluation of initial search in a space to converge towards the solution to be retained subject to the fulfilment of functional requirements and other practical constraints; retention of the chosen design and its detailed design culminated into the design to be finalized.

The general layout of the design process as described above is found in nearly all contemporary design processes. A substantial amount of research has been carried out in the domain of product design. Many product design philosophies and methods exist which enable the product designer to follow a path through different stages of product design. These stages start from the translation of a client's requirements to the final manufacturing plans of the product. The following text provides an overview of current design processes.

2.1.1 Prevailing design models and their stages

A generic product development process starts with needs recognition and ends with the marketing of a finished product (Abouel Nasr & Kamrani, 2007).The major phases of this process are shown in figure 2.1.

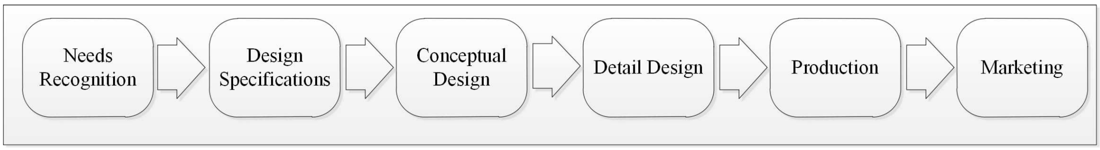

Figure 2.1: Generic Product Development Process encompassing Product Design Process (Abouel Nasr & Kamrani, 2007)

A rudimentary model of the design may be described as:

$$F \rightarrow D \tag{2.1}$$

Where F is the function required, D is a design description and $\rightarrow$ is some transformation (Gero, 1990). Using different methodologies and tools, this model may be explained in a detailed quantitative model. These models can be evaluated by finding the solutions that satisfy the required function. Decision making in design with quantitative models is discussed in detailed in later chapters.

From a systematic point of view by Pahl and Beitz, the aim of design is to have a comprehensive methodology for all the phases of design and development of technical systems. In line with the above aim, the product design life cycle may be broken down into four phases (Pahl & Beitz, 1996) which are:

- Product planning and clarifying the task
- Conceptual Design
- Embodiment Design

- Detail Design

Scaravetti (2004) provides a concise and illustrative comparison of the different popular design processes from an academic and industrial point of view. This comparison presented in figures 2.2 and 2.3 provides a general overview by decomposing where possible, the design processes into sub-stages and comparing them together with each other.

This comparison reveals that all the presented design processes fall well into the distinct stages of the product design process as shown in figure 2.1.

In addition to the processes defined above, there exist, other processes, whose steps are not similar and classifiable as the models presented earlier. These models include: the Axiomatic Design (Suh, 2001) (Benhabib, 2003), Design of experiments and Taguchi's Methods (Taguchi, 1978) (Taguchi, 1987) (Benhabib, 2003) (Phadke, 1989) and Set based design (Malak *et al.*, 2009). Axiomatic design is based on two axioms and their corollaries. The design of experiments and Taguchi's method emphasize the use of experiments and statistical techniques for the search of the parameter values. All of these design phases employ downstream validation and verification steps. In order to integrate variation management early on in the design phase (Maropoulos & Ceglarek, 2010), the verification and validation phase should be expressed early on.

2.1.2 Concurrent design process

The conventional design processes exhibit a sequential nature. This means long delays, rigid structure in decision making, repetition in modification due to sequential cycling within the different design interfaces. In today's highly competitive and global market, time to market has become a critical factor for a product. Designing a product in a short time so that it fulfils customers requirements at low cost and high quality determines the competitive advantage of a manufacturing company. The time required for completion of the product design process translates into the time required to bring the new product to market and thus a significant advantage of capturing the market share by the introduction of new products. Concurrent Engineering is one of the main concepts that allows the designers to achieve this. Mistree *et al.* (1990) define concurrent engineering as

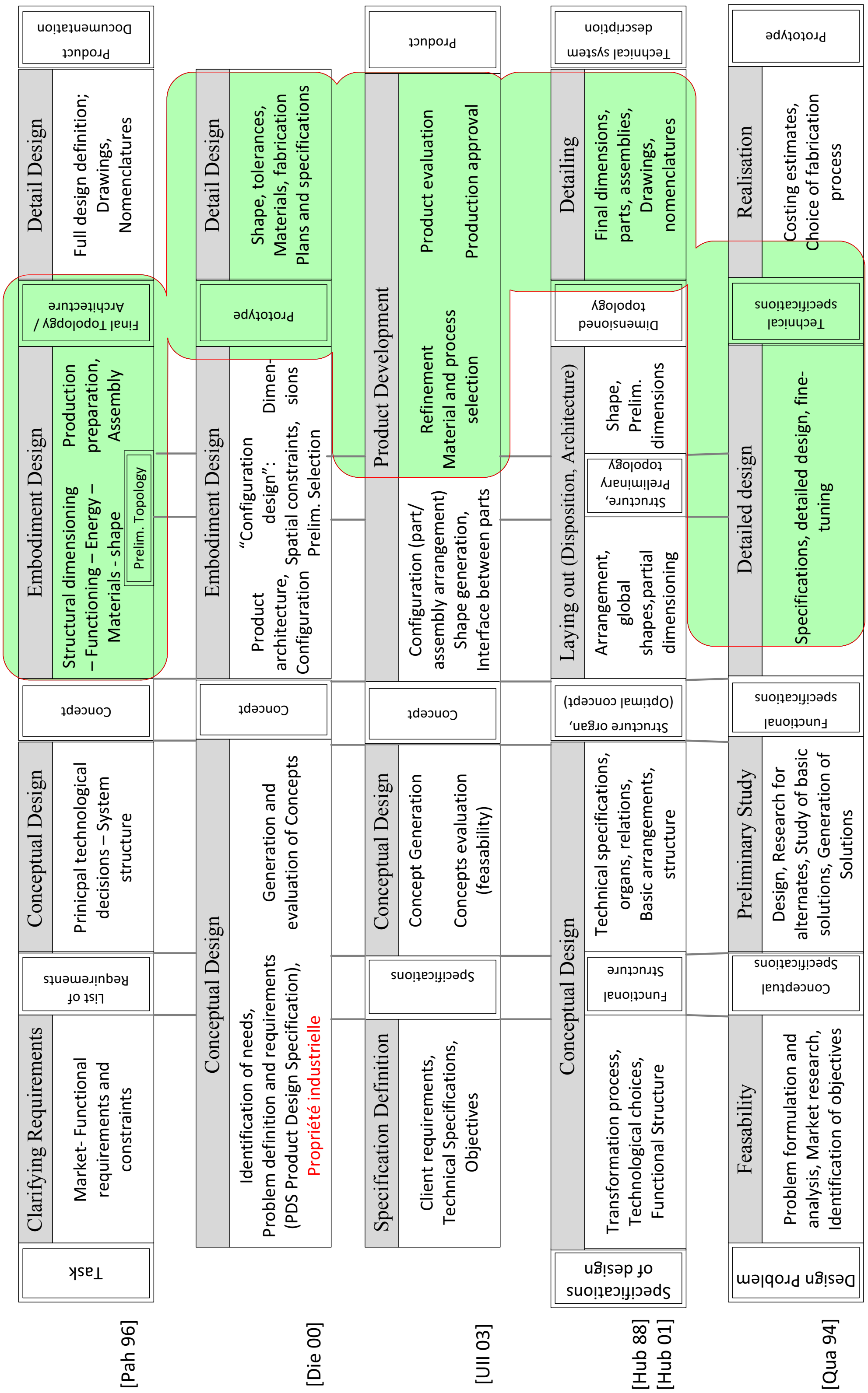

Figure 2.2: A parallel over view of prevailing design processes (Scaravetti, 2004)

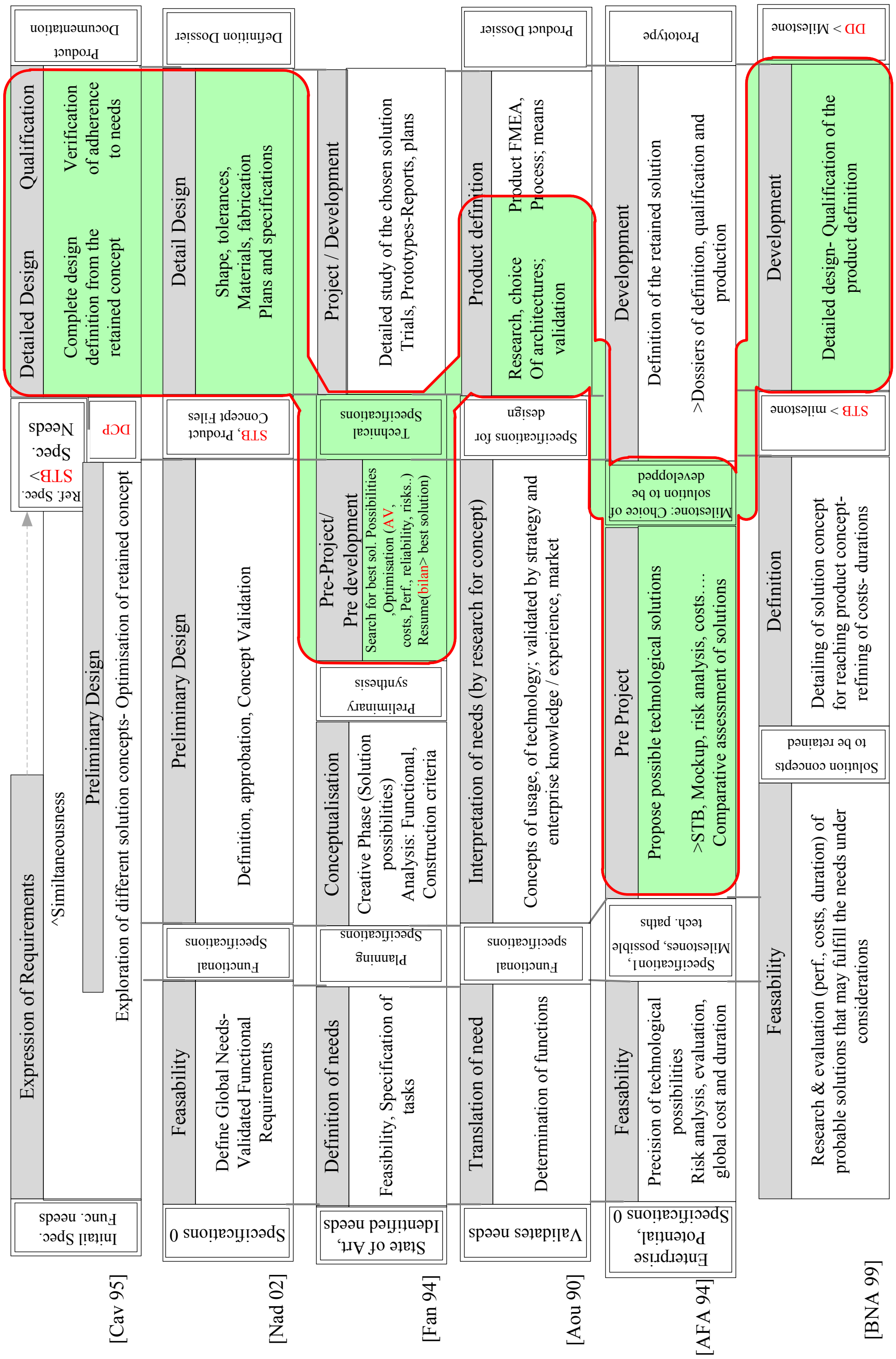

Figure 2.3: A parallel over view of prevailing design processes (Scaravetti, 2004)

a systematic approach to the integrated, concurrent design of products and their related processes, including manufacturing and support.

The basic guidelines for a concurrent design process have been defined by Benhabib as follows: (Benhabib, 2003)

- *Irreversible decisions should be delayed as much as possible, if they cannot altogether be avoided.*
- *Designs should allow "continuous improvement" based on potential future feedback.*
- *Product features should be analyzed with respect to manufacturability, assembly, and human factors.*
- *Product modularity, standardization, and interchangeability should be maximized where profitable.*
- *Product parameters should be designed in anticipation of imperfect use-"design for robustness."*
- *Production processes should be finalized concurrently with product design selection.*
- *Production plans and capacities should be in sync with marketing efforts and aim for short lead times (for delivery).*

Concurrent engineering includes designing for assembly, availability, cost, customer satisfaction, maintainability, manageability, manufacturability, operability, performance, quality, risk, safety, schedule, social acceptability, and all other attributes of the product. The cycle time in the concurrent engineering is greatly reduced due to over lapping of different activities. The decision making activity in concurrent engineering is highly interdependent in contrast to the decision making process in the sequential engineering. A general comparison between concurrent versus linear (serial) engineering as proposed by (Committee on Theoretical Foundations for Decision Making in Engineering Design, 2001) is shown in table 2.1.

Concurrent Engineering	Linear (Serial Engineering)
Parallel design of products and processes	Sequential design
Multi functional team	Independent designer
Concurrent consideration of product life cycle	Sequential consideration of product life cycle
Total quality management tools	Conventional engineering tools
All stakeholder inputs	Customer and supplier not involved

Table 2.1: A general comparison between concurrent versus linear (serial) Engineering (Committee on Theoretical Foundations for Decision Making in Engineering Design, 2001)

It can be concluded from table 2.1 that Concurrent Engineering offers a better integration throughout the product design process as opposed to Serial engineering. Serial engineering suffers from sequential and cyclic delays as well as issues in terms of quality due to the lack of simultaneous consideration of multiple factors and stake holders.

One of the main goals of the research work presented in this thesis is to integrate variation management in the product design at an early stage of design. This will enable integration of the factors which may have an impact on the product manufacturability, performance, cost as well as time to market at an earlier stage to avoid unnecessary delays and costs. This necessitates a design process that offers flexibility and concurrency to be selected for this research work.

2.1.3 Decision based design perspective

Every design process is essentially a sequence of decision based on the set of constraints. There can be numerous solutions that satisfy a given design problem. Therefore, decision making and judgement is required in order to select the appropriate design and to neglect the rest. This step is driven by decision based activities throughout the design process. According to Ullman (Ullman, 2003), "*design is the technical and social evolution of information punctuated by decision making*". It can be therefore concluded that the design methods used throughout the process adhere to the mathematics of decision theory. It is through the decision based design that both the algorithmic and iterative design (Suh, 2001) approaches progress to find the best possible design and avoid the worst designs.

Mistree (Mistree *et al.*, 1990) specifies that in decision based design (DBD), the engineer plays the role of a decision maker who takes the hierarchic decisions sequentially or concurrently using visual or analytical tools. These tools facilitate and shape the design process through different stages. The DBD framework is based on the concepts of rational decisions and decision making under uncertainty. The concept of rational decision prefers the options which are expected to have the highest values. Decision making under uncertainty is based on the fact that in engineering design, it is not possible to exactly know all the information related to the predicted product performance. Therefore, the decisions are undertaken in presence of risk and uncertainty (Gu *et al.*, 2002). Due to this inherent uncertainty and risk, the decision making activity cannot be carried out without human intervention. Hence, it is the responsibility of the design team to select the best alternative for the design after evaluating all the alternatives.

In terms of the hierarchy and concurrency of decision during a design process, the DBD can be divided into two main paradigms i.e.: Point based design and Set based concurrent engineering. It is therefore imperative to discuss the two different approaches in design.

2.1.3.1 Point based design

The Point based design is a generally prevalent design approach in which the design process advances through states known as points. Each point represents a decision point in the design process and provides results in terms of design improvement advances and retention of the best design. The information at that point is then transferred to the downstream functions of the next design stage. Generally, both algorithmic and iterative design approaches use successive evaluation loops and hit & trial methods to reach a Point based design in which a given solution is decided upon to be developed further in detail. The other parallel alternatives at that point are discarded in favor of the chosen concept (Uebelhart, 2006).

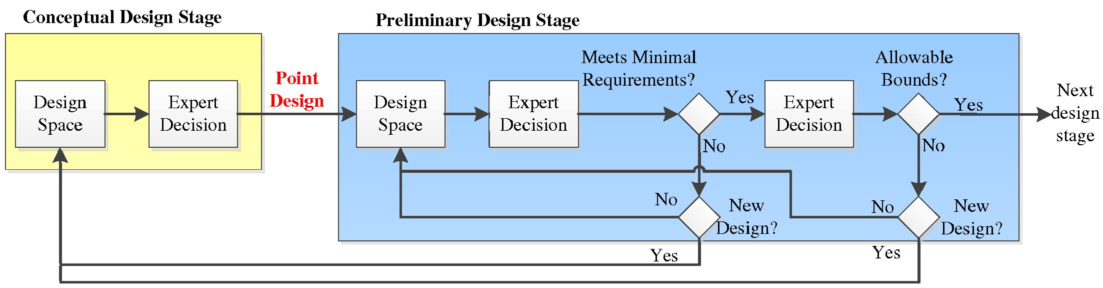

Figure 2.4: Conceptual and preliminary design stages based on a single point design (Uebelhart, 2006)

Figure 2.4 depicts the basic Point based design. This approach to design the products is appropriate in the situations where the computational effort required is too huge to be accommodated by the available computational power (Uebelhart, 2006). In the Point based design, only the selected configuration is refined and evaluated and the alternatives and parallels are left unevaluated.

A sizable body of research work is available in this field. The principal design processes employed for undertaking the decision based point design are systematic design (Pahl & Beitz, 1996), Function, Behavior, Structure (Gero, 1990), Axiomatic Design (Suh, 1990, 2001), or parameter design using mathematical framework (Committee on Theoretical Foundations for Decision Making in Engineering Design, 2001). The design process may be carried out using different mathematical tools such as mathematical programming, probabilistic techniques such as Pareto fronts or a preference structure, statistical tools such as Monte Carlo simulation (Yannou *et al.*, 2009), Design of Experiments and Response surface methods. A close relationship exists between decision making and optimization. In general, the best design option selected has been optimized with respect to one or many objectives, resulting in single objective or multi objective optimization (Gu *et al.*, 2002).

Due to being a point based design, the approach bears some advantages and some disadvantages. One of the advantages is that it helps in narrowing down the design space early on in the design process and helps convergence towards a given solution rapidly. On the other hand, the disadvantage of the approach is that at every point, vital information regarding the alternatives, the flexibility in terms of sets of possibilities is discarded, resulting in increasing rigidity in the design as the design process advances. This often results into high cost in the case of a modification at the downstream design phases. It also decreases the change management capability within the design process at a later stage resulting in the loss of pro-activity of the process. Also, a change made by one member of the team is likely to produce changes on others, these in turn producing more changes, in a chain reaction. There is no fundamental reason for the ensuing change process to ever converge (Chang & Ward, 1995). Developing a project on Point based design therefore is a poor design practice (Ullman, 2003). Use of such deterministic methods may result in systems which are expensive, inefficient and vulnerable to uncertainty and variability.

In order for design to be successful, such deterministic approaches should be replaced with new approaches that use rigorous models to quantify uncertainty and assess reliability (Nikolaidis *et al.*, 2008).

2.1.3.2 Set based concurrent engineering

The Set based concurrent engineering is an approach popularized by a Japanese automobile manufacturer. In this approach, instead of taking a Point based design approach, the designer takes a set based approach towards design and treats sets of design alternatives at both the conceptual and parametric design levels. These sets are gradually refined and narrowed through the process of elimination of ill suited alternatives until the emergence of the final design. In contrast to the Point based design, where one design is refined, the Set based concurrent engineering (SBCE) maintains the alternatives until the emergence of the final design.

Malak and Paredis define the Set based design as an approach in which different design alternatives are evaluated by reasoning and comparing different solutions based on possibilities offered by alternative possible configurations of "SETS" of design parameters. Set based design aims to delaying commitments to a particular design in favor of gathering information about the problem and reduce imprecision to levels at which indeterminacy is resolved (Malak *et al.*, 2009).

Ward explains the SBCE process as follows (Ward *et al.*, 1995):

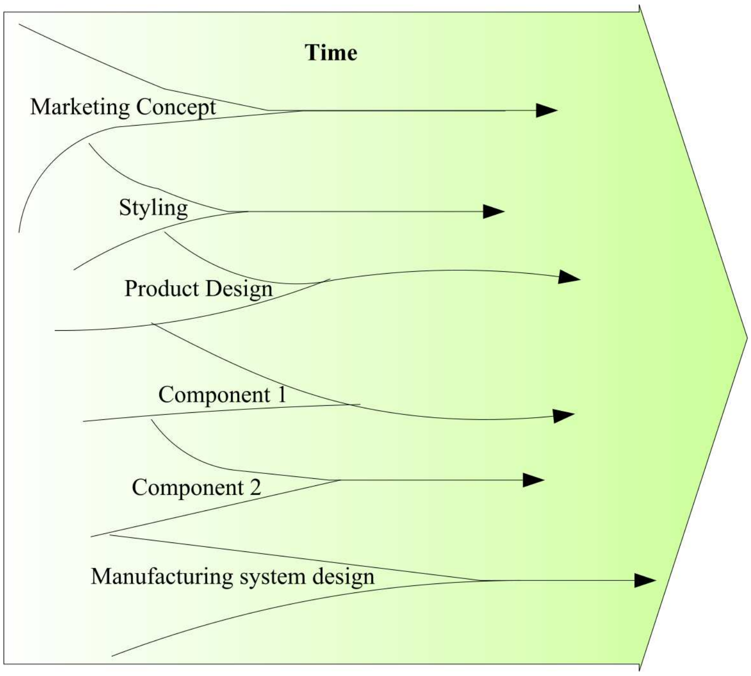

Figure 2.5: Parallel set narrowing process in SBCE sketched by a Toyota Manager (Ward *et al.*, 1995)

"In this approach, designers explicitly communicate and think about sets of design alternatives at both conceptual and parametric levels. They gradually narrow these sets by eliminating inferior alternatives until they come to a final solution. This approach contrasts with the common practice of iterating"— " making modifications or improvements in series "— " on one alternative until a satisfactory solution emerges. We call the iterative approach "point to point" design, since the state of the design moves from point to point in the "design space""

SBCE suggests exploration of sets of alternatives, but within clearly defined sets of constraints. Figure 2.5 shows the general approach of the SBCE graphically.

Rekuc (Rekuc, 2005) outlines the basic principles of SBCE as adapted from Sobek (Sobek *et al.*, 1999) as follows:

1. Define feasible regions

2. Communicate sets of possibilities

3. Look for intersections
4. Explore trade-offs by designing multiple alternatives
5. Impose minimum constraint
6. Narrow sets smoothly, balancing the need to learn with the need to design
7. Pursue high-risk and conservative options in parallel
8. Establish feasibility before commitment
9. Stay within sets once committed
10. Control by managing uncertainty at process gates
11. Seek solutions robust to physical, market, and design variation

A body of research explaining both the process and philosophy of the Set based design and the tools and methods employed to carry it out exist. The term Set-Based Design itself has been coined by Finch & Ward (1997). Early works in the field of catalog based mechanical design by Ward & Seering (1993a,b) use the notion of sets for selecting a design of mechanical systems. This system uses the branch and bound approach using constraint propagation for problem solving by exhaustive search through the design space to find the solutions that satisfy the given constraints. Being a catalog based approach, the results are the catalog numbers returned to aid in the selection of different parts for the mechanical system. This method narrows the design space by elimination, but does not provide a framework for decision making within the feasible space. Ward and Sobek present the theoretical and philosophical basis of Set based design and its application in the automobile industry as a practice (Liker *et al.*, 1996; Sobek *et al.*, 1999; Ward *et al.*, 1995).

The application of the set-based design through development of tools aiding in design has also been done by different researchers. Parunak *et al.* (1997) propose a market based framework implementing the Set based design process for a decision making system in terms of supply demand scenario. This work however does not specify the application methods for the framework. Using the same approach, Parsons *et al.* (1999) present a concurrent, Set based, systematic market approach based conceptual design for the design of a container ship which applies set-based principles to the collaborative key parameters of the design problem. Simpson *et al.* (1998) utilize techniques such as Response Surface Method (RSM) and Taguchi's method to implement a set based key parameter design for an aircraft. This work relies on RSM techniques to narrow down the design domain.

Finch & Ward (1997) develop and implement an inference method for Set based design of electronic circuits based on quantified relations (QR's), to explicitly represent casual relationships between variables in engineering systems, and present an interval propagation theorem (IPT). They also propose an algorithm that propagates intervals through QR's involving continuous and monotonic equations. This approach presents greater flexibility in comparison to the Labeled Interval Calculus (LIC) approach (Ward, 1989) presented by Ward.

Yannou *et al.* (2003, 2007) use constraint programming techniques over reals to carry out Set based dimensional design of mechanical systems while considering the sets of product key parameters. These methods however do not take into account simultaneous manufacturing variations. A specific syntax and framework for describing the interactions among the parameters also does not exist.

Yvars (2008) proposes a constraint satisfaction based design method for the design of a coupling system which takes into account a range of design parameters instead of point values and returns the result via calculation through a third party constraint solver.

Vareilles (2005); Vareilles *et al.* (2009) propose a Set based process design applicable to the heat treatment process of the metals via a Quad tree method. This method provides a decision support system for domain filtering while taking into account continuous and piecewise constraints. It aims to develop the tools for aiding in decision making for the process via help from developed propagation tools.

Sébastian *et al.* (2007) and Chenouard *et al.* (2009) present an application of the set based design process through the numerical constraint satisfaction problem method in conjunction with Constraint Explorer® software and heuristic based techniques. They determine the set of solutions for the starting domains of key parameters of engineering design problem. The problem is demonstrated with examples on heat exchanger and aircraft air-conditioning system with consideration of piece wise constraints. The problems presented however do not independently treat the noise and variation arising from independent sources such as manufacturing variations.

Malak *et al.* (2009) present a Set based approach to conceptual design using multi-attribute utility theory for mechanical systems. The approach uses multi-attribute theory for design evaluation and probabilistic and interval based modeling of associated uncertainty. The approach has been demonstrated through the example of a conceptual design of a fixed ratio vehicle transmission.

Bordeaux & Monfroy (2006) propose the theoretical basis for solving the problem of quantified variables, i.e. variables with existential and universal quantifiers, enabling to address the sets of values instead of point values for the problems where constraints are arbitrary relations over finite domains. Adopting the viewpoint of constraint-propagation techniques for CSPs, they provide a theoretical study of this problem and propose quantified arc-consistency as a natural extension of the classical CSP notion.

Considering the body of research present in the field of set-based-design, research can be divided into the sub categories of: theory of Set based design; methods and tools for Set based design; frameworks of Set based design. A significant body of research is available on the theory and tools for key parameter design of mechanical systems and general philosophy of set based design applied to the mechanical industry. Similar applications in the domain of artificial intelligence exist as well. However, little work is present in the domain of formalization and structuring of a syntax applicable to a generic design model.

Currently, the key parameter Set based design considers the sets of feasible key parameter space but does not simultaneously address the variations arising from other factors. This analysis is carried out in the downstream activities. There is little or no research work that provides a structured framework along with necessary methods and tools to carry out integrated Set based design of mechanical systems with robustness considerations.

2.2 Robust Design

The main objective in the robust design of a system is to design a system in such a way that its performance is not compromised beyond the minimum requirements in presence of variations and uncertainties in PLM (product life cycle) as well as different environmental and other changing parameters. These uncertainties and variations can be classified into: Changing environmental and operating conditions; production tolerances and actuator imprecision; uncertainties in the system output and feasibility uncertainties.

The robust design methodology was popularized by the works of Taguchi (Taguchi, 1987) and Phadke (Phadke, 1989) during the 80's and 90's who established robust design as a prime methodology for improvement in the quality and performance of the products. Taguchi's robust design objective is to reduce the output variation from the target by reducing the sensitivity to noise, such as manufacturing variations and deterioration over time (Maropoulos & Ceglarek, 2010). He suggested that "quality" should be thought of, not as a product being inside or outside of specifications, but as the variation from the target:

$$L(Y) = k(Y - T)^2 \tag{2.2}$$

Where L represents the quality loss of the product (in terms of any established indicator) with Y representing the measured performance parameter of the system in relation with T which is the target or the nominal value of Y and k is some constant. Essentially the equation describes the effect on the performance of the product in function of the displacement from the targeted mean.

Taguchi recommended a three step design process:

- System Design - The step that concerns the creation of a working prototype.
- Parameter design - Step involving the experimentation to evaluate the degree of influence of factors on the product performance. This step concludes by an analysis of the design model in context of the key influencing factors on the product performance.

- Tolerance Design- This step involves setting tight tolerance limits for the critical factors and looser tolerance limits for less important factors.

Another variation to the Taguchi's approach towards robust design is proposed by Chen *et al.* (1996). They define robust design as an improvement on the quality of a product by minimizing the effects of variation without eliminating the causes. The pertaining problems are dissociated into two broad categories according to the type of uncertainty and variations that they deal with:

- Type I - minimizing variations in performance caused by variations in noise factors (uncontrollable parameters)
- Type II - minimizing variations in performance caused by variations in control factor (design variables)

Chen *et al.* (1996) recommend a three step procedure based on Response Surface Model (RSM) and compromise decision support problem (DSP) to solve the problems of both Type I and Type II problems. These steps are as follows:

- Step1. Build response surface models to relate each response to all important control and noise factors using the Response Surface Methodology (RSM)
- Step 2. Derive functions of mean and variance of the responses based on the type of robust design applications
- Step 3. Use the compromise (Decision support problem) DSP to find the robust design solution

The contemporary robust design optimization process in general can be summarized as a work flow consisting of three different processes (Vlahinos *et al.*, 2003):

- The parametric deterministic model (PDM)
- The probabilistic design loop
- The design optimization loop

The PDM is the model that defines the product in a simulated environment. This model may be of mathematical, analytical or geometrical form in a suitable software platform. PDM defines the interaction between the variables and the constraints imposed. The probabilistic design loop is the part that simulates the effect of the uncertainties and variation in a model. These uncertainties may be of deterministic type, probabilistic type, or possiblistic type (Beyer & Sendhoff, 2007).

From a functional point of view, the methods of uncertainty propagation and estimation may be based on any of the following (Chen, 2007):

- **Local Expansion based methods**
  - Taylor series method
- **Simulation based**
  - Monte Carlo Simulation
  - Importance sampling, adaptive sampling
- **MPP (Most probable point based methods)**
  - FORM (First order reliability method)
  - SORM (Second order reliability method)
- **Numerical integration based method**
  - Full factorial numerical integration (tensor product quadrature)
  - Dimension reduction method
- **Functional expansion based method**
  - Polynomial chaos expansion method
- **Meta modeling based method**
  - (Kriging, Radial Basis Function, moving least square, etc.)
- **Non-Probabilistic Methods**

- Possibility-based (fuzzy set)
- Interval analysis
- Evidence Theory

The probabilistic design loop takes as an input, the parameters such as mean values and standard deviation values of the input parameters and provides the resultant probability distribution of the response attributes. The last step can then be used to optimize the results with respect to the given preferences resulting in an optimized robust design solution Vlahinos *et al.* (2003). This whole process can be seen in figure 2.6

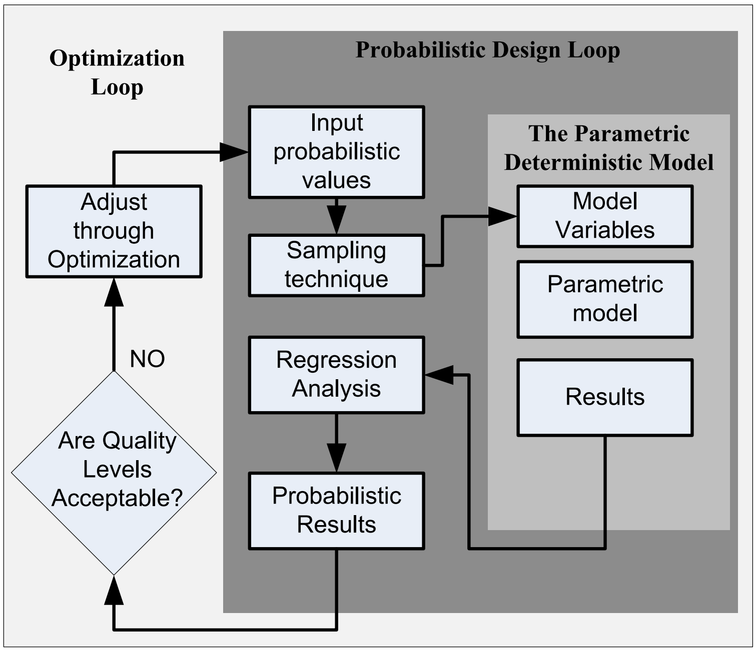

Figure 2.6: Work flow for robust optimization. (Vlahinos *et al.*, 2003)

It can therefore be concluded that robust design as described by Taguchi (1978) and Chen *et al.* (1996) is a process to de-sensitize the final design towards variations. These variations may be due to the uncontrollable environmental factors (Taguchi, 1987) or both the uncontrollable factors and the variation within the design parameters (Chen *et al.*, 1996). The general methodology of robust design process relies heavily on the probabilistic and stochastic processes and follows the general iterative procedure as proposed by Vlahinos *et al.* (2003). This general approach may be carried out by a selection of appropriate variation simulation (Chen, 2007), parametric modeling (Pahl & Beitz, 1996) and optimization tools (Hans-Georg & Bemhard, 2007; Schuëllera & Jensenb, 2008).

In general however, while using most of the techniques mentioned above, the integration of robustness takes place as a downstream function in the design process. Once, the nominal parameters of the design have been decided upon, the different techniques and methods mentioned above are employed to evaluate the robustness of the design. Based on these results, the design parameters are adjusted to the new values to integrate the robustness in design. This process is therefore remarkably similar to the Point based design methodology as explained by Sobek *et al.* (1999). This similarity points towards the repetitive and iterative nature of this process.

The notion of robust design in this thesis relies on the definition of "Type I and Type II" robustness provided by Chen *et al.* (1996) which deals with the variation related to the design parameters as well as the variations related to other uncontrollable sources. Therefore, in this thesis, the notion of robustness applies to the variations in the values of design parameters as well as the variation from uncontrollable and unforeseen factors such as variation in the material properties and the manufacturing process.

For the management of variation in the design process to assure the robust design of the system, the approach in this thesis diverges from the statistical and probabilistic methodology. Instead, it relies on Set based concurrent design process discussed earlier in section 2.1.3.2. This integrates the set based flexible design and takes into account the notion of variation to assure the robust design of the systems. The theory, practical methods and tools developed and utilized to enable such integration will be discussed in detail in chapters 3, 4 and 5.

2.3 Tolerancing

As technology improves and performance requirements continually tighten, the cost and the required precision of assemblies increase as well. There is a strong need for increased attention to tolerance design in order to enable high-precision assemblies to be manufactured at lower costs. As discussed earlier in 2.2, one of the challenges of robust design is to produce quality products that perform according to the functional requirements in spite of variations and uncertainty. One of main sources of uncertainty and variation in the case of mechanical systems consisting of mating or interconnected sub components is the variations arising from manufacturing processes. The ability to design components in presence of variation while keeping their ability to perform as per the functional requirements is the goal of tolerancing. Tolerancing is therefore an important part of the design process and a key component that must be addressed if the robustness of the product is to be ensured. It is therefore imperative that tolerancing activity be accounted for in the earliest possible phase of product design. Dantan *et al.* (2003a) propose early integration of the tolerancing process in the product design instead of its conventional place in the downstream process through Integrated Tolerancing Process (ITP).

Hong & Chang (2002) divide the field of tolerancing into seven fields:

- Tolerancing schemes
- Tolerance modeling and representation
- Tolerance specification
- Tolerance analysis
- Tolerance synthesis or allocation
- Tolerance transfer
- Tolerance evaluation

Tolerance schemes and tolerance modeling are concerned with the choice of tolerance representation and their modeling respectively. Tolerance specification is concerned about how to specify tolerance types and values. Tolerance analysis is the method to verify the proper functionality of a design, taking into account the variability of the individual parts. In order to perform tolerance analysis of a system, a model of the system is needed. This model can be constructed using different mathematical techniques and can be 1D, 2D or 3D (Hong & Chang, 2002). Performing tolerance analysis is an indicator of the quality of a product as it can be used to evaluate the assemblability of the system consisting of multiple sub components as well as the adherence of this system with respect to given dimensional functional characteristics. Therefore it can be concluded that tolerance analysis is a key element in the design process for improving product quality. To do so, a substantial amount of research has been devoted to the development of tolerance analysis. Tolerance analysis can be either worst-case or statistical (Chase & Parkinson, 1991; Hong & Chang, 2002; Nigam & Turner, 1995; Roy *et al.*, 1991).

Worst-case analysis (also called deterministic or high-low tolerance analysis) involves establishing the dimensions and tolerances such that any possible combination produces a functional assembly, i.e. the probability of non-assembly is identically equal to zero. It considers the worst possible combinations of individual tolerances and examines the functional characteristic. Consequently, worst-case tolerancing can lead to excessively tight part tolerances and hence high production costs (Hong & Chang, 2002; Roy *et al.*, 1991).

Statistical tolerancing is a more practical and economical way of looking at tolerances and works on setting the tolerances so as to assure a desired yield. By permitting a small fraction of assemblies to not assemble or function as required, an increase in tolerances for individual dimensions may be obtained, and in turn, manufacturing costs may be reduced significantly (Roy *et al.*, 1991). Statistical tolerance analysis computes the probability that the product can be assembled and will function under given individual tolerance.

The analysis methods are divided into two distinct categories based on the type of accumulation input: displacement accumulation or tolerance accumulation.

2.3.1 Displacement accumulation

The aim of displacement accumulation is to simulate the influences of deviations on the geometrical behavior of the mechanism. Usually, tolerance analysis uses a relationship of the form (Nigam & Turner, 1995):

$$Y = f(X_1, X_2, ...X_n) \tag{2.3}$$

where Y is the response (characteristic such as gap or functional characteristics) of the assembly and $X = \{X_1, X_2, \ldots X_n\}$ are the values of some characteristics (such as situation deviations or/and intrinsic deviations) of the individual parts or subassemblies making up the assembly. The part deviations could be represented by kinematic formulation (Desrochers, 1999), small displacement torsor (SDT) (Bourdet *et al.*, 1996), matrix representation (Gupta & Turner, 1993), vectorial tolerancing (Roy & Li, 1999)...

The function f is the assembly response function which represents the deviation accumulation. The relationship can exist in any form for which it is possible to compute a value for Y given values of $X = \{X_1, X_2, \ldots X_n\}$. It could be an explicit analytic expression or an implicit analytic expression. In a particular relative configuration of parts for an assembly consisting of gaps without interference between parts, the composition relations of displacements in some topological loops of the assembly permits to determine the function f. For an over constrained assembly, determination of the function f is very complex, whereas this determination is easy for an open kinematic chain.

For statistical tolerance analysis, the input variables $X = \{X_1, X_2, \ldots X_n\}$ are continuous random variables which enable the representation of tolerances. In general, they could be mutually dependent. A variety of methods and techniques (Linear Propagation (Root Sum of Squares), Non-linear propagation (Extended Taylor series), Numerical integration (Quadrature technique), Monte Carlo Simulation ...) are available for the estimation of the probability distribution of Y and the probability $P(T)$ with respect to the geometrical requirement (Nigam & Turner, 1995).

2.3.2 Tolerance accumulation

The aim of tolerance accumulation is to simulate the composition of tolerances i.e. linear tolerance accumulation, 3D tolerance accumulation. Based on the displacement models, several vector space models map all possible manufacturing variations (geometrical displacements between manufacturing surfaces or between a manufacturing surface and the corresponding nominal surface) into a region of hypothetical parametric space. The geometrical tolerances or the dimensioning tolerances are represented by deviation domain (Giordano, 1993; Giordano *et al.*, 2005; Teissandier *et al.*, 1999), T-Map® (Bhide *et al.*, 2007; Davidson & Shah, 2003), or specification hull (Dantan *et al.*, 2003a; Dantan & Ballu, 2002). These three concepts are a hypothetical Euclidean volume which represents all possible deviations in size, orientation and position of features.

For tolerance analysis, this mathematical representation of tolerances allows calculation of accumulation of the tolerances by Minkowsky sum of deviation and clearance domains (Giordano *et al.*, 2005; Teissandier *et al.*, 1999); to calculate the intersection of domains for parallel kinematic chain; and to verify the inclusion of a domain inside an other one. The methods based on this mathematical representation of tolerances are very efficient for tolerance analysis.

However, these two approaches do not take into account the quantifier notion. This notion translates the concept that a functional requirement must be respected in at least one acceptable configuration of gaps (existential quantifier “there exists”), or that a functional requirement must be respected in all acceptable configurations of gaps (universal quantifier “for all”) (Dantan *et al.*, 2003a, 2005). A configuration is a particular relative position of parts in an assembly featuring gaps without interference between parts. The current methods also do not take into account the gaps in a mechanism that have an effect on the performance of the mechanism. In presence of variations, each feature of all the individual components of a given assembly will exhibit departure from its ideal dimension. These departures from nominal will uniquely affect each assembly and each configuration resulting into a unique set of deviations and gaps arising from these variations. The evaluation of these deviations and gaps is necessary to establish the conformance of the assembly in terms of the cumulative effect of these variations on the performance of the product and its conformity to the functional conditions. For this purpose, it is necessary to quantify the deviations and gaps in a given assembly components. The quantifier notion impacts the result of the tolerance analysis (Dantan *et al.*, 2003a, 2005). Therefore, we propose a mathematical formulation of tolerance analysis which simulates the influences of geometrical deviations on the geometrical behavior of the mechanism, and integrates the quantifier notion.

2.4 Synthesis

Management of variation in product design holds a central position in the product design process. It is clear that the management of variation is treated mostly as a downstream activity. It is considered to be incorporated in the detailed design phase through the use of different tools and methodologies.

All the major design processes address the variation management through two major analysis and integration phases by using different tools. These phases are the robust design phase and the tolerance analysis phase. The robust design phase addresses the management of variation in the key design parameters with respect to the variation in the design variables. The tolerance analysis phase addresses the management of variation due to manufacturing or process related variations.

In order to put these phases into practice, there exist different types of tools and methods specific to each of the above phases. Generally, the robust design phase is contained as a downstream activity of the Point based design. In this activity, a solution deemed feasible is optimized for robustness through an iterative optimization process often based on statistical and or probabilistic tools. Depending upon the tools used for robust optimization, an engineer formulates the process directly. However, currently, no formal structure or syntax exists that allows the formulation and expression of the different key design parameters in a quantifiable manner such that a variation management problem can be expressed independently and in relation to the variables involved. In addition to this, a syntax or structure that directly relates the concept of robustness quantitatively to the design problem and allows simultaneous robust design in a concurrent engineering context does not exist. The same observation is true for the variation management phase of the mechanical system tolerancing.

It is therefore clear that a structured syntax that allows the fundamental expression of interaction between the design and noise variables while considering the constraints in an engineering design problem is necessary to bring an integration of variation in the design process at concurrent level. Also, in order to bring robustness through simultaneous management of type I and type II variations, it is necessary to distinguish and quantify the sources of variations that arise from the design variables and variations that arise from other sources. For this purpose, a structure is needed that not only distinguishes different variables and their mutual interactions but also is able to quantify them separately over a set of ranges. From a design process point of view, this is achievable by avoiding the Point based approach and following the Set based design approach.

However, in order to apply the Set based approach on a computable and expression level, a quantification strategy for the variables is to be selected. This is possible through the paradigm of mathematical and logical quantifiers in the framework of First order logic. Adoption of these quantifiers allow application of Set based design on an expressive level but require development and usage of specialized mathematical tools for their transformation in the form of executable solutions.

The next chapter provides the base for formalization in terms of a mathematical and logical framework developing an integrated and structured expression for variation management in the product design by expressing Set based robust design solution space exploration as well as tolerance analysis phase in a unified theory.

When viewed in terms of robust mechanical design, this translates into ensuring a choice of multiple design configurations, which may empower a design team to choose an appropriate solution while at the same time ensuring the robustness of the design with respect to the functional, as well as quality and cost constraints. When viewed in terms of tolerance analysis, this translates into ensuring the proper assembly and functioning of the mechanical system with respect to desired dimensional and functional characteristics, therefore minimizing the chance of failure and increasing the reliability of the product.

In terms of design decision making, it means that the different stake holders in the design process may be able to communicate their feasible sets at the start of the design process i.e. the design engineering may communicate the sets of design parameters where as manufacturing may communicate the sets of machining /

process capability. The resulting design space consisting of the design space and its corresponding variation parameter space can then be explored for their feasible regions. The intersections of the feasible design space can then lead to possible solutions and an intersection of the design and variation space would lead to solutions which may be inherently robust.

Chapter 3

Formalization For Variation Management

This chapter presents a unification to represent, manage and to integrate variation in the product design process. In order to successfully do so, a formal mathematical and logical formalization and structure is required that describes the interactions between different parameters and constraints in the design problem clearly. This formalization must be capable of quantifying the relationships among the parameters and allow the evaluation of the validity of possible solutions, or the lack of it. It should adopt the set based design approach and should be flexible to adopt ranges and sets instead of point based design. It should permit a progressive design solution validation in presence of variation, thereby, ensuring that the final solutions are inherently valid and robust. Eventually, the formalization should be uniform and homogenous enough to be applicable to variation management through robust design as well as tolerancing. In order to do so, this chapter builds upon the research work based on quantifier notion in the research by Dantan & Ballu (2002) and extends it with the help of the formal logic to develop an integrated formalization of variation management within the context of product design. For this purpose, this chapter provides an introduction towards the domain of logic, quantifiers and quantified constraints satisfaction problems, and proposes a unique formalization based on both of the above, for expression and resolution of the product design problem.

The chapter is divided into an introduction to the expression of product design in terms of constraint satisfaction, which lays down the basis for the construction of the product design expression in terms of logic and utilization of the quantifier notion. Once this has been explained, the main theoretical expression of the design problem within the logic framework with the help of quantifiers is developed. This is followed by the construction and adaption of the approach to the two different domains of variation management, i.e. variation management through robust design and through tolerance analysis of mechanism.

3.1 Formalization of Theory

As discussed in the Chapter 2, in decision based design, design process is based on taking decisions about the product design while keeping into account the constraints stemming from the client requirements related to the product (Mistree *et al.*, 1990). These constraints are then translated into the qualitative and quantitative domain.

A fundamental step at the heart of this process is to identify the client's requirements and associate these requirements in the form of key parameters to a corresponding design model. Also, once a design problem has been identified and converted into a model, a number of tools can be applied to find a solution to the problem by solving the design model for feasible values of the parameters involved. The tools involved in this step use different mathematical, experimental and scientific methodologies to solve the design model for the appropriate values of key parameters identified earlier for a satisfactory result.

However, another important phase in the design process is the phase that connects the two stages described above, i.e. the phase of expression and relation of the identified parameters and their associated domains. Very little work has been done in the domain of formal expression of the design problems that permits: a clear and formal description of the types of design parameters involved, their qualities, their domain ownership, their relationships, dependencies and interdependencies in a given design model that would allow seamless transition from the client requirement translation phase to the solution phases with minimum manual intervention.

3.1.1 Expression of product design in terms of constraint satisfaction

As discussed earlier, design is a decision making problem, in which the designer needs to take the decisions to move from one design step to another throughout the product design life cycle (Mistree *et al.*, 1990). These decisions are taken while considering different conditions specified by the client requirements. The client requirements are translated in the design process in terms of functional requirements. These functional requirements are the first level constraints that dictate the key requirements and product performance expectations. As the design process proceeds, these requirements are translated in a quantifiable form in terms of the key parameters and related design constraints. Key parameters are the main qualitative and quantitative parameters that have a profound effect on the product performance. These may be involved in univariate or multivariate relations that dictate the fundamental properties related to product physics and performance. The design constraints are seldom simple and explicit and are often complex, consisting of implicit multiple simultaneous requirements. The resulting set of parameters and related design constraints forms a mathematical model of the product design. All engineering design processes must, at some level, pass through this stage. The main objective of the designer, thus, is to focus on finding the feasible solution belonging to the design spaces for the key parameters such that the imposed constraints are satisfied. Using the above approach, the product design process can be transformed into a process of constraint identification and satisfaction.

Tsang (1993) defines a constraints satisfaction problem as: "*A CSP is a problem composed of a finite set of variables, each of which is associated with a finite domain, and a set of constraints that restricts the values the variables can simultaneously take. The task is to assign a value to each variable satisfying all the constraints.*"

In order to formalize and visualize the expression of parametric design, applicable to the context of the work presented, the generalization of the design problem in terms of a CSP as perceived by Yvars (2008) is adopted. This results in a generalized mapping from a three dimensional space $\{V, D, C\}$ to a set of solution designs $S = \{s_1, \ldots.., s_i\}$ where $V = \{v_1, \ldots..., v_j\}$ is a set of variables, $D = \{d_1, \ldots, d_j\}$ is a set of domains of variables and $C = \{c_1, \ldots, c_k\}$ is a set of constraints .

The set of variables V includes all the variables associated to the design problem. Unique values of these variables propose a unique solution for the given design problem. The set of domain D constitutes the space of domains of all the variables in the set V such that for each variable $v_i \in V$, there is as corresponding domain $d_i \in D$ which describes the starting search space for the variable v_i. The set of constraints C includes all the constraints resulting from the translation of the client requirements into corresponding functional requirements and their further translation in terms of fundamental mathematical relations governing the design of the product. Each c_i is a constraint in the form of a relation between the variables or parameters.

For a successful design solution, the goal is to determine the proper values for the design variables which satisfy the design constraints. The design variables may be of the geometric nature, engineering nature or manufacturing nature and may deal with shape, configuration, material, manufacturing process etc. The design constraints are generally expressions consisting of the design parameters, constants and variables. In an engineering model, the representation of constraints may be algebraic equations or predicates, sometimes with a few additional logical constraints (Beyer & Sendhoff, 2007). Essentially, it can be viewed as solving a system of simultaneous equation for a homogenous solution. Different iterative methods for solving such systems exist which may be generalized in the category of Constrained Satisfaction Problems (CSP). Existing methods to solve constraint based design problems include symbolic manipulation, demand-driven evaluations, and numerical methods, each of which have their qualities and limits (Thornton, 1996).

Design process can be modeled in the form of a constraint satisfaction problem (CSP) and has proved its efficiency within the framework of single-designer design. Many engineering design software packages now have some sort of built in support and features to aid the designer by integrating the constraints satisfaction subsystems within the softwares (Thornton, 1996). Thornton (1996) also presents a Constraint Satisfaction algorithm using Genetic algorithms and

Simulated annealing heuristic techniques for the solution of a simple mechanical design problem. Yvars (2008) presents an algorithm for the CSP resolution of a gear train pre-dimensioning problem with the help of object oriented constraint modeling and a commercial CSP solver. The presented approach deals with continuous and mixed CSPs for the resolution of instantiated values of solution but has limited capability with the models with differential expressions. Yvars (2009) extends the functions of CSP to the context of multi-concept design of the same artifact by proposing cooperative constraint satisfaction problem (CoCSP). Yannou *et al.* (2003) propose the use of meta modeling techniques to carry out approximated constraint programming based design for engineering design problems with help of real CSP, design of experiments (DOE) and Response surface method (RSM) for systems in which analytical relations describing product behavior are not available and therefore need to be approximated. Further work by Yannou explores the engineering design (by using constraint satisfaction problems in the context of constraint programming) in conjunction with search algorithms for design of simple mechanical systems, its downstream optimization for robustness with respect to variation and uncertainty, using techniques such as Taguchi's Design of Experiments (DoE) Response Surface, Pareto front and probabilistic methods (Yannou *et al.*, 2007, 2009).

Constraint satisfaction using traditional methods as a stand alone technique or in conjunction with other downstream optimization tools is pertinent in the case of deterministic design. In case of the management of the variation and uncertainty, discussed in section 2.4, to integrate and unify the variation management within the product design, a gap exists in the expression and formalization of variable interaction. To address this gap, we extend the existing research in the domain of the variable quantification in tolerancing by Dantan & Ballu (2002) for assembly specification as well as other further research (Dantan & Qureshi, 2009; Dantan *et al.*, 2005). These works provide an example of variable quantification and expression for the problem of variation management in tolerancing. They are based on the existential and universal mathematical quantifiers. The work presented in this chapter generalizes the concept of these quantifier in the framework of the constituent logical framework i.e. First Order Logic (FOL).

Adopting the paradigm of FOL allows quantification of variables as well as necessary set of symbols, predicates and relations that enable building a formal syntax for the expression, evaluation and deduction of the product design process with the integration of variation management.

The following text in the chapter is dedicated to an introduction to the basis of logic followed by the development of a theory for an application to the problem of robust design and tolerance analysis.

3.2 Main Theoretical Basis in Terms of Logic

Hurley (2008) defines Logic as the science of evaluating the arguments. Logic has been used since the ancient times in Greek philosophy and mathematics. Numerous works of Plato include the use of logic to establish the syntactic relation between principles and their verity or falsity. The first known systematic study in Logic inference was carried out by Aristotle and assembled by his students in 322 B.C. based on Aristotle's Syllogisms known more popularly as inference rules. Since then Logic has developed over the period of time to encompass the fields of Philosophy, Mathematics, Psychology, Cognitive behavior, and recently in the last century towards applied the fields of Artificial intelligence, Computer science and Engineering.

In engineering, popular application of Logic remains in the fields of electronics engineering as well as in mechanical and industrial engineering, where under the sub domain of industrial and manufacturing engineering, operations research and planning, logic is implemented in applied form in the shape of different programming and algorithmic techniques from its sub domain, artificial intelligence.

Most of the work done in the domain of Mechanical and industrial engineering is of applied nature and no formal logical syntactic expression is used to describe or formalize an engineering problem. Instead, the Algorithmic and artificial intelligence routines are used directly in most of the cases to resolve a problem according to a given set of constraints, putting the practice near the domain of applied constraint satisfaction or constraint programming with the help of search routines.

A problem may be expressed in logic via classical bivalent logic such as syllogistic logic, propositional logic or via First order logic (Hurley, 2008), or multi-valued logic such as ternary logic or fuzzy logic. Bivalent approach provides the two valued (”TRUE” or ”FALSE”) validation of a proposition where as multi-valued approaches provide up to infinite different possibilities. In terms of bivalent approaches, earlier research in the formalization of the mechanical design problems with the help of predicate logic were done by Finch & Ward (1997). This work proposed an initial predicate logic based mechanical engineering design proposition. In the field of multi valued approaches, numerous applied research works exist in the field of control theory as well as some applications in the field of mechanical design. Massa *et al.* (2009) propose a multi-objective optimization approach for mechanical structures based on the principles of fuzzy logic. This being a multi-valued logical possibilistic approach, presents the optimization of the design problem of a structure with some variations to the design parameters and, in conjunction with genetic algorithm, proposes a design space exploration for possible solutions. This approach deals with the type I uncertainty with the help of the fuzzy set notation by using Fuzzy Finite Element method in conjunction with the fuzzy design variables.

The research work presented in this thesis formulates the problem of variation management with the help of formal logic. As discussed earlier, formal logic is a general extension of the quantifier notion used in the earlier existing works by Dantan *et al.* (2005). For the scope of the research in this thesis, a bivalent logical paradigm is assumed sufficient to provide the basic consistency for the validation of the design space. FOL provides the essential integration of quantifier notion, as well as an associated set of symbols, operatives, and rules for treating the problem of bivalent evaluation, and combines the distinctive features of syllogistic and propositional logic with the added capabilities of quantification in predicates. We formulate the problem using the standard notion of first order logic as defined in Table 3.1. A detailed description of the symbols used is given in appendix A.1.

The basic syntactic elements of First-order logic (FOL) are connectives, quantifiers, predicates, functions and variables. In this work, the basic notation of the first order logic is used to formulate the problem of expression and integration of

Sentence	$\longrightarrow$	Atomic Sentence
	$\mid$	(Sentence Connective Sentence)
	$\mid$	Quantifier Variable,.... Sentence
	$\mid$	$\neg$Sentence
AtomicSentence	$\longrightarrow$	Predicate(Term,...) $\mid$ Term=Term
Term	$\longrightarrow$	Function(Term,...)
	$\mid$	Constant
	$\mid$	Variable
Connective	$\longrightarrow$	$\Rightarrow \mid \vee \mid \wedge \mid \Leftrightarrow$
Quantifier	$\longrightarrow$	$\forall \mid \exists$
Constant	$\longrightarrow$	$A \mid X_1 \mid Length \mid ...$
Variable	$\longrightarrow$	$a \mid x \mid l \mid ....$
Predicate	$\longrightarrow$	$Variation \mid HasMaterial \mid HasFeature \mid ....$
Function	$\longrightarrow$	$Tangent \mid Perpendicular$

Table 3.1: Syntax of First-order logic specified in Backus-Naur form. Adopted from (Russell & Norvig, 2003)

variation during different stages of engineering design (Robust design and Tolerance analysis) in a syntactic form. A framework is proposed to find a computational solution to the problem with algorithms suited to the resolution of these expressions. In order to proceed further, it is necessary to introduce the basic concept of quantifiers and quantified constraint satisfaction problems.

3.2.1 Quantifiers

One of the restrictions of the earlier logic description systems such as propositional logic was the inability to expressively generalize a basic concept or a relation that holds for a specific condition over a class of variables and situation. Propositional logic allowed only instantaneous description which meant that in order to fully describe or express a situation or a condition an exhaustive list of description had to be written. The idea behind the quantifiers in FOL relates to the idea of variables that could be bound by certain operators. Since the birth of logic, the idea of operators which allow quantification of the variables has been expressed in different forms starting from Aristotle. The generalized quantifiers emerged in the later 19th and earlier 20th century and were extensively used in fields varying from linguistics to computer science and logic. Two types of quantifiers are existential quantifiers $\exists$ and universal quantifiers $\forall$. In order to grasp the basic concept behind the quantifiers in FOL logic model, the following definitions may be adopted as presented in (Peters & Westerstahl, 2006): A logic is based on a language (syntax), a model or interpretation $\mathcal{M}$ and truth relations between sentences of the language and models. In this case, the language is FOL which contains the quantifiers, the model or interpretation is the corresponding interpretation of the engineering systems and the relations are the sentences whose verity is required for the system to function. Let

$$\mathcal{M} = (M, A, B) \tag{3.1}$$

where $A, B \subseteq M$, be a model for an unspecified vocabulary with two unary predicate symbols, sometimes even using "A","B" for these symbols as well. Then, the definitions to follow for the existential quantifier $\exists$ and universal quantifier $\forall$ is given by table 3.2

Quantifier	**Linear (Description)**
$\forall$	$all_M(A, B) \Leftrightarrow A \subseteq B$
$\neg\exists$	$no_M(A, B) \Leftrightarrow A \cap B == 0$
$\exists$	$some_M(A, B) \Leftrightarrow A \cap B \neq 0$
$\neg\forall$	$notall_M(A, B) \Leftrightarrow A - B \neq 0$

Table 3.2: Description of Quantifiers (Peters & Westerstahl, 2006)

The quantifiers used through out the work follow the descriptions given in the table 3.2. The next section explores the quantified constraints satisfaction problems.

3.2.2 Constraint satisfaction problem (CSP) limitations

As discussed earlier, the aim of variation management is to take into account at the earliest possible stage, the different uncertainties and variations within the product life cycle. Therefore it is essential that the product design model be capable of representing uncertainties and variations. The representation of the product design problem as a CSP enables the simultaneous representation of the product design variables and constraining parameters but is limited in terms of representation of incomplete or uncertain data. In order to account for a change in a specific design requirement, a product design problem in terms of CSP needs to be revalidated each time there is a change in the variables. This necessity limits the applicability of CSP towards robust design. In addition to the aforementioned points, CSP is limited in terms of search space validation and exploration over a continuous interval i.e. in order to account for the validity of a specific solution over an interval of changing values of variables, a more general and flexible framework is required. It is therefore useful to have a method which might integrate the uncertainty in the system or in the environment at an early design stage. A methodology is being proposed which not only integrates the notion of uncertainty in the product, but also enables the quantification of the variables. Such an approach is based on the integration of the quantifiers described earlier with the CSP which results in a Quantified constraint satisfaction problem (QCSP) .

3.2.3 Quantified constraint satisfaction problem (QCSP)(Verger & Bessiere, 2006)

The quantified constraint satisfaction problem (QCSP) is a general extension of the constraint satisfaction problem (CSP) in which variables are totally ordered and quantified either existentially or universally (Chen, 2004). This generalization provides a better expressiveness for modeling problems by allowing the universal quantification of the variables. For each possible value of such variables, the values have to be found for remaining, existentially quantified, variables so that all the constraints in the problem are satisfied. The QCSP can be used to model PSPACE-complete decision problem from areas such as planning under uncertainty, adversary game playing, and model checking (Gent *et al.*, 2005). QCSPs find their application in the fields from video game design to manufacturing problems.

A standard definition of QCSP as defined by Gent *et al.* (2005) follows: A Quantified Constraint Satisfaction Problem (QCSP) is a formula of the form QC where Q is a sequence of quantifiers $Q_1 v_1, ...Q_n v_n$ where each quantifier Q_i ($\forall$ or $\exists$) quantifies a variable v_i and each variable occurs exactly once in the sequence.

C is a conjunction of constraints $(c_1 \wedge .... \wedge c_k)$ where each c_i involves some variables among $(v_1, \ldots, v_j)$.

The semantics of a QCSP QC can be defined recursively as follows. If C is empty, than the problem is true. If Q is of the form $(\exists v_1 Q_2 v_2 .... Q_n v_n)$ then QC is true iff there exists some value $a \in D(v_1)$ such that $Q_2 v_2 .... Q_n v_n C[(v_1, a)]$ is true.

The research in QCSP is recent. Bordeaux & Monfroy (2006) have extended the notion of Arc consistency of CSP to the QCSP. Mamoulis & Stergiou (2004) have defined an algorithm for arc consistency for QCSP for binary constraints. Dantan & Qureshi (2009) have proposed the integration of QCSP and quantifiers in the domain of product design by solving the problem of product assembly, tolerance analysis and tolerance allocation for mechanical assemblies with QCSP (Dantan & Ballu, 2002; Dantan & Qureshi, 2009; Dantan *et al.*, 2005). Qureshi *et al.* (2010) propose a robust design methodology for mechanical systems using Quantifiers, QCSP and set based design.

Based on the principles of set based design, formal logic, quantifiers and QCSP, we now propose a general theory of variation management within engineering design with the help of tools the mentioned above.

3.3 General Theory

Considering the expression of design in terms of CSP as mentioned in 3.1.1, we can summarize the general design problem in terms of CSP as:

$$\{V, D, C\} \tag{3.2}$$

Where V, D, C defined earlier are:

$$S = \{s_1, \ldots .., s_i\} \tag{3.3}$$
$$V = \{v_1, \ldots ..., v_j\} \tag{3.4}$$
$$D = \{d_1, \ldots ., d_j\} \tag{3.5}$$
$$C = \{c_1, \ldots ., c_k\} \tag{3.6}$$

$S = \{s_1, \ldots .., s_i\}$ is the set of acceptable solutions where each individual s_i is an individual solution such that $s_i \in D$.

Using the basis of formulas and sentences in terms of formal logic (Peters & Westerstahl, 2006), the following definitions are made:

1. If P is an n-ary predicate symbol in V and $v_1, ..., v_n$ are variables, then $P(v_1, ..., v_n)$ is a V-formula.

2. If v_i and v_j are variables, then $(v_i = v_j)$ is a V-formula.

3. If φ and ψ are V-formulas, then so are $\neg\varphi, (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi)$ and $(\varphi \leftrightarrow \psi)$.

4. If φ is a V-formula and v_i a variable, then $\forall v_i \varphi$ and $\exists v_i \varphi$ are V-formulas.

5. If φ is a V-formula, and V_φ is the set of non-logical symbols occurring in φ then, $V_\varphi \subseteq V$ and V_φ is a V_φ-formula.

The cruicial sementic relation for the design model to be true then becomes:

$$\mathcal{D} \models \varphi(a_1, ...a_2) \tag{3.7}$$

Where $\mathcal{D}$ is the model for a vocabulary V, $\varphi = \varphi(v_1, ...v_n)$ is a V-formula containing at most the variables $v_1, ...v_n$ free and $a_1, ...a_n \in D$ are assigned to $v_1, ...v_n$ respectively. Alternatively, the satisfaction relation may also be written as:

$$\mathcal{D} \models \varphi[f] \tag{3.8}$$

which holds between a model, a formula, and an assignment f of values in M to the variables such that in particular $f(v_i) = a_i, 1 \leq i \leq n$. For the sake of simplicity, substituting $\bar{a}$ for the sequence $a_1, ...a_n$, the definition of the satisfaction of the resulting model in terms of existential and universal quantifiers can then be stated respectively as:

$$\mathcal{D} \models \forall v \varphi(v, \bar{a}) \quad \Leftrightarrow \quad \text{for all } b \in D, \mathcal{D} \models \varphi(b, \bar{a}) \tag{3.9}$$

$$\mathcal{D} \models \exists v \varphi(v, \bar{a}) \quad \Leftrightarrow \quad \text{for some } b \in D, \mathcal{D} \models \varphi(b, \bar{a}) \tag{3.10}$$

$$\mathcal{D} \models (\varphi \wedge \psi)(\bar{a}) \quad \Leftrightarrow \quad \mathcal{D} \models \varphi(\bar{a}) \text{ and } \mathcal{D} \models \psi(\bar{a}) \tag{3.11}$$

Having defined the above relations, considering a model with two interpretation V-formulas φ and ψ we can define the following relations for the solution:

$$\mathcal{D} \models (\varphi \wedge \psi)(\bar{a}) : \bar{a} \in D \Rightarrow s_i \models (\varphi \wedge \psi)(\bar{a}) \tag{3.12}$$

$$\mathcal{D} \models \exists V(\varphi \wedge \psi)(V_{(\varphi \wedge \psi)}, \bar{a}) : \bar{a} \in D \Rightarrow s_i \models \exists V(\varphi \wedge \psi)(\bar{a}) \tag{3.13}$$

$$\mathcal{D} \models \forall V(\varphi \wedge \psi)(V_{(\varphi \wedge \psi)}, \bar{a}) : \bar{a} \in D \Rightarrow s_i \models \forall V(\varphi \wedge \psi)(\bar{a}) \tag{3.14}$$

The above expressions lay the concrete foundations for the application of the quantifiers to the problem of engineering design. In the next sections we will use the basis developed in this section for an application to two specific steps in engineering design, i.e. the theory of robust design and the theory of tolerance analysis.

3.4 Formulation of Set Based Robust Design

The main goal of robust design is to design a product so that its performance remains acceptable even in the presence of variations and uncertainty. Conventionally, robust design is considered to rely upon the statistical approaches based on Taguchi's principles and the associated tools and methods. This by far remains to be a popular approach. There are other methods and concurrent definitions for attaining robustness in design. The research work in this thesis takes the general definition of robust design per Chen as discussed in section 2.2. We consider a design to be robust if it is able to effectively integrate the type I and type II variations in the design process and maintain the performance to a desirable level.

Keeping in view the above definition, the work in this thesis achieves robustness in the design by integration of type I and type II variations in the product design phase. This is achieved by using set based design, variable quantification through universal and existential quantifiers and logical expressions, to evaluate a design space, in the presence of constraints defining a product model, for a solution. The goal in the robust design approach presented here is to systematically search the design space and identify a set of solutions using the logical expressions, such that in presence of type I and type II variations, the minimum performance remains acceptable as stipulated by the constraints.

In order to extend the developed basis for the variation management in the set based design so as to assure a robust design of the mechanical systems, the semantics of the model described above are to be defined. Therefore, in consistence with the approach developed in section 3.3, we define the fundamental parts of the FOL which are Vocabulary, Interpretation and Universe. The corresponding terms in QCSP are Variables, Constraints and Domain. This is shown in table 3.3.

FOL Framework	Linear (QCSP Framework)	
Vocabulary	$\Longleftrightarrow$	Variables
Interpretation	$\Longleftrightarrow$	Functions
Universe	$\Longleftrightarrow$	Domain

Table 3.3: Corresponding terms in FOL and QCSP framework

In the following text we will lay down the foundation for the terms shown in table 3.3

3.4.1 Variable definition for Robust Design

V represents the set of all variables in the robust design problem. The elements of V, which are individual variables, are denoted by v . The types of these variables present in a design problem are governed by the constraints which are imposed on them. The variables in the context of this problem may be categorized into symbolic, numeric, discrete, continuous,or design or noise variables, depending on the point of view. From a design point of view, the variables may be categorized as design variables or noise variables.

The symbolic variables should have a defined semantic. These are generally used to define a list of symbols. These variables may be considered as discrete variables as they normally take the choice of a symbol within a list of symbol. An example of symbolic variable in the context of the problem being discussed is the variable representing the choice of a specific material among the list of different material names e.g. a specific material among a given list of materials (Aluminium, SS 304, SS146. . . .).

The numerical variables cannot have semantic. They can be further decomposed into discrete numerical variables or continuous numerical variables. The discrete numerical variables may be of the integer or real type or of the intervals over the integers and an example in this respect may be the choice of the number of bolts in a flange coupling. The continuous numerical variables are continuous variables limited by real intervals. The variable for the thickness of a beam would be a continuous numerical variable bound by its domain e.g. $t \in [0.1, 0.6]$. Continuous variables are represented by a small case italic letter.

From a design perspective, the variables may be decomposed into design variables or noise variables. The design variables are the variables which determine the key design dimensions or key parameters. The thickness, width and length of a beam in a beam design are examples of design variables. The noise variables are all those variables which determine the variation in the design problem. The examples of noise variables include the error in a specific dimension, or variation percentage in a specific material property. Manufacturing variation related to a specific dimension is also an example of noise variable. For illustrative purposes, the set V is further subdivided into two sets; the set of design variables which denotes the set of key parameters, and the set of noise variables. Mathematically, this can be described as:

$$V = DV \cup \Delta \tag{3.15}$$

$$DV = \{v_1, ..., v_n\} \tag{3.16}$$

$$\Delta = \{\delta_{v_1}, ..., \delta_{v_n}\} \tag{3.17}$$

3.4.2 Domain

The set D consists of the sets of domains of the variables V. It is useful to mention all the different views of the set D that will be utilized. As the set D contains the initial domains of all the variables at the start of the design problem, therefore it will be titled Design Domain as this space provides the starting point for the set based solution exploration space. Definitions will follow.

Domain D is a vector containing the starting sets for each element of the set V. The elements of D are conditioned as per the requirements imposed on them by the class of variables. It also represents the universe of FOL. As domain is the universe for the model, therefore all necessary information related to the possible values of the Variables (Vocabulary) must be contained within the domain. Domain is subdivided into design domain and noise domain.

Design Domain For the problem of robust design, we define Design domain D_{DV} as a vector containing the starting sets for each element of the set DV. The elements of D_{DV} are conditioned as per the requirements imposed on them by the class of variables. It also represents the possible sets of the key parameters. As design domain is the universe for the key variables in the model, therefore all the necessary information related to the possible values of the these variables (Vocabulary) must be contained within the design domain.

Noise Domain For the problem of robust design, we define D_Δ as a vector containing the starting sets for each element of the set Δ. The elements of D_Δ are conditioned as per the requirements imposed on them by the class of variables. As noise domain is the universe for the noise variables δ_i, therefore all necessary information related to the possible values of the such variables (Vocabulary) must be contained within the noise domain.

Mathematically this is represented as:

$$D = D_{DV} \cup D_\Delta \tag{3.18}$$

$$D_{DV} = \{d_{v_1}, ..., d_{v_n}\} \tag{3.19}$$

$$D_\Delta = \{d_{\delta_{v_1}}, ..., d_{\delta_{v_n}}\} \tag{3.20}$$

3.4.3 Constraints

Constraints are an important part of the engineering design and provide the fundamental qualitative and quantitative boundaries to restrain the domain within the necessary requirements. Although constraints used during design include explicit and implicit relations, heuristics, tables, guidelines, and computer simulations, a majority of those used can be expressed as mathematical constraints (Thornton, 1996). We define three types of constraints relative to our problem which are:

Discrete constraints The constraints where a choice between two or more discrete entities is required.

Continuous constraints The constraints where all the variables involved and parameters belong to the domain of real and continuous numbers.

Mixed constraints Constraints involving both the discrete and continuous variables.

Relational constraints Constraints based on relational database necessitating the use of rule based database selection.

Complex constraints Constraints involving the use of expressions dealing with complex numbers; i.e. numbers having real and complex parts..

The constraints are expressed in explicit or implicit relations of variables belonging to the set V and may of the n-ary form, expressed with an equality or inequality. The general form of the constraints may be:

$$C = \{c_1, ...c_n\} : \tag{3.21}$$

$$c_i = f(\bar{x}) : \bar{x} \in V, d_x \in D, f(\bar{x}) \in \mathbb{R}^\infty \tag{3.22}$$

$$c_i = f(\bar{x}) : \bar{x} \in V, d_x \in D, f(\bar{x}) \in \mathbb{C} \tag{3.23}$$

$$c_i = \bar{x} : x \in R\{R|r_1, r_2, r_3...\} \tag{3.24}$$

$$c_i = \begin{cases} IF..... & THEN.... \\ ELSEIF.... & THEN.... \end{cases} \tag{3.25}$$

$$c_i = \begin{cases} f(\bar{x}) \leq 0 \\ f(\bar{x}) \geq 0 \\ f(\bar{x}) = 0 \end{cases} \tag{3.26}$$

3.4.4 Quantifier based expression for the robust set based design exploration

We define a robust solution set s_i of design as a design which performs as required in the presence of variation. It is defined as:

"A solution s_i is a robust solution s_{robust} if it satisfies the imposed constraints for all the values of possible assignments contained within its domains and accounts for each variable $v_i \in V$ for variation $\delta_{v_i} \in \Delta$ over the totality of assigned domains $v_i \in d_{v_i}$ and $\delta_{v_i} \in d_{\delta_{v_i}}$."

The mathematical translation of the above statement requires the use of the universal and existential quantifiers and its expression requires the syntax of FOL. The existential quantifiers are used to condition the variables in set V so that each variable (design or noise) is conditioned according to it's requirement. An explanation of a valid design set and robust design in terms of its mappings of V, D and C can be described as follows:

Valid Set In case of a valid design, there must exist values for all variables from the design domain $v_i : \{v_i \in DV, d_{v_i} \in D_{DV}\}$ such that all the constraints c_i in the set C are verified.

Robust Set In case of a robust design, for all values of parameter variables from the parameter domain, $v_i : \{v_i \in DV, d_{v_i} \in D_{DV}\}$ while keeping in account all possible values of noise variables $\delta_{v_i} : \{\delta_{v_i} \in \Delta, d_{\delta_{v_i}} \in D_\Delta\}$, all the constraints c_i in the set C are verified.

In order to address the above described definitions for the valid set and robust set, we need to transform the tuple V, D, C with the help of the integration of the quantifiers as described in Section 3.2.3. Therefore, we integrate the general design problem described with QCSP, proposing the mathematical model of the system based on QCSP in the following generalized form:

$$\{QV, D, C\} \tag{3.27}$$

Where:
QV= Set of Quantified variables. Each quantified set is a set of the form $Q_1x_1, .., Q_nx_n$, where Q_i denotes a quantifier (existential "$\exists$" or universal "$\forall$") and x_i is any variable that is to be quantified.
$QV = QDV \cup Q\Delta$
QDV= Set of Quantified design variables
$Q\Delta$ = Quantified noise or uncertainty variables.

The emerging system is a quantified constraint satisfaction problem. It can now be used to formalize the conditions for a valid design and a robust design in terms of quantified variables. For this purpose, the following text presents the formal expressions for the evaluation of the design space for the existence of a solution as well as for the solutions that satisfy the robustness criteria.

3.4.5 Consistency evaluation for solutions

Having described the concepts of variables, domains and constraints, now we propose the condition for the consistency of existence of solution and existence of robust solution. Using the $\{QV, D, C\}$ notation in conjunction with the syntax of FOL, as discussed earlier and the basic concepts of hull and box consistency (Benhamou *et al.*, 1999; Bordeaux & Monfroy, 2006; Chenouard, 2007; Mamoulis & Stergiou, 2004), we propose the following two conditions for the consistency of a solution. The first condition deals with the consistency of existence of a solution for a set. The second condition performs the consistency for the existence of a robust solution for the sets validated by the first condition.

3.4.5.1 Existence of a solution

For any design solution that attempts to satisfy the needs of a given design problem, the first test is the ability to satisfy the core requirements of the clients. In terms of a product model, it means that the solution is able to satisfy the fundamental constraints imposed by the translation of the client's requirements in terms of a model. These constraints are often the threshold functional requirements which define the success or failure of the product.

In terms of the logical and mathematical structure proposed, this translates in terms of set based design as evaluating if there exists a successful intersection of different interface sets allowing to converge towards a robust solution. Therefore, a solution s_i may be a valid solution if:

"There exists a solution s_i belonging to the set of solutions S such that at least one configuration of design variables with their assign-

ments belonging to their respective domain must exist and the functional requirements are fulfilled"

It can be translated mathematically as:

$$\exists s_i \in S : s_i = \mathcal{D}_i \models (D, C) \tag{3.28}$$

$$s_i \models \exists \bar{v} C(\bar{v}, \bar{a}) \tag{3.29}$$

Equation 3.29 translates the basic requirement for evaluating the design space D with respect to the constraints C for consistency of existence of a solution. Therefore, if for an assignment $\bar{a}$ of values assigned from the design space, the constraints are valid for the existence of a solution, then a valid solution exists.

3.4.5.2 Existence of a robust solution

The robust design aims to assure that the product performance remains acceptable under the influence of type I and type II variations. Therefore, the second condition deals with the expression and evaluation of a solution that is robust. For a valid solution that promises to fulfill the constraints for a given assignment of values to the variables involved, in order to assure that the performance will hold in the presence of variation, it is necessary to quantify the design variables in a way that in spite of the variation in their values, the constraints should be satisfied. This quantification can then address the type I variation in the design process. In addition to the type I variation, other sources of variation need to be added to the design problem in order to account for the variations resulting from outside factors such as manufacturing variations. For this purpose, another class of variables has been defined earlier: the noise variables need to be quantified and included in the expression to integrate the type II uncertainty. This is done by integrating the noise variables in the example.

Using the above approach, the second condition for the existence of a robust solution can then be described to be that there must exist a solution satisfying the constraints for all the values of design variables within their domains while keeping in account all possible values of noise variables within their domains. This can be defined as:

"There exists a solution (robust) belonging to the set of solutions such that for all possible assignments of the values of design parameters belonging to their respective domains and for all possible assignments of noise variables belonging to their respective domains, the constraints must be respected".

This can be mathematically translated as:

$$\exists s_i \in S : s_i = \mathcal{D}_i \models (D, C) \tag{3.30}$$

$$s_i \models \forall \bar{v} \forall \bar{\delta v} C(\bar{v}, \bar{\delta v}, \bar{a}) \tag{3.31}$$

A solution s_i that fulfills the above two conditions is a robust solution. Using the above two conditions, it is possible to apply the set based design space exploration that takes the starting design space as an input, and which explores this space by quantifying the design space existentially and universally in form of sets of involved variables to return the regions of feasible intersections which are inherently robust and insensitive to the variations within the regions validated for robust solution. The QCSP formulation has also been developed and applied in earlier research works as stipulating conditions for assembly and functional condition verification of mechanical components for 2D and 3D tolerance analysis applications (Dantan & Qureshi, 2009).

Using the above two conditions, it is possible to deduce the validity of a given design space for the solution. In order to apply these logical expressions, it is necessary to explore the application of the resolution strategy of the above framework. This has been addressed in chapter 4.

3.5 Formulation of Tolerance Analysis

Using the syntax developed in the section 3.3, this section will present the formalization of tolerance analysis. As discussed in section 2.3, tolerancing is an important downstream phase of product design, especially in the case of products with multiple components and assemblies. This work generalizes and extends the earlier research carried out in the field of tolerance synthesis, assembly specification and virtual gauge by Dantan *et al.* (2003a, 2005). Using the mathematical existential and universal quantifiers, they simulate the influences of geometrical deviations on the geometrical behavior of the mechanism. This simulation takes into account not only the influence of geometrical deviations but also the influence of the types of contacts on the geometrical behavior of the mechanism. Their approach translates the concept that a functional requirement must be respected in at least one acceptable configuration of gaps (existential quantifier there exists), or that a functional requirement must be respected in all acceptable configurations of gaps (universal quantifier for all). The theory presented here generalizes the quantifier based tolerance expression into a logical expression of tolerance analysis for mechanical assemblies. In order to formalize the problem, we proceed by adopting the semantics of the generic model described earlier in table 3.3 in terms of tolerance analysis.

3.5.1 Variable definition for Tolerance analysis

V represents the set of all variables in the tolerance analysis problem. The geometrical definition by Dantan & Ballu (2002) has been adopted for tolerance analysis problem. This definition necessitates the definition of the variables related to the nominal dimensions and their corresponding variations/deviations. Also, the definition calls for means to express the gaps in a mechanism and the function characteristics that are set as a requirement of the product performance. These are defined as follows.

Situation Deviations The situation deviations define the orientation and position variations between a substitute surface and the nominal surface. The situation deviation space is denoted by symbol Sd

Intrinsic Deviations The intrinsic deviations of the substitute surface are specific to their type. They define the surface variations. For instance, the intrinsic variation of a substitute cylinder is the radius variation between the substitute cylinder and the nominal cylinder. The intrinsic deviation space is denoted by symbol I.

Gaps The gaps define the orientation and position variations between two substitute surfaces in contact and are denoted by symbol G

Functional Characteristics The functional characteristics define the orientation and position variations between two substitute surfaces in functional relation. The space of functional characteristics is denoted by symbol Fc

Table 3.4 represents the symbols in a tabular form.

Space Type	**Symbol**	**Designation**
Situation	Sd	Situation deviations of parts
Intrinsic	I	Intrinsic deviation of parts
Gap	G	Gaps between parts
Functional Characteristic	Fc	Functional characteristics between parts

Table 3.4: Spaces in tolerance analysis problem (Dantan & Ballu, 2002)

Having defined the appropriate symbols ,we now mathematically define the vocabulary for the tolerance analysis problem. We define the set of vocabulary/variables as V. V consists of sets of Situation and Intrinsic deviations, Gaps and functional characteristics. Therefore mathematically this is represented as:

$$V = DV \cup Sd \cup I \cup G \cup Fc \tag{3.32}$$

$$DV = \{v_1, ..., v_n\} \tag{3.33}$$

$$Sd = \{sd_1, ..., sd_{n1}\} \tag{3.34}$$

$$I = \{i_1, ..., i_{n2}\} \tag{3.35}$$

$$G = \{g_1, ..., g_{n3}\} \tag{3.36}$$

$$Fc = \{fc_1, ..., fc_{n4}\} \tag{3.37}$$

These sets will be discussed in detail in the section related to the formalization of the theory of tolerance analysis.

3.5.2 Domain

The universe or domain D for the tolerance analysis problem includes the possible assignments for the members of the vocabulary V. These assignments play an important role in the analysis problem as they contribute to or control the search for the design solution. These assignments include the values that the variables in the functional characteristics space can take on as established from the client requirements or as needed by the different constrains. The domain also includes the assignment values for the variables related to the nominal dimensions.

3.5.3 Constraints

The interpretation functions or constraints for the tolerance analysis problem are based on the expression of the geometric behavior of the mechanism. The vocabulary for the geometric behavior has been described in the variables section. Using these variables, it is possible to establish the constraints/interpretation functions that form the evaluation knowledge base of the model. Three different types of constraint families have been defined based on the interaction of the defined spaces which are termed hulls. These hulls are:

Compatibility Hull The relations between the small displacements of surfaces of parts (Dantan & Ballu, 2002) lead to the compatibility hull. Composition relations of displacements in the various topological loops express the geometrical behavior of the mechanism (Ballot & Bourdet, 1997; Dantan & Ballu, 2002; Dantan *et al.*, 2005; Soderberg & Johannesson, 1999). The composition relations define compatibility equations between the situation deviations and the gaps. The set of compatibility equations, obtained by the application of composition relation to the various cycles, forms a system of linear equations. Since the system of linear equations admits a solution, it is necessary that compatibility equations are checked. These compatibility equations characterize some hyperplanes in the $Situation \times Gap \times Functional\ characteristic$ space. The group of constraints resulting from the compatibility hull is denoted in the following text by $H_{compatibilty}$

Interface Hull The constraints of contacts between parts surfaces nominally in contact lead to the interface hull. Interface constraints limit the geometrical behavior of the mechanism and characterize the non-interference or association between substitute surfaces, which are nominally in contact (Dantan &

Ballu, 2002; Giordano, 1993) Roy & Li (1999). These interface constraints limit the gaps between substitute surfaces. These constraints define the interface hull in *Gap* $\times$ *Intrinsic space*. In the case of floating contact, the relative positions of substitute surfaces are constrained technologically by the non-interference, the interface constraints result in inequations defined in *Gap* $\times$ *Intrinsic space*. In the case of slipping and fixed contact, the relative positions of substitute surfaces are constrained technologically in a given configuration by a mechanical action. An association model exists for this type of contact; the interface constraints result in equations defined in *Gap* $\times$ *Intrinsic space*. The group of constraints resulting from the Interface hull is denoted in the following text by $H_{interface}$

Functional Hull The functional constraints between part surfaces in the functional relations lead to the functional hull. The functional requirement limits the orientation and the location between surfaces, which are in functional relation. This requirement is a condition on the relative displacements between these surfaces. This condition could be expressed by constraints, which are inequalities. These constraints define the functional hull in *Functional characteristic* $\times$ *Intrinsic space*.The group of constraints resulting from functional hull is denoted in the following text by $H_{functional}$

The mathematical form of these constraints is in terms of linear or non-linear expressions involving members of V. The relations may be of type equality or inequality. The relations coming under the compatibility Hull $H_{compatiblity}$ are in the form of linear equations where as the relations from interface hull and functional hull ($H_{interface}$ and $H_{functional}$) are in the form of inequality or equality.

Mathematically they may be expressed as:

$$C = \{H_{compatibility}, H_{interface}, H_{functional}\} : \tag{3.38}$$

$$H_{compatibility} = \{c_{comp_1}, ..., c_{comp_j}\} \tag{3.39}$$

$$c_{comp_i} = f(\bar{x}) = 0 : \bar{x} \in V, f(\bar{x}) \in \mathbb{R}^{\infty} \tag{3.40}$$

$$H_{interface} = \{c_{int_1}, ..., c_{int_k}\} \tag{3.41}$$

$$c_{int_i} = \begin{cases} f(\bar{x}) \leq 0 \\ f(\bar{x}) \geq 0 \end{cases} : \bar{x} \in V, f(\bar{x}) \in \mathbb{R}^{\infty} \tag{3.42}$$

$$H_{functional} = \{c_{fonc_1}, ..., c_{fonc_l}\} \tag{3.43}$$

$$c_{fonc_i} = \begin{cases} f(\bar{x}) \leq 0 \\ f(\bar{x}) \geq 0 \end{cases} : \bar{x} \in V, f(\bar{x}) \in \mathbb{R}^{\infty} \tag{3.44}$$

$$\tag{3.45}$$

3.5.4 Quantifier based expression for the Tolerance Analysis for Mechanical Assemblies

The objective of the mathematical formulation for the tolerance analysis problem is to define the necessary constraints on the deviations of each part, i.e. the spaces Sd and I. The previous geometrical behavior description and the formalization with the help of FOL and quantifiers enable defining the admissible deviations of parts such that the functional requirement is respected. These admissible deviations form a hull in situation and intrinsic spaces called the specification hull. To define it, we formalize a textual relation and a mathematical relation between various hulls (Dantan & Ballu, 2002; Dantan *et al.*, 2005).

In order to generalize the problem in terms of FOL, the general flow of the tolerance analysis problem at the decision level is discussed. Via the tolerance analysis, the probability of assembly of a given set of mechanical components is calculated. For the sake of understanding, this process can be divided into three simplified main steps.Step one is concerned with the identification of the concerned variables in the problem i.e. vocabulary and relationships that govern the translation of the assembly in a mathematical model (the Interpretation functions), the second step is the verification of the model via the values in the universe and the third step is the presentation of the results over the number of different attempts of validation of the universe. This is depicted in figure 3.1.

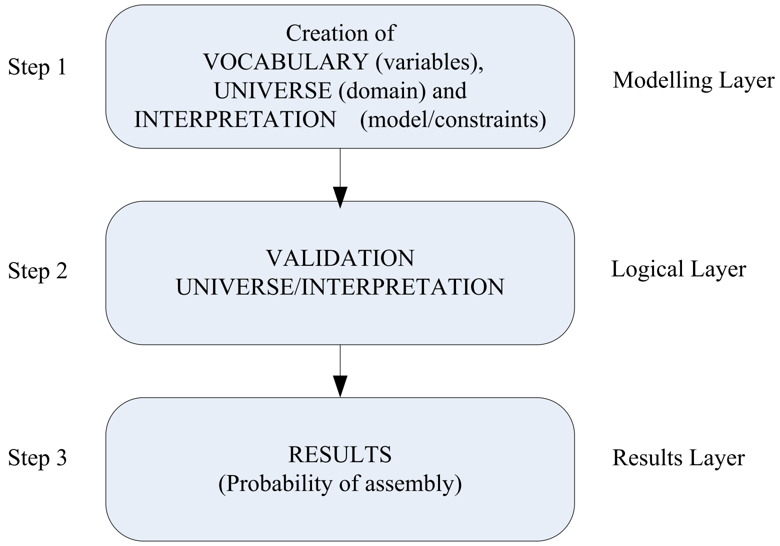

Figure 3.1: Generalized Break Down of Tolerance Analysis Problem

The requirements pertaining to step one in the figure are addressed by the geometric behavior model through which the vocabulary is developed and through the Inference functions and constraints that form the model. The step two however consists of logical steps in which the instances of values from the domain are validated with the help of the developed model. This is a two step process consisting of evaluating the assemblability of the mechanism and the respect of the functional conditions.

These steps can be generalized in the logical form as follows:

3.5.4.1 Respect of assemblability of the mechanism

A mechanism is a set of components in a given configuration with each components having deviations and the gaps that result through the given assembly configuration of components. In order for a mechanism to assemble successfully, the different components in the presence of deviations should assemble without interference and should have a specific set of gaps that characterize the instance of the assembly. An acceptable solution s_g can then be defined as a solution that allows the assembly which validates the existence of gaps with values from the universe such that the constraints related to the assemblability are satisfied. This condition stipulates the use of an existential quantifier for an initial search for the existence of a feasible configuration of gaps. Therefore using the existential quantifier, the solution s_g is defined as:

"the deviations are admissible" is equivalent to "there exists an admissible gap configuration of the mechanism such that the assembly requirement (interface constraints) and the compatibility equations are respected".

It can be translated as:

$$\begin{aligned} \exists s_g \in S : s_g &= \mathcal{D}_{assembly} \models (D, H_{Compatibility} \cap H_{Interface}) \\ s_g &\models \exists G \; H_{Compatibility} \cap H_{Interface}(V, \bar{a}) : \bar{a} \in D \end{aligned} \tag{3.46}$$

3.5.4.2 Respect of functional requirements

The condition of the assemblability in 3.5.4.1 describes the essential condition for the existence of gaps that ensure the assembly of the components in the presence of part deviations. Once a mechanism assembles, in order to evaluate its performance under the influence of the deviations, it is necessary to describe an additional condition that evaluates its core functioning with respect to the basic product requirements. In terms of tolerance analysis, the basic requirement becomes the maximum or minimum clearance on a required feature that would have an impact on the mechanism's performance.

The most essential condition therefore becomes that for all the possible gap configurations of the given set of components that assemble together, the functional condition imposed must be respected. In terms of quantification needs, in order to represent all possible gap configurations, the universal quantifier is required. The second condition therefore implies the universal quantifier "$\forall$" . Therefore, the solution s_{Fc} is defined textually as:

"The deviations are admissible" is equivalent to "for all admissible gap configurations of the mechanism, there exists a functional charac-

teristic such that the geometrical behavior and the functional requirement are respected".

This may be written as:

$$\begin{aligned} \exists s_{Fc} \in S : s_{Fc} &= \mathcal{D}_{Fc} \models (D, C) \\ s_{Fc} &\models \forall G \exists Fc\, C(V, \bar{a}) : \bar{a} \in D \end{aligned} \tag{3.47}$$

Any solution s_i that fulfills the above two conditions is a solution that performs according to the desired performance in the presence of variation. A solution validated by equation 3.46 is a solution that may assemble without any information about the respect of the functional characteristics. However, if a solution subsequently validates the expression for the respect of functional conditions (equation 3.47) then it is a solution that assembles and respects the functional conditions. The conditions in equations 3.46 and 3.47 form the fundamental logical and validation basis for step 2 in figure 3.1 and are, therefore, at the heart of the tolerance analysis problem. This formalization can be used for assembly and functional condition verification of mechanical assemblies for tolerance analysis applications (Dantan & Qureshi, 2009).

Detailed text dedicated to demonstrating the application of these expressions to with the help of an algorithm for 3D tolerance analysis of assemblies is presented in chapter 5.

3.6 Synthesis

This chapter presents the formalization that unifies and presents two major variation management steps in the product design process namely variation management through robust design as well as variation management through tolerance analysis. This work takes its theoretical roots in the earlier research work by Dantan & Ballu (2002) who developed the idea of quantifier based expression to describe the interaction of the different deviations and gaps in a mechanical assembly. Using the developed notion of quantifier by Dantan et al, this work provides a generalization and harmonization of the quantifier notion in a more structured and syntactic paradigm of formal logic.

Expression and generalization through formal logic has allowed development of a uniform and identical expression which can then be used to integrate the variation management in general in the design phase. In the field of robust design, the developed formalization expresses the search for a robust solution through a bi-conditional expression that tests the design space for the existence of a possible solution followed by the validation of that solution in the presence of variation, both in the design parameter as well as from other sources such as manufacturing variation. This integration effectively renders the solution inherently robust and insensitive to the changes within the decided ranges. This formalization also makes it possible to carry out the robust design in a set based design process by manipulating and evaluating the sets of variables instead of points in design space which is the main premise of the set based design. Also, through universal or existential quantification of the variables involved, design progress through set based filtering is carried out, adding to the flexibility in the design stage via availability of alternatives throughout the design phase.

The fundamental work in the tolerance synthesis were developed by Dantan & Ballu (2002). This approach has been broadened and structured with the framework of FOL. The developed framework successfully encompasses the existing mathematical quantifier notion in the paradigm of formal logic and provides means to integrate the variation management through tolerance analysis of mechanisms. The existing definitions of hulls and deviations have been retained for the geometrical behavior model and are supplemented by logical conditions stipulating the checks for assemblability and respect of functional conditions for the assembling mechanism. All this is formalized through the common logical framework described in the beginning of the chapter and therefore allowing homogeneous expression and syntax.

The formalization of variation management in product design allows the development and application of the formalization to robust design and tolerance analysis problems in mechanical design. For this purpose, important questions have to be answered about the transformation of the developed logical conditions, in a computable form, which can then be applied to any given model, in conjunction with appropriated algorithmic tools, to perform search and evaluation of the design space. These questions are addressed in the next two chapters.

Chapter 4 provides a detailed discussion on the application of the formalization of robust design to the problem of product design of mechanical components and provides an over view of the techniques and methods employed to carry out the transformation of the logical expression in computable form.

Similarly, chapter 5 provides a detailed over view of the process for practical application of the developed tolerance analysis syntax for the tolerance analysis of the mechanical assemblies. The chapter provides a discussion over the selection, development and application of the necessary tools and methods needed to convert and transform the logical expressions developed in an applicable algorithmic form of application to different mechanical assemblies.

Chapter 4

Application to Set Based Robust Design

The formalization of the robust design in the product design phase was presented in Chapter 3. In order to implement the formalization in a computable form, it is necessary to transform the formalization so that it can be applied to examples in mechanical product design. This chapter, therefore, presents the process for the transformation of the developed formalization of the robust design. For this purpose, the necessary steps for the development of the implementation strategy, tools, techniques and methods for transformation are discussed. This is achieved by: selection of tools and development of methods, that can be used to apply the set based design space exploration using developed formalization; appropriate methods to express and evaluate the quantified expressions resulting from formalization; necessary methods and techniques in the algorithmic design space exploration for a search and evaluation algorithm for space exploration for sets of robust solutions. The capability to evaluate the quantified expressions is an important step in the application process. This is done through transformation via consistency verification.

Once the application is developed, different examples from the mechanical product design are discussed. These examples are solved for robust set based design solutions through space exploration. The conclusion discusses the results of the application to the examples and provides a qualitative analysis of the approach.

4.1 Considerations for Application of Robust Design Formalization

Application of the design process requires adoptions of the tools and methods appropriate to the transformation of each process. A diagram of the major design theories and methods used to transform and apply them is shown in Figure 4.1.

		Primary Basis					
Type	**Method/Process**	Knowledge Engineering	Logic/Set Theory	Matrix Algebra	Probability	Statistics	Economics
Practical	Concurrent Engineering						■
Qualitative	Decision Matrix			■			■
	Pugh Method			■			
	QFD			■			
	AHP			■			
	Product Plan Advisor	■		■	■		
Statistical	PLS				■	■	
	Taguchi Method				■	■	
	Six Sigma				■	■	
Creative	AI Support	■					
	TRIZ	■					
Axiomatic	Suh's Theory		■	■			
	Yoshikawa Theory		■				
	Math Framework		■		■	■	■
Validating	Game Theory				■		■
	Decision Analysis		■		■		■
	SBCE	■	■				

Figure 4.1: Over view of Existing Design methods and processes (Committee on Theoretical Foundations for Decision Making in Engineering Design, 2001)

The considerations for the application of the robust design formalization is to find the appropriate tools, in line with the primary basis of the SBCE as shown in figure 4.1, which allows the application of the developed formalization.

The expressions for robust design, developed in Chapter 3, express the quantified logical conditions for the existence of a solution followed by the conditions for the existence of robust solutions. In order to implement these expressions into an applied form, which can be used to evaluate a given design model, it is necessary to transform these quantified expressions into an applicable algorithm. This algorithm should be capable of taking the initial design space as an input and then of applying the logical condition for the validation of the design space. To achieve this, tools and methods need to be chosen and/or developed into three main categories.

In the first step, the algorithm needs to manage the data pertaining to the entirety of the initial design space.

Once this data is available, the next step is to have a capability of aggregating the fundamental analytical model, which represents the set of constraints for the product as per the functional requirements. This analytical model is to be evaluated with respect to the quantified expressions as stipulated by the logical expressions containing universal or existential quantifiers. This step, therefore, should develop applied methods to translate and evaluate the quantified expressions and confirms the validity of the design space under consideration.

The last step involves the development of an algorithm that performs the set based design space search. The algorithm should be able to take the initial design space, decompose it into sub spaces, i.e. sub set of the design spaces, which can then be evaluated by a method for the validity of the space with respect to the constraints. The algorithm to be designed should also be capable of filtering the space into the sets of valid solutions, robust solutions and the space without solutions.

Eventually, in order to understand and interpret the results, a domain visualization system is also needed that is capable of displaying the results in a user friendly yet flexible environment allowing the designer to visualize the resulting design space with multiple view points and presenting the solution sets for aiding in the decision making.

The following sections address these identified areas and present an overview of the tools searched, developed, and retained for the transformation of the robust design formalization for application to problems.

4.2 Design Space Representation

In order to apply the developed formalization, the chapter addresses the design stages, where fundamental quantitative constraints with respect to the functional conditions can be formulated. This means that the clients requirements for a given product has been mapped qualitatively as well as quantitatively, resulting into the set of fundamental constraints as well as the key design parameters whose initial values can then be communicated in terms of sets.

The first step deals with the type of data representation for the initial design space. As the set based design approach is retained, therefore, in line with the fundamental principle of the set based design, the design space and the solution space should be capable of manipulating the data in the form of sets and ranges instead of points, as normally done in point based design. These sets may be in the form of ranges of continuous variables or sets of discrete integers. The data types can be a combination of any types of data described earlier in the section 3.4.1. Once these sets have been decided, the sets of associated noise variables are decided. Depending upon the arity of the key design parameters and the associated noise variables, the initial design space is then formulated as an n-dimensional hypercube that represents the starting point for the design problem. It is assumed that the sets of desired solutions are the intersections of the planes lying within this hypercube.

4.3 Consistency Evaluation

Once the initial design space has been defined, the next step is to evaluate the design space for validity in terms of the constraints. This essentially starts by decomposing the initial design space as per the desired strategy to evaluate the decomposed parts for validity through constraint propagation. In order to do so, it is necessary to provide the tools and methods for the evaluation of the quantified expressions. The evaluation of quantifiers falls under the domain of Quantified Constraint Satisfaction Problem Resolution (QCSP). A number of research works exist in the field of Mathematics and Computer science that address the theoretical and algorithmic aspects of resolution of QCSP and logical constraints. These tools include hull and box consistency techniques (Benhamou *et al.*, 1999; Cruz & Barahona, 2001, 2003), quantified arc consistency techniques (Bordeaux & Monfroy, 2006; Mamoulis & Stergiou, 2004), constraint logic programming over reals and integers (Benhamou & Older, 1997), constraint logic programming over interval (Benhamou *et al.*, 1994), predicate calculus based logic for solving search problems (East & Truszczynski, 2006) and interval propagation to reason about sets (Gervet, 1997).

The arc consistency techniques aim at filtering and reducing the variable domains by taking into account the individual variable domain consistency in a given arc and then re-evaluating it with respect to other variable domains in an iterative manner until all the variable domains are consistent with the constraints involved. Enforcing hull consistency usually requires decomposing the user's constraints into so-called primitive constraints while the box consistency treats constraints without decomposing them (Benhamou *et al.*, 1999).

To implement the consistency evaluation in a given domain subset, with respect to the conditional logical expressions, the box-consistency technique has been adopted. It is therefore necessary to discuss the fundamental steps required to implement the box consistency technique to transform the expressions for consistency evaluation of set based robust design.

4.3.1 Transformation

To implement the approach described, we need to transform the notions of the existential and universal quantifiers in a computable form for resolution. Box consistency technique has been selected to transform the quantifiers. In order to implement the box consistency technique, the first step is to convert the design domain and the associated constraints into interval arithmetic. This transformation is tool independent and can be incorporated and used on a variety of computational tools. The following text describes the basic notations and definitions used for the transformation of the problem into an interval based problem. It is then, extended to the constraints to carry out the required evaluations using box consistency. The transformation into the computable approach in this work has been carried out in the Mathematica® software.

4.3.2 Basic notations and definitions

The notations and definitions used regarding the intervals and related operations are adopted from the interval notations in (Vareilles, 2005) and (Parsons & Dohnal, 1992).

- Consistent with earlier descriptions, the real numbers are represented by a small letter in italics and bold "a".
- The intervals are represented by a non italic small letter in bold **a**.
- The higher limit of an interval is represented as $\bar{\mathrm{a}}$.
- The lower limit of an interval is represented as $\underline{\mathrm{a}}$.
- $\tilde{\mathrm{a}}$ is an instance of the interval **a**

Also: $\mathbb{R}^{\infty} = \mathbb{R} \cup \{-\infty, +\infty\}$= Set of real numbers

- "c" represents a constraint over real numbers.
- "c" represents a constraint over intervals.
- An interval of real numbers **a**$= [\underline{\mathrm{a}}, \bar{\mathrm{a}}]$ with $\underline{\mathrm{a}}$ and $\bar{\mathrm{a}} \in \mathbb{R}^{\infty}$ is a set of real numbers r such that $\{r \in \mathbb{R}^{\infty} | \underline{\mathrm{a}} \leq r \leq \bar{\mathrm{a}}\}$ if $\underline{\mathrm{a}}$ or $\bar{\mathrm{a}}$ is one of $-\infty$ or $+\infty$, then **a** is an open interval.

4.3.2.1 Interval operations

Interval arithmetic is based on the extension of constraints applicable to real numbers, so that they become applicable to the intervals. If constraint applies on variables in real number domain then it applies to the intervals as well. These extensions exist for most of the elementary operators $(+, -, /, \times, etc)$. The following example summarizes the concept of expression of an operator in terms of real numbers as well as interval expression.

Let $f(x, y)$ be a function of two variables x and y then using this function, the definition of the arithmetic operators over real numbers and intervals would be as follows (Chenouard, 2007; Parsons & Dohnal, 1992; Vareilles, 2005) :

Addition The addition in terms of real numbers and intervals is as follows:
$f\colon (x, y) \mapsto x + y$
f: $(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{x} \oplus \mathbf{y} = [\underline{\mathrm{x}} + \underline{\mathrm{y}}, \bar{\mathrm{x}} + \bar{\mathrm{y}}]$

Substraction The substraction operation in terms of real numbers and intervals is as follows:
$f\colon (x, y) \mapsto x - y$
f: $(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{x} \ominus \mathbf{y} = [\underline{\mathrm{x}} - \bar{\mathrm{y}}, \bar{\mathrm{x}} - \underline{\mathrm{y}}]$

Multiplication The multiplication in terms of real numbers and intervals is as follows:
$f\colon (x, y) \mapsto x \times y$
f: $(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{x} \otimes \mathbf{y} = [\min(\bar{\mathrm{x}}\times\bar{\mathrm{y}}, \underline{\mathrm{x}}\times\underline{\mathrm{y}}, \underline{\mathrm{x}}\times\bar{\mathrm{y}}, \bar{\mathrm{x}}\times\underline{\mathrm{y}}), \max(\bar{\mathrm{x}}\times\bar{\mathrm{y}}, \underline{\mathrm{x}}\times\underline{\mathrm{y}}, \underline{\mathrm{x}}\times\bar{\mathrm{y}}, \bar{\mathrm{x}}\times\underline{\mathrm{y}})]$

Division The division in terms of real numbers and intervals is as follows:
$f\colon (x, y) \mapsto x/y$
f: $(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{x} \oslash \mathbf{y} = [\min(\bar{\mathrm{x}}/\bar{\mathrm{y}}, \underline{\mathrm{x}}/\underline{\mathrm{y}}, \underline{\mathrm{x}}/\bar{\mathrm{y}}, \bar{\mathrm{x}}/\underline{\mathrm{y}}), \max(\bar{\mathrm{x}}/\bar{\mathrm{y}}, \underline{\mathrm{x}}/\underline{\mathrm{y}}, \underline{\mathrm{x}}/\bar{\mathrm{y}}, \bar{\mathrm{x}}/\underline{\mathrm{y}})]$ if $0 \notin [\underline{\mathrm{y}}, \bar{\mathrm{y}}]$ else $[-\infty, +\infty]$,

The symbols $\otimes, \ominus, \oplus$ and $\oslash$ are the extensions of $\times, -, +$ and $/$ operators on the intervals.

4.3.2.2 Extension of constraints

Any given constraint function c_i is a natural extension of a constraint c_i if c_i is obtained by replacing each occurring constant k_j in the expression by the smallest possible corresponding interval $\mathbf{k}_j$, each possible assignment a_l of variable $\boldsymbol{v}_l$ by the smallest possible interval assignment $\mathbf{a}_l$ and each arithmetic operation by its interval extension. In this way, we can convert the constraint to an interval constraint. As discussed earlier, the constraint may be an equality or an inequality. In the case of an equality, due to the iterative inner floating point operations carried out, the constraints involving zeros on one side of the equation are evaluated for the given machine precision ϵ which denotes the minimum possible incremental precision attainable by the machine/calculation engine.

4.3.2.3 Interval Analysis

The application of the interval arithmetic essentially allows us to convert the problem from a real number solution to an interval solution which can then be applied to the quantifier translation. In order to show this, we will take the previous example of $f(x, y)$ and demonstrate the effect of moving towards the interval arithmetic.

Let x and y be two variables such that $x = 30$ and $y = 5$. The value of the arithmetic operations performed with these values and their corresponding interval extension calculations by assigning intervals $\mathbf{x}$=[10,40] and $\mathbf{y}$=[4,6] respectively instead of real numbers is illustrated in table 4.1

Real numbers $(x = 30, y = 5)$		**Intervals** ($\mathbf{x}$=[10,40],$\mathbf{y}$=[4,6])	
Operation	Result	Operation	Result
$x + y$	35	$\mathbf{x}\oplus\mathbf{y}$	$[14, 46]$
$x - y$	25	$\mathbf{x}\ominus\mathbf{y}$	$[4, 36]$
$x \times y$	150	$\mathbf{x}\otimes\mathbf{y}$	$[40, 240]$
x/y	6	$\mathbf{x}\oslash\mathbf{y}$	$[5/3, 10]$

Table 4.1: Real number operations and corresponding interval operations.

The above example shows the usage of interval arithmetic to convert basic arithmetic functions applicable to real numbers to their corresponding intervals. The effect achieved is to include the total interval in the calculation rather than to calculate one unique value. The resulting output of an interval arithmetic calculation is also an interval which defines the boundary of the solutions originating from the input intervals. This concept will be applied to the constraints of the design problem.

For the sake of clarity, it is necessary to define some terms which will be used during the transformation.

Definition 1 The design variables involved in the problem are expressed in the forms of intervals except in the case of design variables of discrete nature. Each interval is a set of connected reals with lower and upper bounds as floating point intervals. The corresponding interval assignment $\mathbf{a}_i$ for an real number assignment a_i to an *ith* design variable v_i is defined as:

$$\mathbf{a}_i = [\underline{\mathrm{a}}_i, \bar{\mathrm{a}}_i] \equiv \{\mathrm{a}_{v_i} \in \mathbb{R} | \underline{\mathrm{a}}_{v_i} \leq a_{v_i} \leq \bar{\mathrm{a}}_{v_i}\} \tag{4.1}$$

Also, the relationship between the interval $\mathbf{a}_i$ and the domain d_{v_i} of the variable v_k is given as:

$$\mathbf{a}_{v_i} \subseteq d_{v_i} \tag{4.2}$$

Definition 2 A Cartesian product of n intervals $B = \mathbf{a}_{v_1} \times ....., \times \mathbf{a}_{v_n}$ is called a box; a domain d_{v_i} associated to a variable v_i is either an interval $\mathbf{a}_{v_i}$ or a union of disjoint intervals. B is equal to or is a subset of the domain set D:

$$B \subseteq D \tag{4.3}$$

Definition 3 The set of the initial domains of all the variables involved is D-Box. A D-Box with arity n is the Cartesian product of n intervals where n is the number of design variables involved in the problem. It is denoted by $\langle \mathbf{a}_{v_1}, ..., \mathbf{a}_{v_n} \rangle$ where each $\mathbf{a}_{v_i}$ is an interval. In the following text the term "BD" will be used for the D-Box

$$\mathbf{A} = \{\mathbf{a}_{v_i} | i \in [1, n], \mathbf{a}_{v_i} = d_{v_i}\} \tag{4.4}$$

$$BD = \mathbf{A} \tag{4.5}$$

Definition 4 An SD-Box with arity n is the Cartesian product of n intervals where n is the number of variables involved in the problem. It is denoted by BSD. SD-Box is formed when a D-BOX is split. In the following text the term "BSD" will be used for the SD-Box.

$$\begin{aligned}
\mathfrak{A}_i &= \{\mathbf{a}_{v_i} | i \in [1,n], \mathbf{a}_{v_i} \subset d_{v_i}\} && (4.6)\\
\mathfrak{A}_i &\subset \mathbf{A} && (4.7)\\
BSD &= \mathfrak{A}_i && (4.8)\\
BSD &\subset BD && (4.9)\\
& && (4.10)
\end{aligned}$$

The term $\mathfrak{A}_i$ is the ith BSD resulting from the decomposition of the BD into BSDs through a specified decomposition process.

Definition 5 An interval extension of a constraint $c_i(\bar{v}) : c_i(\bar{v}) = c_i(v_1 ....., v_n) : \mathbb{R}^n \mapsto \mathbb{R}$ with an assignment $\bar{a}$ such that $c_i(\bar{a}) = c_i(a_1 ....., a_n) : \mathbb{R}^n \mapsto \mathbb{R}$ is a mapping $\mathbf{c}_i(\bar{\mathbf{a}}) = \mathbf{c}_i(\mathbf{a}_1 ....., \mathbf{a}_n) : \mathbb{I}^n \mapsto \mathbb{I}$ where:

$$\begin{aligned}
V &= \{v_i, \delta_{v_i} | i \in [1,n], v_i \in d_{v_i}, \delta_{v_i} \in d_{\delta_{v_i}}\} && (4.11)\\
\mathbf{A} &= \{\mathbf{a}_{v_i}, \mathbf{a}_{\delta_{v_i}} | i \in [1,n], \mathbf{a}_{v_i} \in d_{v_i}, \mathbf{a}_{\delta_{v_i}} \in d_{\delta_{v_i}}\} && (4.12)
\end{aligned}$$

$\bar{v}$ represents a specific constraints with the specific variables present in the relation. The constraint to be extended may be an inequality or equality and may be in any of the following forms:

$$\begin{aligned}
c_i(v_1, ..., v_j) &= 0 \rightarrow \mathrm{c}_i(\mathbf{a}_{v_1}, ..., \mathbf{a}_{v_j}) = 0 && (4.13)\\
c_i(v_1, ..., v_k) &\leq 0 \rightarrow \mathrm{c}_i(\mathbf{a}_{v_1}, ..., \mathbf{a}_{v_k}) \leq 0 && (4.14)\\
c_i(v_1, ..., v_k) &\geq 0 \rightarrow \mathrm{c}_i(\mathbf{a}_{v_1}, ..., \mathbf{a}_{v_k}) \geq 0 && (4.15)
\end{aligned}$$

Where $\mathbf{a}_{v_i}$ is the interval assignment to the variable v_i. Using the above definitions, an algorithm has been developed which applies the quantifier notion to robust design.

4.3.3 Consistency for the existence of a solution

Equation 3.29 expresses the existence of solution in terms of the existential quantifier. Its transformation into the algorithm with the help of the interval analysis stipulates that BSD should be consistent with the given constraints:

$$\begin{aligned}
DV &= \{v_k | k \in [1,n], v_k \in d_{v_k}, \}\\
\mathbf{A} &= \{\mathbf{a}_{v_k}, | k \in [1,n], \mathbf{a}_{v_i} \in d_{v_k}\}\\
s_i &\models (\exists DV \min \mathbf{C}(V, \mathbf{A}) \vee \exists DV \max \mathbf{C}(V, \mathbf{A})) && (4.16)
\end{aligned}$$

For the BSD to be validated through the check performed by equation 4.16, a further robustness check is performed for the consistency of BSD in the presence of noise as stipulated by equation 3.31.

4.3.4 Consistency for the existence of a robust solution

The presence of noise/uncertainty is denoted by Δ as described earlier, where δ_{v_i} is the noise/uncertainty related to the design variable v_i. A solution is robust if the BSD is box-consistent in the presence of the noise parameters. Therefore, equation. 3.31 may be expressed as:

$$\begin{aligned}
V &= \{v_k, \delta_{v_k} | k \in [1, n], v_k \in d_{v_k}, \delta_{v_k} \in d_{\delta_{v_k}}\} \\
\mathbf{A} &= \{\mathbf{a}_{v_k}, \mathbf{a}_{\delta_{v_k}} | k \in [1, n], \mathbf{a}_{v_k} \in d_{v_k}, \mathbf{a}_{\delta_{v_k}} \in d_{\delta_{v_k}}\} \\
s_i &\models (\forall V \min \mathbf{C}(V, \mathbf{A}) \wedge \forall V \max \mathbf{C}(V, \mathbf{A}))
\end{aligned} \tag{4.17}$$

The effect of the conversion is to assign the corresponding interval assignments to the quantified variables. Each variable is assigned an upper and lower bound taken from the extremities of the interval. This operation is carried out for all the involved variables including the noise and design variables. Similarly, the constraints are also transformed into interval constraints which are then able to take the interval assignments to the vocabulary. The constraints are then evaluated for the condition of existence of a solution. If a BSD does not contain any solution, it is discarded and subsequently BD is reduced. If an existence solution is found then this BSD is evaluated for the consistency of it's universal quantifier in the presence of the noise. In case of a successful evaluation, the BSD is saved as a robust design solution space.

4.4 Space Exploration Tools

The above sections develop the basis for the transformation of the formalization into an applicable consistency evaluation form. With these consistency evaluation techniques, a BSD can be evaluated for the existence and validation of a solution. This however needs a methodology for searching, dividing and pruning the departing design space so that the domain reduction towards feasible regions can be achieved. In order to develop this capability, an algorithm has been developed that carries out the domain decomposition in a tree based search of initial design space and applies the consistency techniques to the BSDs to carry out the domain reduction.

The search method in the algorithm is of branch and bound type. This method is also known as tree search. This method divides the domain of the variables into sub domains therefore generating a sub space which can then be explored for the solution. This is analogous to dividing the problem into sub problems which can then be solved separately. Such a search strategy can be divided into two main categories i.e. the depth first search (DFS) or Breadth first search (BFS). A number of algorithms exist which allow the tree based search through space for solutions. The algorithm deployed for the set based design space exploration is a breadth first search based exhaustive algorithm.

The initial design domain specified by the design engineer is encapsulated in BD and is used as the starting search space for the algorithm. The algorithm then proceeds by dividing BD in the number of BSD as specified.

The algorithm developed for the set based robust design space exploration is divided into three main parts. The first step takes the BD as an input and is responsible for dividing the BD in BSD. It then assigns the BSD to be evaluated to the next step for the evaluation of the existence of robust solution. The conversion from BD to the BSD is a splitting process which depends on the arity of the set BD. The domain splitting process ensures that that each BSD is a subset of BD and that the splitting process applies equally to each of the members of BD. Similarly , during the subsequent iterations, the splitting of the further BSDs into smaller BSDs is also performed by this module. The splitting of the BSD in further iterations is automatically increased in the resolution as per the depth requirement of the split to be achieved.

Once a BSD is created, this BSD is evaluated by the algorithm for consistency of existence of the solution. Using the transformations described earlier, the algorithm carries out the existence consistency check. Subject to the validation of the existence of the solution consistency, the algorithm then passes on to the next step of evaluation of the consistency of the robust solution for the

BSD currently being evaluated. This results into the verification of existence of a robust solution or the instruction for further decomposition of the design space.

The algorithm iteratively processes the entire design space for the domain reduction for finding the sets of robust solutions and at the end of the iterations, returns the results in the form of the feasible regions within the design space. The results returned by the algorithm are stored in the results module which then processes the results to present the domains related to the robust solution. These results are presented in form of 2D or 3D projections between the selected variables and represent the feasible regions of the space graphically to help the designer in decision making for choosing the most feasible solution.

Algorithm 4.1 presents the global algorithm for carrying out the search and evaluation of the initial design space for the feasible set of robust solutions. It can be seen that in order to conserve the much needed computational power, the consistency check for the robust sets which involves the evaluation of the universal quantifiers is not carried out unless a design space with the possibility of a solution is encountered. Once the regions of the possible solutions are encountered, the algorithm then proceeds to evaluate the consistency for robust solution and finally stores the results in terms of filtered design space.

4.5 Illustrative Example

This section illustrates the application of the formalization of the set based robust design space exploration of the design and variation space. A simple example is presented to show how the transformation of the formalization takes place with the help of the tools and methodology discussed in the previous sections.

4.5.1 Problem Description

For the ease of demonstration, a very simple mechanical system as shown in Figure 4.2 is considered. This is a simple, vertically fixed beam FG of a rectangular cross section with length l and a mass M to be supported at the suspended end. While considering the problem, it is assumed that the beam is of a uniform rectangular cross-section throughout its length.

Algorithm 4.1 Global Algorithm for robust set based design space exploration

Require: $\{QV, D, C\}, res, dim, iterations$ $\{res$ =Number of Required sub interval resolution, dim =Length of DV$\}$
Ensure: Feasible sets of robust solution

$BD \leftarrow D$
$BSD\_list = \{\bigcup_{i=1}^{res}\{d_{i_{v_j}} : j \in \{1, dim\}\}\}$
$Exist\_list \leftarrow \varnothing$ {Set of Possible Solutions}
$Robust\_list \leftarrow \varnothing$ {Set of Robust Solutions}
$NoSol\_list \leftarrow \varnothing$ {Space Without Solution}
for $k = 1$ **to** $iterations$ **do**
    for $l = 1$ **to** Length of BSD_list **do**
        Pick BSD_l
        Convert BSD_l to interval based sets
        if Consistency of existence of solution of BSD_l= **true then**
            if Consistency of robust solution of BSD_l= **true then**
                $Robust\_list \leftarrow Robust\_list \cup BSD_l$
            else
                $Exist\_list \leftarrow Exist\_list \cup BSD_l$
            end if
        else
            $NoSol\_list \leftarrow NoSol\_list \cup BSD_l$
        end if
    end for
    $BSD\_list \leftarrow Exist\_list$
end for
Display Results

For this example, it is considered that the mechanical properties of the material and the section geometry remain uniform throughout the length of the beam. The stress and strain in the beam are considered to be in the elastic region. Also the stress distribution throughout the beam is considered to be uniform. The beam is required to support a mass M and its own weight. The design problem is to find the sets of values of the design parameters a and b which are the dimensions of the cross section $o - o'$ of the beam.

The analytical model of the beam is based on three principal constraints that are: maximum admissible stress in the beam, maximum admissible mass of the beam, and maximum admissible elastic elongation in the beam. These constraints

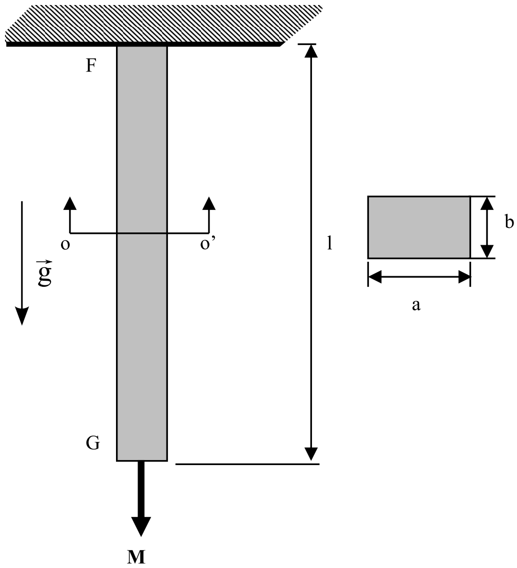

Figure 4.2: Simple Beam with two design variables

are mathematically expressed by the following expressions:

$$\sigma \leq \sigma_{adm} | \sigma = -\frac{Mg}{(a+\delta_a)(b+\delta_b)} - ((l+\delta_l) - x)\rho g \tag{4.18}$$

$$m \leq m_{adm} | m = \rho(l+\delta_l)(a+\delta_a)(b+\delta_b) \tag{4.19}$$

$$\partial l \leq \partial l_{adm} | \partial l = -\frac{g(l+\delta_l)}{E}\left(\frac{M}{(a+\delta_a)(b+\delta_b)} + \frac{(l+\delta_l)}{2}\rho\right) \tag{4.20}$$

where $\sigma, m, M, a, b, l, \partial l$ denote stress, mass of the beam, mass to be supported, width of the beam, thickness of the beam, length of the beam and change in length of beam, respectively, whereas the corresponding symbols with subscript *adm* denote the maximum allowable limits for the respective symbols. Table 4.2 summarizes the variables.

All the constraints in this case are inequalities. In order to satisfy the constraints, such sets of a and b are to be found, which allow a robust solution accounting for the noise which simulates the variations in the design parameters as described in Table 4.2. The model can be expressed by transforming the equations 3.29 and 3.31 as discussed earlier, resulting into a model described by equations 4.21 and 4.22:

Symbol	Type	Type	Description	Range
l	Real	Constant	Length	0.6m
a	Real	Design Variable	Width	[0.005,0.07]m
b	Real	Design Variable	Breadth	[0.005,0.07]m
mt	Discrete	Design Variable	Material	[Steel A36, Al 2014-T6]
M	Real	Design Variable	Mass to be supported	5000 kg
m	Real	Design Variable	Mass of the beam	To be calculated
σ	Real	Design Variable	Calculated Stress	Material Dependent
σ_{adm}	Real	Constant	Admissible Stress	Material Dependent
M_{adm}	Real	Constant	Maximum mass	6 kg
∂l_{adm}	Real	Constant	Max. deflection	0.0002m
∂l	Real	Design Variable	Change in length	To be calculated
δ_l	Real	Noise Variable	Variation in length	[-0.001,0.001]m
δ_a	Real	Noise Variable	Variation in width	[-0.001,0.001]m
δ_b	Real	Noise Variable	Variation in Breadth	[-0.001,0.001]m

Table 4.2: Variables for the suspended beam example.

$$
\begin{aligned}
V &= DV \cup \Delta \\
DV &= \{a \in d_a, b \in d_b, \partial_l, mt \in d_{mt}\} \\
\Delta &= \{\delta_a, \delta_b, \delta_l\} \\
C &= \{\sigma(X), m(X), \partial l(X) | X \in V\} \\
s_i &\models \exists DV\, C(V, A) \qquad (4.21) \\
s_i &\models \forall V\, C(V, A) \qquad (4.22)
\end{aligned}
$$

4.5.2 Conversion to interval arithmetic for consistency

Having developed the descriptive expressions for the solution consistency and robust solution consistency for the problem in the form of equations 4.21 and 4.22, respectively, the next step is to convert the expressions in a computable form. This means converting the design space so that consistency techniques discussed earlier can be applied. For this purpose, the starting design space is converted into the set of real interval and thus an initial BD is created.

Considering the equation 4.21, the existential quantifier performs a check of existence for a design solution. In order to implement this check via consistency, we transform the problem by the replacement of the existential quantifier by extension to the interval. To implement this, three generalizations pertaining to the three general types of constraints are defined as proposed in (Vareilles *et al.*, 2009).

$$c_i(a,b) = 0|c_i \in C \tag{4.23}$$
$$c_j(a,b) \leq 0|c_j \in C \tag{4.24}$$
$$c_k(a,b) \geq 0|c_k \in C \tag{4.25}$$

Let equation 4.19 be written as $c(a,b) \leq m_{adm}$ which is an implicit constraint, with variables $a \in d_a$ and $b \in d_b$. By the interval extension discussed in an earlier section, this constraint transforms to $\mathrm{c}(\mathbf{a},\mathbf{b}) - m_{adm} \leq 0|\mathbf{a} = [\underline{\mathrm{a}},\bar{\mathrm{a}}], \mathbf{b} = [\underline{\mathrm{b}},\bar{\mathrm{b}}]$. Once the constraint has been extended over the interval, the next step is to calculate the boundary values of $\mathrm{c}(\mathbf{a},\mathbf{b})$ for the given interval, i.e. $\mathrm{c}(\mathbf{a},\mathbf{b}) = [\underline{\mathrm{c}}(\mathrm{a,b}), \bar{\mathrm{c}}(\mathrm{a,b})]$. If, for the given interval assignments, $\underline{\mathrm{c}}(\mathbf{a},\mathbf{b}) - \mathbf{m}_{adm} \leq 0$, there is a possibility of a solution. If $\bar{\mathrm{c}}(\mathbf{a},\mathbf{b}) - \mathbf{m}_{adm} \leq 0$, there is a possibility of a solution over the totality of interval (robust solution). If, $\underline{\mathrm{c}}(\mathbf{a},\mathbf{b}) - \mathbf{m}_{adm} > 0$, no solution exists. Similarly for constraints of the form $\mathrm{c}(\mathbf{a},\mathbf{b}) = 0$ if, $0 \in [\underline{\mathrm{c}}(\mathrm{a,b}), \bar{\mathrm{c}}(\mathrm{a,b})]$ the constraint is consistent and there is a possibility of solution. If, $0 \notin [\underline{\mathrm{c}}(\mathrm{a,b}), \bar{\mathrm{c}}(\mathrm{a,b})]$ and for the third case $\mathrm{c}(\mathbf{a},\mathbf{b}) \geq 0$ if, $\bar{\mathrm{c}}(\mathrm{a,b}) \geq 0$, there is a possibility of a solution, if $\underline{\mathrm{c}}(\mathrm{a,b}) \geq 0$, there is a possibility of a solution over the totality of the interval (robust solution) and if $\underline{\mathrm{c}}(\mathrm{a,b}) < 0$, no solution exists.

Therefore, for all the cases presented above, there are three possible outcomes of interval consistency with regard to the constraints. The outcome can be either of the following (Vareilles *et al.*, 2009):

1. The interval is consistent with the constraint.

2. The interval is inconsistent with the constraint.

3. The interval is consistent and inconsistent with the constraint

In order to translate the consistency developed above, for application to quantified expressions, we can now establish that over a given interval, the universal quantifier can be evaluated to be true if the interval is consistent with the constraint. Similarly, if the interval is consistent and inconsistent with the constraint, the existential quantifier can be evaluated to be true. If the interval is inconsistent with the constraint the quantifier evaluation returns false.

Using the above rules, the equations for solution consistency and robust solution consistency with existential and universal quantifiers, respectively, as described above transform into:

$$\begin{aligned}
&\min(\rho \otimes \mathbf{l} \otimes \mathbf{a} \otimes \mathbf{b}) \leq \mathbf{m}_{adm} \wedge \\
&\min\left(\left|-\frac{\mathbf{M} \otimes g}{\mathbf{a} \otimes \mathbf{b}} \ominus (\mathbf{l} \ominus x) \otimes \rho \otimes g\right|\right) \leq \sigma_{adm} \wedge \\
&\min\left(\left|-\frac{g \otimes l}{E}\left(\frac{\mathbf{M}}{\mathbf{a} \otimes \mathbf{b}} \oplus \frac{\mathbf{l}}{2} \otimes \rho\right)\right|\right) \leq \partial l_{adm}
\end{aligned} \tag{4.26}$$

Equation 4.26 describes the consistency for the existence of the solution. It lays down the first check for the design variables to evaluate if the BSD under consideration contains a feasible solution or no. If the BSD contains a feasible solution, then the next step is to carry out the consistency check for robust solution. This is shown by equation 4.27.

$$\begin{aligned}
&\min(\rho \otimes \mathbf{l} \otimes \mathbf{a} \otimes \mathbf{b}) \leq \mathbf{m}_{adm} \wedge \\
&\min\left(\left|-\frac{\mathbf{M} \otimes g}{\mathbf{a} \otimes \mathbf{b}} \ominus (\mathbf{l} \ominus x) \otimes \rho \otimes g\right|\right) \leq \sigma_{adm} \wedge \\
&\min\left(\left|-\frac{g \otimes l}{E}\left(\frac{\mathbf{M}}{\mathbf{a} \otimes \mathbf{b}} \oplus \frac{\mathbf{l}}{2} \otimes \rho\right)\right|\right) \leq \partial l_{adm} \\
&\max\left(\rho \otimes (\mathbf{l} \oplus \delta_l) \otimes (\mathbf{a} \oplus \delta_a) \otimes (\mathbf{b} \oplus \delta_b)\right) \leq \mathbf{m}_{adm} \wedge \\
&\max\left(\left|-\frac{\mathbf{M} \otimes g}{(\mathbf{a} \oplus \delta_a) \otimes (\mathbf{b} \oplus \delta_b)} \ominus ((\mathbf{l} \oplus \delta_l) \ominus x) \otimes \rho \otimes g\right|\right) \leq \sigma_{adm} \wedge \\
&\max\left(\left|-\frac{g \otimes (\mathbf{l} \oplus \delta_l)}{E}\left(\frac{\mathbf{M}}{(\mathbf{a} \oplus \delta_a) \otimes (\mathbf{b} \oplus \delta_b)} \oplus \frac{(\mathbf{l} \oplus \delta_l)}{2} \otimes \rho\right)\right|\right) \leq \partial l_{adm}
\end{aligned} \tag{4.27}$$

Equation 4.27 translates the universal quantifier via interval analysis for consistency check for a robust solution. This equation integrates the variations as described in Table 4.2 and evaluates the constraints on the boundary of the intervals to establish whether the BSD contains a robust solution or not.

4.5.3 Results

In the case of this example, the algorithm was run for 6 iterations with a sub interval resolution of 2, i.e. each interval is divided into 2 sub intervals during the BSD generation. Being a two dimensional design space, the solution returned by the algorithm is shown in figure 4.3 which is a 2D plot showing the design space decomposed into BSDs of increasing resolution showing distinctly; robust sets with the help of blue boxes (light gray in monochrome prints); space without solution in red (dark gray in monochrome prints) and unexplored space with probability of solution with help of yellow boxes (white in monochrome prints).

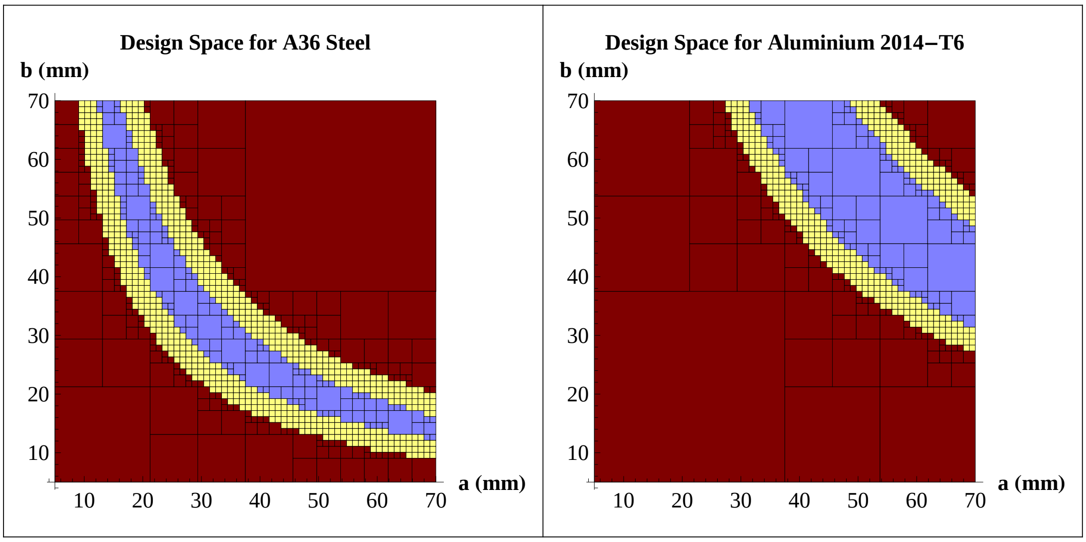

Figure 4.3: Algorithm results for robust solution/probable solution/no solution.

The BD and BSD splitting process can be visualized from the results in Figure 4.3. The figure shows the results for the design problem while considering two materials (steel on left and Al alloy on right). The BD which is the initial design space is progressively split into smaller BSD as per the given sub interval resolution with each iteration.

In figure 4.3, during the 1st iteration, BD is split into 4 BSDs and each is evaluated for the existence of the solution. It is evident in both the steel and Al alloy diagram that 25 % of the space was discarded in the first iteration as having no solution (large red box on top right in steel diagram and on bottom left in Al-alloy diagram), leaving the algorithm with 75% of the BD therefore saving critical computational effort. In the second and third iteration further design space reduction takes place with the first robust solution appearing with Al alloy as material in the third iterations and with steel as material in thethe fourth iteration of algorithm. The solution is then refined more with the appearance of further solutions within the remaining iterations.

4.5.3.1 Results verification

The results obtained through the algorithm can be validated to establish the soundness of the transformation techniques used. For this purpose, the constraints as expressed by equations 4.18, 4.19 and 4.20 can be re-written in terms of design variables a and b while considering the variations to be equal to zero, ignoring the negative sign for the direction of the stress and replacing the inequalities by equalities for the maximum value of the admissible stress, weight and change in length. This results into the following equations:

$$b = \frac{Mg}{a(\sigma_{adm} - l\rho g)} \tag{4.28}$$

$$b = \frac{m_{adm}}{\rho l a} \tag{4.29}$$

$$b = \frac{glM}{aE}\left(\frac{1}{\delta l_{adm} - \frac{gl^2\rho}{2E}}\right) \tag{4.30}$$

Using the above three equations, and taking the input values of a as given by the starting set [0.005,0.07] and taking the corresponding values for material properties for the selected materials, it is possible to plot the curves for the values of b as given by each equation. This will give rise to three curves which can then be plotted against the results of the algorithm. Figure 4.4 shows the results returned by the algorithm with an overlay of the curves as generated by equations 4.28 (orange curve), 4.29 (red) and 4.30 (green).

It can be clearly seen that the restricting constraints having the most effect on the solution are the constraints of weight and the deformation in length. The stress constraint (orange plot) is relaxed and does not affect the final solution space. It can be observed that the other two constraining plots (weight in red and deformation in green) are well placed at the boundaries of the BSDs without solution and the BSDs with possible solution. It can also be further observed that the robust solution sets as returned by the algorithm lie well within the solution space as given by the plots and are in a space which is the subset of the solution

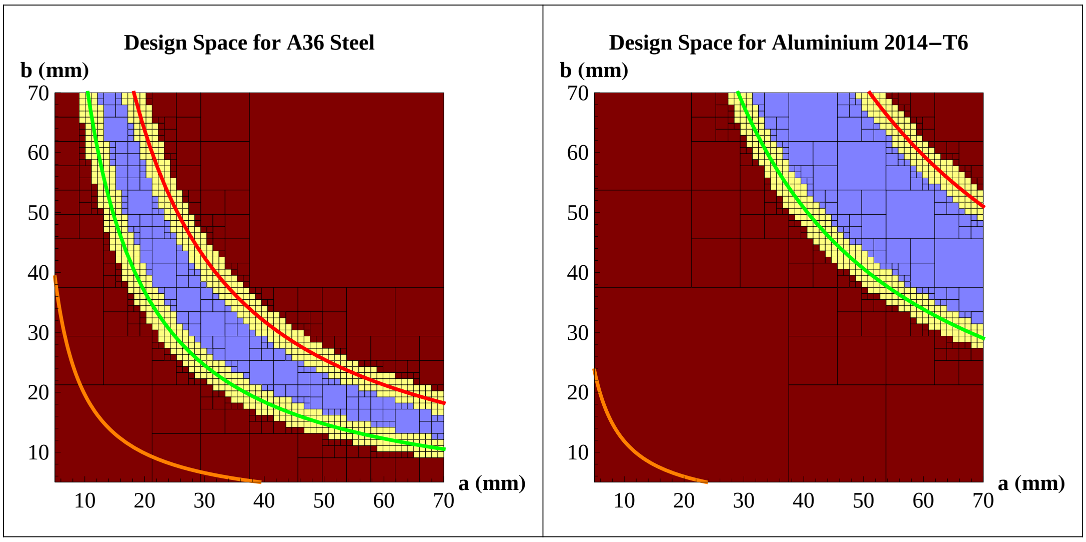

Figure 4.4: Verification of results for the simple beam.

space with the design parameters only. Therefore it can be concluded that the results returned by the algorithm are well in line with the results obtainable by traditional methods. The space of possible solution can eventually be divided into the space without solution and robust solution through further iterations, if deemed necessary.

4.6 Application to Examples

In order to validate the developed approach through more complex and pertinent examples from engineering design problems, different examples have been treated with the proposed approach. The examples have been carefully selected to demonstrate the applicability of the developed approach towards the different problems of engineering design. Three examples are presented in this work. The examples have been selected to validate the performance of the developed approach to the problems related to continuous variables, discrete variables and mixed variables as well as to the constraints of linear, non-linear and discrete constraints. Also, from an engineering point of view the problems address the commonly occurring themes in engineering design such as mechanical design, dimension design and path generation... These examples are: embodiment design of a two member truss; embodiment design of a flange coupling and embodiment design of a 6 bar mechanism for path generation.

The following sections elucidate upon the application of the theory to the selected examples.

4.6.1 Embodiment design of a two bar structure

A truss structure is shown in figure 4.5.

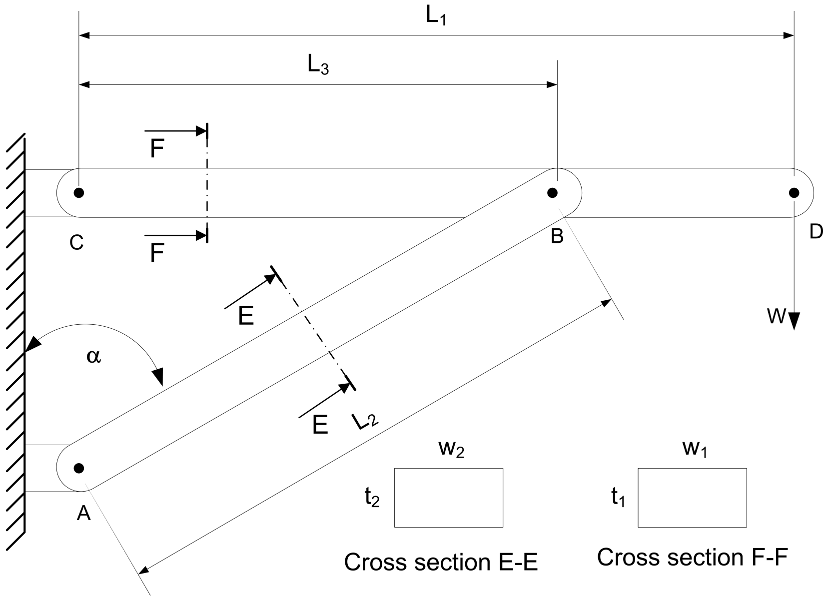

Figure 4.5: Two Member Truss

The model is adapted from earlier research works by Wood & Antonsson (1989), Scott & Antonsson (2000) and Yannou *et al.* (2007). The initial problem as described in the texts is the design of a mechanical structure that would bear a suspended load W at its overhanging end. The model consists of two beams CD and AB which are restrained by the pin joints with the wall supports and a pin joint between each other. The original model as presented in the previous research has parameters expressed in terms of the beam CD and a fixed angle between the wall and the beam AB. The objective of the design problem is to find the sets of solution pertaining to the dimensions of the beams (length, width, breadth) and the weight to be supported. In the initial problem, the dimensions of the beam CD are dependent on the dimensions of the beam AB. In order to add the depth and complexity to the problem as well as making it more uncoupled, the problem has been redesigned by adding individual dimensional parameters to each truss as well as decoupling the fixed pin joint location between the truss CD and AB. Also, the angle between the truss AB and the wall support has been decoupled. This results in the additional design variables related to the dimension of the beam CD. The variables including the design and noise variables used in the model are presented in table 4.3. Additional intermediate variables related to the choice of material are used as well such as different physical and mechanical properties related to each material.

Symbol	**Type**	**Description**	**Domain**
l_1	Real	Length of CD	[3,4]m
w_1	Real	Width of CD	[0.04,0.13]m
t_1	Real	Thickness of CD	[0.04,0.10]m
l_2	Real	Length of AB	[2,4]m
w_2	Real	Width of AB	[0.04,0.13]m
t_2	Real	Thickness of AB	[0.04,0.10]m
l_3	Real	Position of pin joint AB-CD	[2.8,3.2]m
M	Discrete	Material of members	[Steel1, Steel2, Alu1]
W	Real	Weight to be supported	[15 000,20 000]N
M_{Max}	Real	Max. mass of system	3400Kg
F_{Max}	Real	Max. force in system	Design Constraint
σ_{Max}	Real	Max. Stress	Design Constraint
∂_{l_1}	Real	Noise variable	[-0.003,0.003]m
∂_{w_1}	Real	Noise variable	[-0.003,0.003]m
∂_{t_1}	Real	Noise variable	[-0.003,0.003]m
∂_{l_2}	Real	Noise variable	[-0.003,0.003]m
∂_{w_2}	Real	Noise variable	[-0.003,0.003]m
∂_{t_2}	Real	Noise variable	[-0.003,0.003]m
∂_{l_3}	Real	Noise variable	[-0.003,0.003]m
∂_{M}	Real	Noise variable	[-0.003,0.003]m
∂_{mat}	Real	Noise variable	2%

Table 4.3: Variables for the two members beam example.

An analytical model based on these parameters has been developed for verification of the following constraints:

- $Max(W_{system})$: Maximum allowable weight of the system
- σ_b: Maximum bending stress in the truss CD must be less than or equal to the allowable bending stress limit of the corresponding material.
- Maximum Compressive force F_{AB}: Maximum allowable compression force in truss member AB. This force should not exceed that buckling force F_b.
- Material Cost: The maximum material cost for fabricating the assembly should not exceed the client's constraint.

It is assumed that the mechanical properties of the material and the section geometry remain uniform throughout the length of the beam. The stress and strain in the beam are considered to be in the elastic region and determined by classical strength material theory without end effect (Saint-Venant Principle).

The constraints are mathematically described as follows:

$$\sigma_b = \frac{3(l_1 - l_3)(2W + g(l_1 - l_3)\rho t_1 w_1)}{w_1 {t_1}^2} \tag{4.31}$$

$$W_{sys} = (t_1 \rho g l_1 w_1) + (t_2 \rho g l_2 w_2) \tag{4.32}$$

$$F_b = \frac{\pi^2 E w_2 {t_2}^3}{12 l_2} \tag{4.33}$$

$$F_{AB} = g l_2 \rho t_2 w_2 \sqrt{1 + \frac{{l_1}^2 (w + \frac{1}{2} g l_1 \rho t_1 w_1)^2}{(g l_2 \rho t_2 w_2 l_3)^2 (1 + \frac{{l_3}^2}{{l_2}^2})} + \frac{l_1(2W + g l_1 \rho t_1 w_1)}{g l_2 \rho t_2 w_2 l_3}} \tag{4.34}$$

$$s_\sigma = \frac{\sigma_r}{\sigma_b} \tag{4.35}$$

$$s_f = \frac{F_b}{F_{AB}} \tag{4.36}$$

$$s = \min(s_\sigma, s_f) \geq 1 \tag{4.37}$$

A mathematical and logical model, based on these variables and constraints, is developed by using the formalization in 3.4. The developed model can be written as the following expression for the consistency of existence of a solution:

$$\exists l_1 \in d_{l_1}, \exists w_1 \in d_{w_1}, \exists t_1 \in d_{t_1}, \exists l_2 \in d_{l_2},$$

$$\exists w_2 \in d_{w_2}, \exists t_2 \in d_{t_2}, \exists l_3 \in d_{l_3}, \exists M \in d_M, \exists W \in d_W :$$

$$W_{sys} \leq W_{max} \tag{4.38}$$

$$s \geq 1 \tag{4.39}$$

Similarly, for the robust solution consistency, the model may be described by the following expression:

$$
\begin{aligned}
&\forall l_1 \in d_{l_1}, \forall w_1 \in d_{w_1}, \forall t_1 \in d_{t_1}, \forall l_2 \in d_{l_2},\\
&\forall w_2 \in d_{w_2}, \forall t_2 \in d_{t_2}, \forall l_3 \in d_{l_3}, \forall M \in d_M, \forall W \in d_W :\\
&\forall \delta_{l_1} \in d_{\delta_{l_1}}, \forall \delta_{w_1} \in d_{\delta_{w_1}}, \forall \delta_{t_1} \in d_{\delta_{t_1}}, \forall \delta_{l_2} \in d_{\delta_{l_2}},\\
&\forall \delta_{l_3} \in d_{\delta_{l_3}}, \forall \delta_M \in d_{\delta_M}, \forall \delta_W \in d_{\delta_W},\\
&W_{sys} \leq W_{max} \qquad (4.40)\\
&s \geq 1 \qquad (4.41)
\end{aligned}
$$

The above expression are then transformed into interval based form as described in the previous sections and used in conjunction with the algorithm developed. The algorithm performs the search for the robust solution and returns the results.

4.6.1.1 Results

For the given assembly, the algorithm returned the robust results for the example problem with integrated robustness within three iterations. Figures 4.6-4.11 show the process through the different iterations of the algorithm in three dimensional projections between three variables l_1, w_1 and t_1. Figure 4.6 shows the initial design domain in terms of the variables l_1, w_1 and t_1.Figure 4.7 shows the domain reduction after the first iteration.

The empty space in the 3D space in figures 4.7-4.11 shows the design space that has been rejected during the first iteration due to the failure of the concerned BSDs to be evaluated for the consistency of the existence of solution thereby showing that the space does not contain a solution that satisfies the defined model. The BSDs depicted by the green cubes (gray in monochrome) represent the BSDs that have validated the solution existence consistency. At the end of the 1st iteration, no BSD has however validated the robust solution consistency check in integrality.

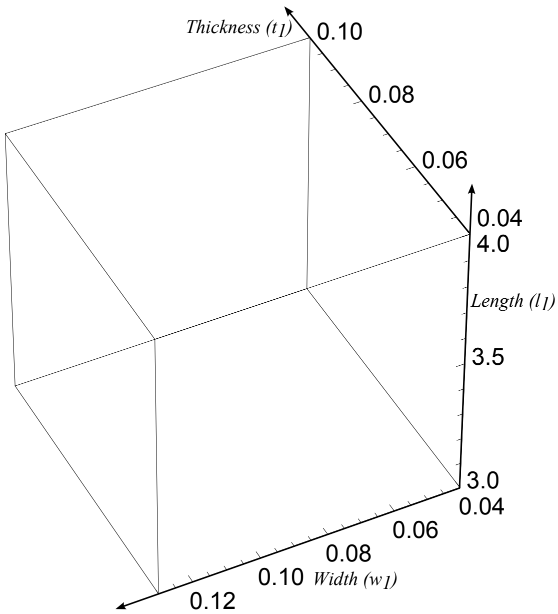

Figure 4.6: Two Member Truss - Algorithm - Initial Design BD

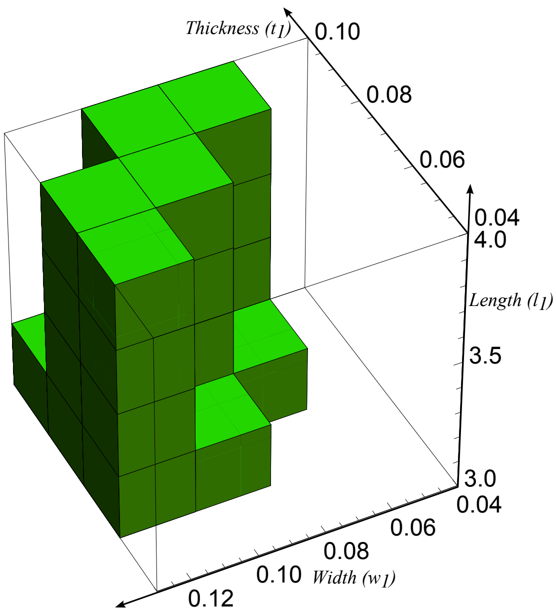

Figure 4.7: Two Member Truss - Algorithm - 1st Iteration results

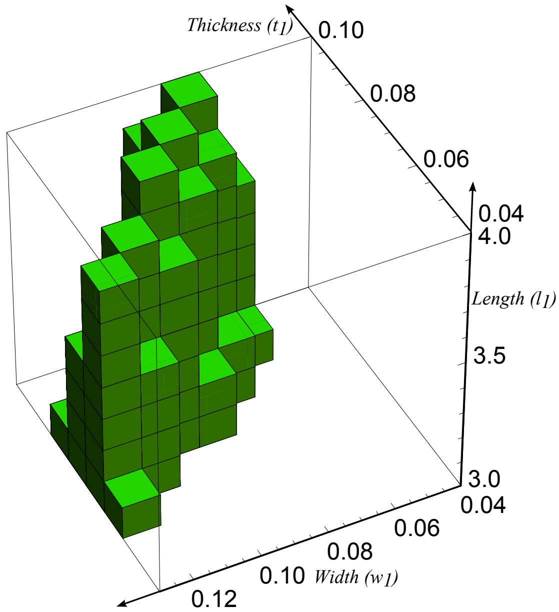

Figure 4.8: 2nd Iteration results

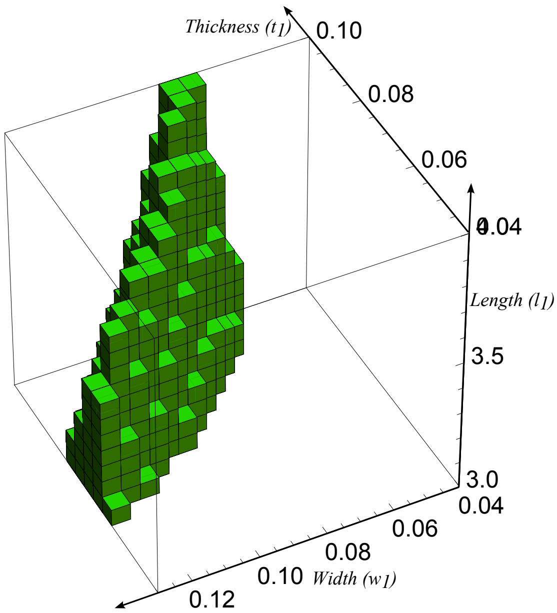

Figure 4.9: 3rd Iteration results

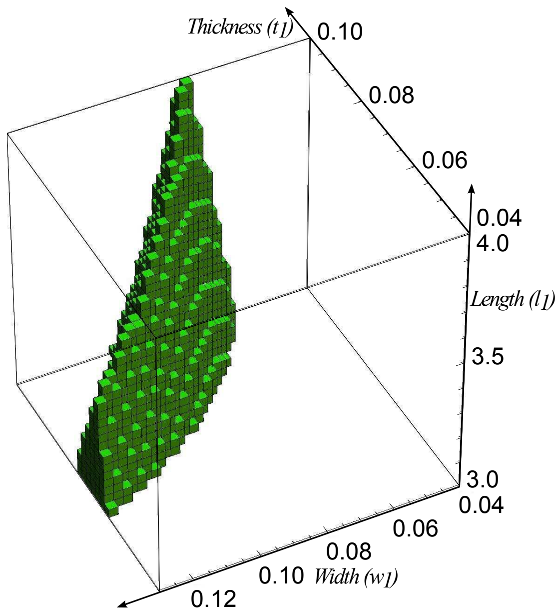

Figure 4.10: 4th Iteration results

Having searched and categorized the integrality of the design space for the 1st iteration, the algorithm than proceeds to the next iteration and evaluates the BSDs retained for the possibility of a solution with a finer resolution by sub dividing each BSD obtained from the previous iteration. Figure 4.8 shows the results obtained from this iteration. It can be observed that the number of BSDs, and their resolution, both have increased. The space rejected by the solution consistency check has also increased, reducing considerably the design space and thus narrowing down towards a solution.

The third and fourth iterations result in further design space rejection and a higher resolution of the BSDs due to their consequent division. This is shown in figures 4.9-4.10. The first robust solutions appear during the fourth iteration.These are the BSDs that have been validated for the robust solutions through the robust solution consistency conditions as described by equation 4.27. These BSDs are depicted by Blue cubes in the figure (dark gray cubes in monochrome print) and are shown in figure 4.11.

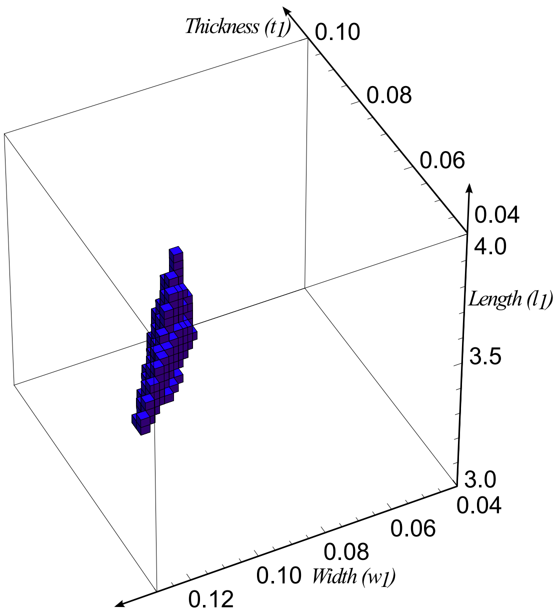

Figure 4.11: Robust solution sets for the truss structure

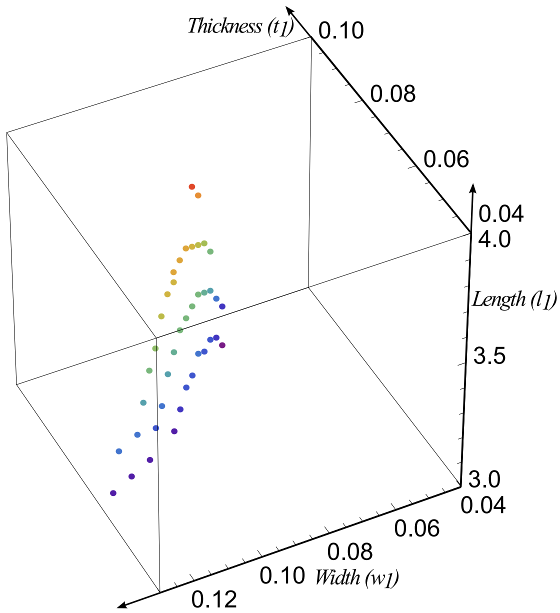

Figure 4.12: Discrete valid domain map for the truss structure

4.6.1.2 Results verification

In order to validate the results obtained from the algorithm, a comparison of the robust solutions is carried out with the discrete mapping of the design space for valid solutions. For this purpose, the initial design space is exhaustively mapped for the regions that satisfy the design constraints by taking points in the design space at regular intervals. The results of the valid design space is shown in figure 4.12. All the points in the space represent the valid solutions for the design problem in terms of the length, width and thickness of the beam AB.

Figure 4.13 shows the robust solution sets with an overlay of the discrete domain map. It can be concluded clearly that the sets of robust solutions as returned by the algorithm lie within the valid design space and thus validate the results. Figures 4.14 - 4.16 show the same overlay in the form of 2D projections between the variables l_1, w_1 and t_1.

This example is relatively more complex than the illustrative example. It deals with more complex constraints and involves a total of 18 variables with further secondary and intermediate variables. It also incorporates basic handling of dis-

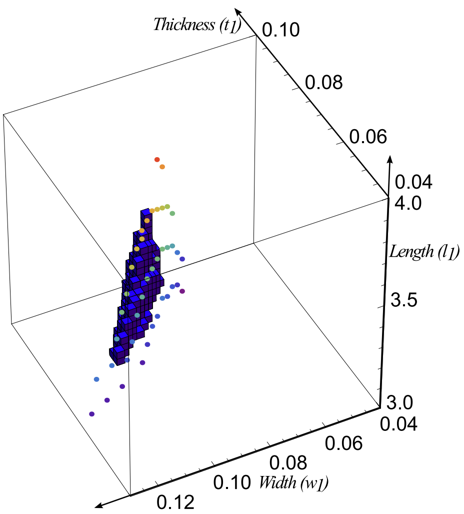

Figure 4.13: Results verification - 3D Overlay

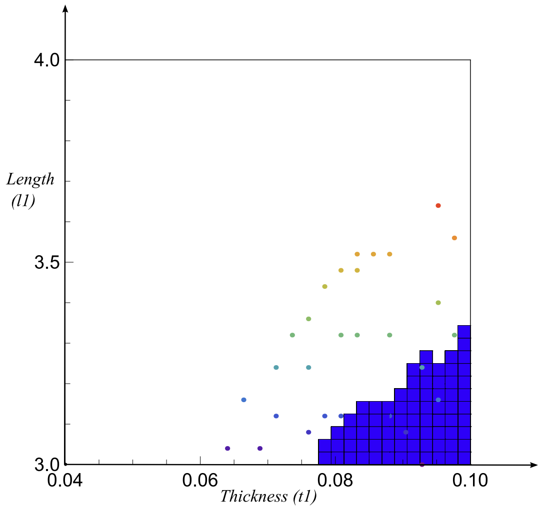

Figure 4.14: 2D Projection between l_1 and t_1

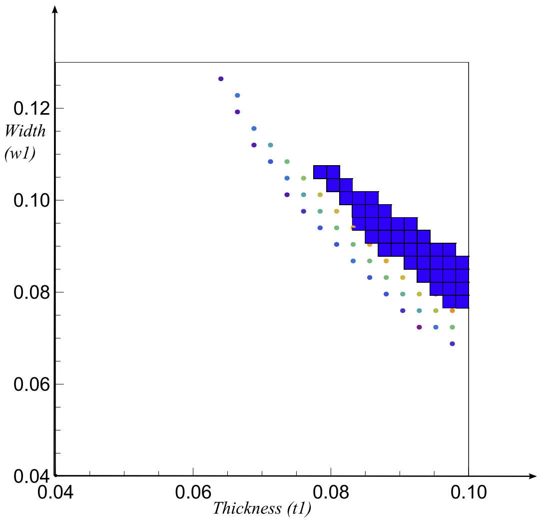

Figure 4.15: 2D Projection between w_1 and t_1

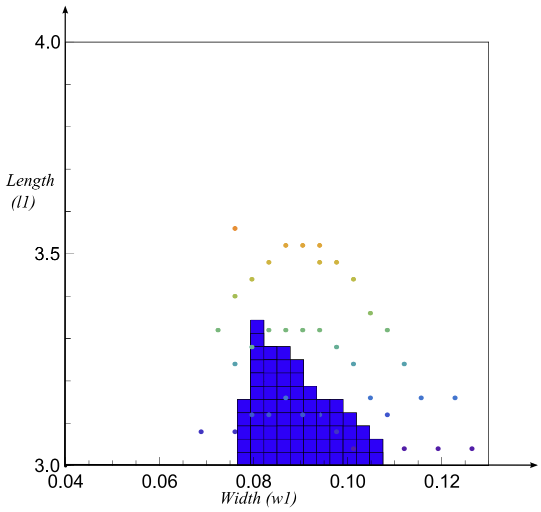

Figure 4.16: 2D Projection between l_1 and w_1

crete variables in the form of material choice. The example has been successfully validated through a comparison with the discrete domain mapping for a solution of the same design space. This has been done by running the two programs on the same computer. It was found that the developed algorithm outperforms the exhaustive method in terms of time by a large margin thereby allowing faster convergence towards solutions. The example also allows a comparison with the existing research works that have used the same example. The results returned by this approach are satisfactory, in comparison with the results provided by the earlier research works, and integrate the robustness inherently which has been addressed separately in previous cases. Instead of point based design, the results in this example are multiple and set based, allowing the designer a greater flexibility and choice in the selection of a final design solution.

4.6.2 Embodiment design of a rigid flange coupling

This example discusses the design problem of a flange coupling for the search of sets of robust solutions. A generic rigid flange coupling is shown in figure. 4.17.

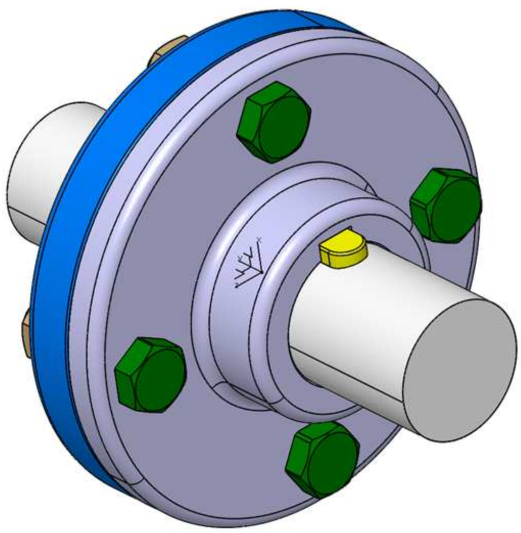

Figure 4.17: Flange coupling model assembly

The coupling is to be used to connect two shafts for torque transmission in varied applications. It may be used to connect a prime mover such as a small steam turbine or an electric prime mover such as a motor to the driven machinery such as a pump or a compressor etc. A coupling maybe of different types such as rigid or flexible. In this case, the example considered is of rigid type. The prime consideration for the coupling is to transmit the power between the connected shafts in a reliable and safe manner with the lowest possible loss of transmission as well as the optimum cost versus quality balance. The above mentioned factors being the prime considerations, the example presented provides a design methodology that integrates these requirements for a solution which remains consistent with the reliability, performance and safety requirements while being economical at the same time. The example is inspired from earlier research work on the selection of bolts for a coupling (Yvars *et al.*, 2009).

4.6.2.1 Problem description

A rigid flange coupling is to be designed for transmitting 39.5 kW of power from a four pole AC synchronous motor to a centrifugal pump. The shaft is made of steel alloy, flanges out of Cast Iron and bolts out of Steel. The permissible stresses are given as:
Shear stress on shaft (τ_s)=100MPa
Yield stress on shaft (σ_{y_s})=250MPa
Shearing stress on cast iron (τ_f)= 200MPa

4.6.2.2 Design constraints

Once the requirements have been decided upon, the design constraints can then be laid down to ensure the adherence of the design process to the required criteria. the following main relationships can be established.

- The Performance requirement is translated by the torque to be transmitted.
- The safety and reliability requirement is translated by designing the coupling in a robust way to ensure the capacity of the coupling to transmit the torque while remaining within the zone of safe mechanical operations as given by the torque requirements and taking into account the uncertainty related to the design parameters.

- The dimensional design of the coupling should allow ease of assembly and disassembly using standard tools available with consideration to the studs/bolts being used.

4.6.2.3 Flange design

In order to establish the fundamental design model, it is considered that the material is homogeneous, isotropic and purely elastic. The holes drilled in the coupling are perfectly aligned and the coupling axes are concentric. The bolts used are assumed to have uniform mechanical properties. The elements are free of surface defects. Friction between surfaces follows the Coulomb law. All constraints are to be explored and no prior knowledge about the constraints effects exists. The design variables to be evaluated are mentioned in figure 4.18 and are the key dimensional parameters of the flange as well as the selection of type and number of bolts.

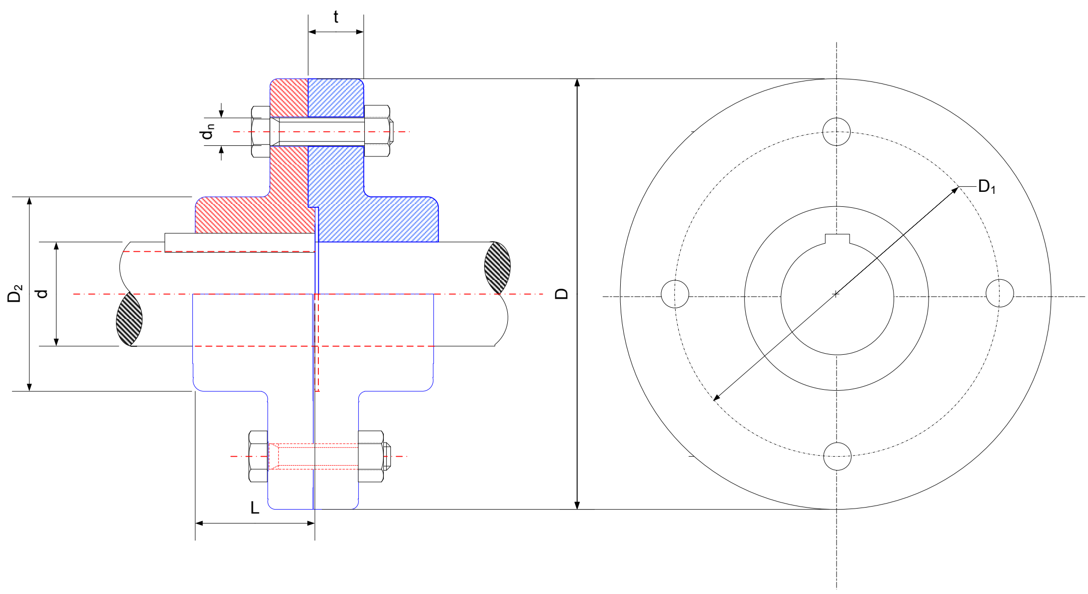

Figure 4.18: Flange coupling model assembly

Table 4.4 shows the main design variables used in the example with their starting sets and types. Out of the 14 design parameters selected, 7 are continuous variables whereas 7 are discrete. The discrete variables may have additional defined attributes such as different material properties related to a specific bolt/flange material. Additional nomenclature related to the symbols used in the design model is presented in appendix A.2

Symbol	Type	Description	Domain
t	Real	Thickness of flange	[0.0015,0.02]m
D	Real	Outside Diameter of Flange	[0.035,0.13]m
D_1	Real	Bolt Circle Diameter	[0.03,0.11]m
D_2	Real	Hub Outside Diameter	[0.03,0.09]m
μ	Real	Coefficient of friction between flange surfaces	[0.1,0.55]
f_1	Real	Bolt Coefficient of friction	[0.04,0.10]
f	Real	Bolt Pre load force	Design Constraint
i	Discrete	Number of bolts	[3,4,5,6]
d_n	Discrete	Bolt nominal Diameter	ISO M bolts
mat_b	Discrete	Bolt Material	Bolt Classes
p	Discrete	Thread Pitch	ISO M bolts
d_2	Discrete	Pitch diameter	ISO M bolts
m_b	Discrete	Bolt edge clearance	ISO M bolts
p_b	Discrete	Bolt tool clearance	Tool Charts
∂_t	Real	Noise variable	[-0.001,0.001]m
∂_D	Real	Noise variable	[-0.001,0.001]m
∂_{D_1}	Real	Noise variable	[-0.001,0.001]m
∂_{D_2}	Real	Noise variable	[-0.001,0.001]m
∂_μ	Real	Noise variable	± 2.5%
∂_{f_1}	Real	Noise variable	± 2.5%
∂_f	Real	Noise variable	± 2.5%
∂_{mat_b}	Real	Noise variable	± 2.5%

Table 4.4: Variables for the coupling design example.

In order to model the noise / uncertainty in the model, eight noise generating variables are defined related to the design variables. All the other information related to the intermediate variables and references to the discrete variables have been taken from relevant ISO/U.S. Standards related to bolts and tools.

The basic analytical model of the coupling dictating the required constraints according to the guidelines as laid down in Norton (2005) and Shigley *et al.* (2003) can now be described. Appendix A.2 describes the development of the constraints, related to the design problem in detail, which are then used to write the design model as described in the next section.

4.6.2.4 Design model

Having developed the analytical model of the coupling along with the identification of the design variables, the consistency for the existence of solution and for the robust solution can now be expressed.

Solution Consistency Using the constraints developed for the example, the expression for the consistency of the existence of the solution can be expressed by equation (4.42).

$$\begin{aligned}
&\exists t \in d_t, \exists D \in d_D, \exists D_1 \in d_{D_1}, \exists D_2 \in d_{D_2}, \\
&\exists \mu \in d_\mu, \exists f_1 \in d_{f_1}, \exists f \in d_f, \exists i \in d_i, \exists d_n \in d_{d_n}, \exists mat_b \in d_{mat_b} \\
&\exists p \in d_p, \exists d_2 \in d_{d_2}, \exists m_b \in d_{m_b}, \exists p_b \in d_{p_b} : \\
&T_{b_{bearing}} \geq T \\
&\tau_{f_{calculated}} \leq \tau_f \\
&L \geq 1.5d \\
&m \geq 3 \\
&T_{hub} \geq T \\
&T_{friction} \geq T \\
&D \geq D_1 + 2b_{A/C} + 2m_b \\
&T_{b_{shear}} \geq T \\
&0.9\sigma_y \geq \sigma eq_{max} \\
&s_c \geq p_b
\end{aligned} \tag{4.42}$$

Robust Solution Consistency Similarly, the consistency for the existence of robust solutions is expressed by the expression as stated in equation (4.43):

$$\forall t \in d_t, \forall D \in d_D, \forall D_1 \in d_{D_1}, \forall D_2 \in d_{D_2}, \tag{4.43}$$
$$\forall \mu \in d_\mu, \forall f_1 \in d_{f_1}, \forall f \in d_f, \forall i \in d_i, \forall d_n \in d_{d_n}, \forall mat_b \in d_{mat_b}$$
$$\forall p \in d_p, \forall d_2 \in d_{d_2}, \forall m_b \in d_{m_b}, \forall p_b \in d_{p_b}$$
$$\forall \partial_t \in d_{\partial_t}, \forall \partial_D \in d_{\partial_D}, \forall \partial_{D_1} \in d_{\partial_{D_1}}, \forall \partial_{D_2} \in d_{\partial_{D_2}},$$
$$\forall \partial_\mu \in d_{\partial_\mu}, \forall \partial_{f_1} \in d_{\partial_{f_1}}, \forall \partial_f \in d_{\partial_f}, \forall \partial_{mat_b} \in d_{\partial_{mat_b}} :$$

$$T_{b_{bearing}} \geq T$$
$$\tau_{f_{calculated}} \leq \tau_f$$
$$L \geq 1.5d$$
$$m \geq 3$$
$$T_{hub} \geq T$$
$$T_{friction} \geq T$$
$$D \geq D_1 + 2b_{A/C} + 2m_b$$
$$T_{b_{shear}} \geq T$$
$$0.9\sigma_y \geq \sigma eq_{max}$$
$$s_c \geq p_b$$

These expressions are checked for consistency through the transformations discussed in section 4.3. Once the transformation is done, the next step is to launch the algorithm explained in section 4.4and described in Algorithm 4.1.

The coupling example, being a mixed problem containing continuous and discrete variables,it needs a strategy for effective branching and bounding of the discrete and continuous design space. In this case, the continuous variables are branched first and then the discrete variables are evaluated for solution and robust solution consistency.

4.6.2.5 Results

The results obtained for the given example are shown in the form of three dimensional projections between three variables D, D_1 and D_2. In Figure 4.19 (a-d), the main box represents the initial BD projected in terms of the three selected variables with the starting intervals along their respective axes. In Figure 4.19 (a),

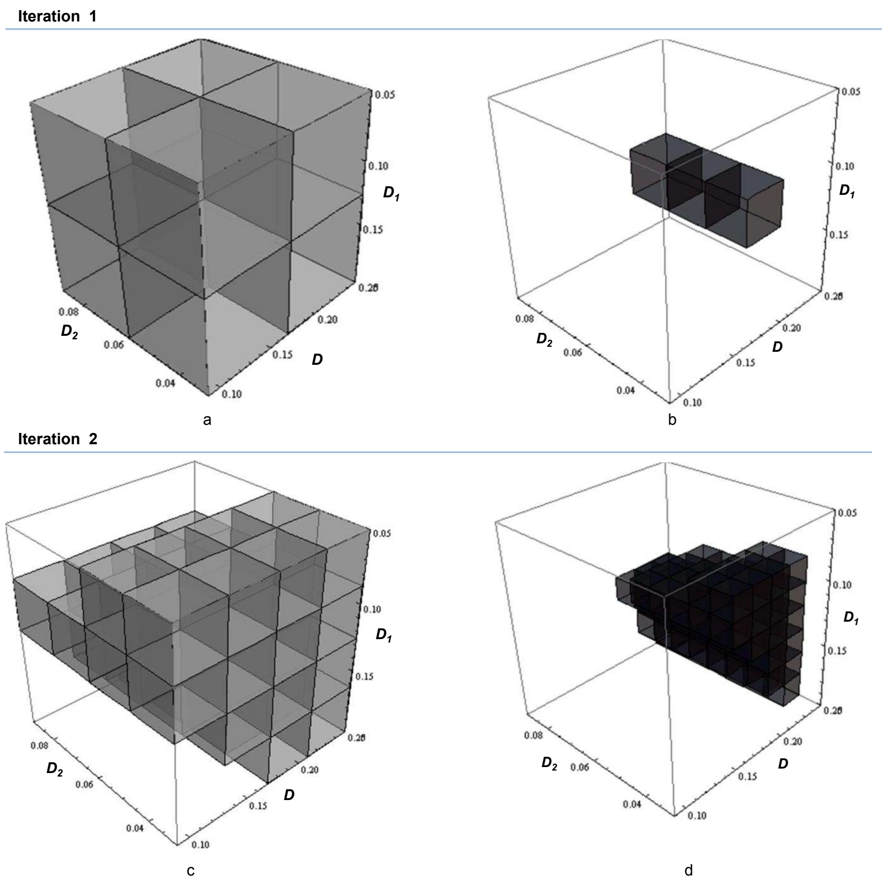

Figure 4.19: Flange coupling example results

light gray boxes after the first iteration show the possible search space (BSDs) marked by the algorithm for validated solution consistency. Figure 4.19 (b), shows the sets of robust solution within the search space in the form of dark gray boxes found after the first iteration consistent for a robust solution. In a similar fashion, Figure 4.19 (c) and (d), show the results for consistency of the solution and consistency of the robust solution in the 2nd iteration. The choice of the discrete variables can also be shown in a similar way as shown in Fig. 4.20.

The application of the developed theory and formalization of the problem of mixed constraints shows the capacity of the approach to handle the problems

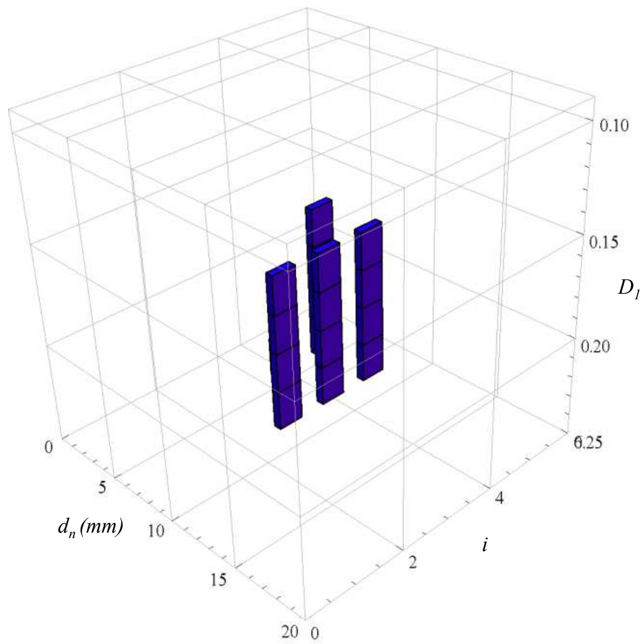

Figure 4.20: Flange coupling example-Projection between real and discrete variables

containing a mix of discrete and continuous design variables. It has also shown the possibility of a standard based catalog design selection procedure capability for the approach as many of the parameters in the design problems were restricted to a selection from mechanical design standards and subject to the choices made in the algorithm where the design problem was restricted to choose from a set of standard options. This validates the possibility of carrying out catalog based design by the developed approach . In conjunction with the discrete parameters, this example also shows the ability of the approach to handle the discrete and continuous variables together to satisfy the design constraints. This example also demonstrates a simultaneous approach towards dimensional as well as performance based design of the system thereby ensuring that the final design rests on the robustness as well as operational capacity of the mechanism.

This example, however, also highlights one of the issues related to the handling of the problems with mixed discrete and continuous variables. With the increase in the number of discrete variables in conjunction with the continuous variables, the algorithm handling the branching and pruning of the design space faces a combinatorial explosion in the case of an exhaustive search algorithm and therefore risks increasing the time of the algorithm substantially.

In order to optimize the time and avoid combinatorial explosion, a discrete variable handling strategy in conjunction with the continuous domain pruning algorithm is required that may manage the decomposition of the BD in BSDs more efficiently in terms of mixed problems.

It is also possible to export the numerical tables for the results which can then be utilized for analysis or representation of the results. Table 4.5 shows one of the robust solution sets found among other robust solution sets. These sets were verified by individual constraint verification and found to be satisfying all the constraints.

Variable	**Robust Set**
D	[0.21,0.25]m
D_1	[0.125,0.1625]m
D_2	[0.045,0.06]m
T	[0.006125,0.01075]
μ	[0.13825,0.2755]
F_1	[0.12,0.1775]
i	4
d_n	8mm
$Mat.Class$	2

Table 4.5: One of the set based robust solutions for the coupling example

4.6.3 Design of a 6 Bar Mechanism

The third example, used to demonstrate the application of the developed theory, is a six-bar mechanism. This is shown in figure 4.21

The objective is to carry out the dimensional design of a six-bar mechanism such that the required assemblability conditions for the mechanism and path generation requirements for point G are fulfilled. This example is selected due to its relevance to the engineering field of parameter design and kinematics. From a mathematical point of view, this example includes some mathematical modeling of continuous and discrete variables as well as complex, cyclic, derivable,

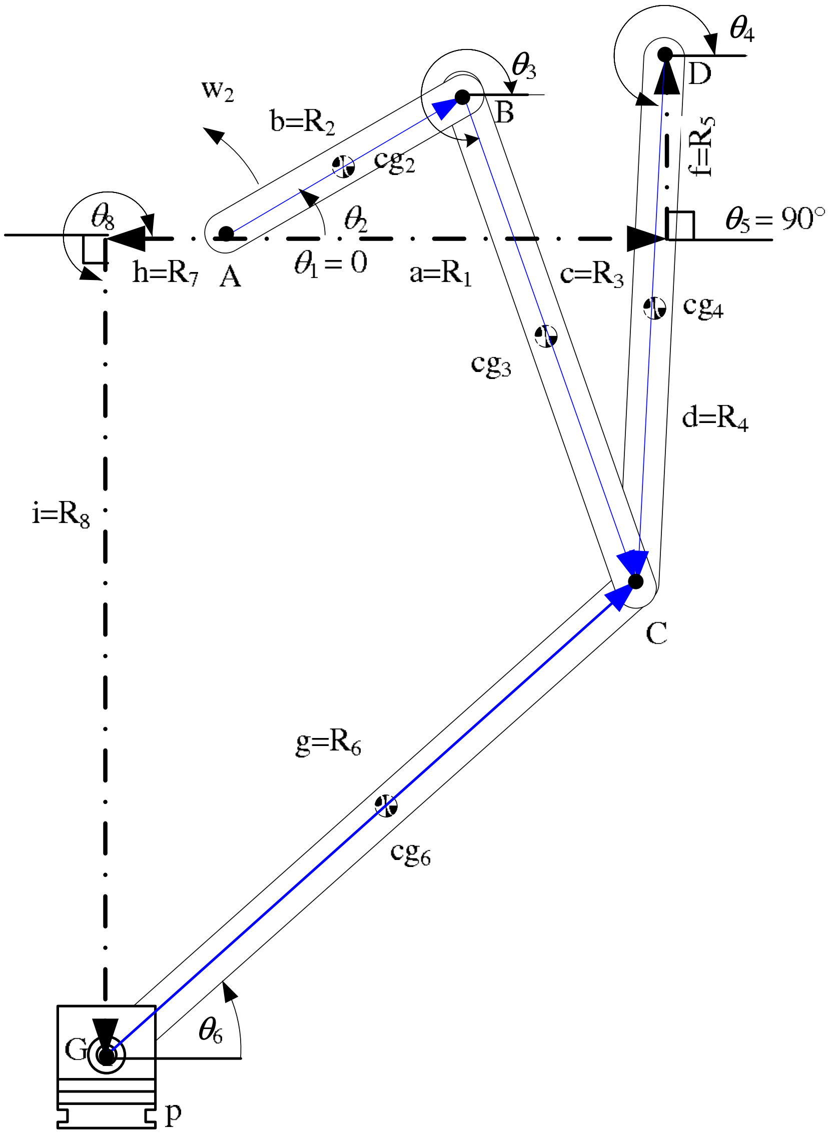

Figure 4.21: Six-Bar Mechanism dwell mechanism

temporal, and trigonometric functions for positioning, path generation and curve tracing.

4.6.3.1 Problem Description

The mechanism shown in Figure 4.21 is required to operate and come fit into a frame of 5800mm x 3800mm with the following supplementary information:

Strokes/min Strokes per minute describes the amount of time that a press strokes down in one minute. The minimum number required is 50 strokes/minutes.

Dwell Period The dwell period, defined as "The number of degrees of the driver link (R_2), during its one full rotation, for which point G on the driven link (R_6) remains closed (extreme position)", should not be less than 100°.

Stroke Length The maximum vertical displacement in the position of point G on link R_6 should not be less that 250mm during one full rotation of the driver link R_2.

Dwell Variation Dwell variation is the change in the vertical position of the point G during the dwell period. This variation should not be more than 1% of the stroke length.

4.6.3.2 Design Constraints

Once the requirements have been decided, a model can be constructed that defines the fundamental design requirements translated through an appropriate analytical model featuring interaction with the key design variables and constraints that define the limits of the design problem. Such an analytical model will simulate the kinematic positioning of the mechanism members as a function of the θ_2. In order to find the mathematical expressions which translate the problem requirements in the form of design constraints, the following three conditions need to be satisfied.

- The Six-Bar Mechanism should assemble
- The Six-Bar Mechanism should fit in the frame
- The Six-Bar Mechanism should trace the motion as per design requirement

In order to develop the three conditions above, using vector analysis, assembly analysis as well as position analysis of the assembled mechanism needs to be carried out. This results into three main constraint groups i.e assembly constraints, framing constraints and path generation constraints. These are developed and discussed in detail in Appendix A.3.

4.6.3.3 Design Model

Once the constraints for the mechanism related to assemblability, fitting, positioning and path generation have been developed, using these constraints in conjunction with the design requirements, a design model can be defined. This model, containing design parameters and constraints, can be expressed with the developed formalization. The first step is to decide upon the key design variables. The variables involved in the design model are presented in table 4.6. Using the

Symbol	Type	Description	Domain
a	Real	X-axis displacement between A and D	[2.0,2.06]m
b	Real	Length of link R_2	[1.0,1.03]m
c	Real	Length of link R_3	[3.0,3.09]m
d	Real	Length of link R_4	[3.0,3.09]m
f	Real	Y-axis displacement between A and D	[1.0,1.03]m
g	Real	Length of link R_4	[6.0,6.18]m
h	Real	Y-axis displacement between A and D	[-1.0,1.0]m
X	Real	Length of Frame	3.8m
h	Real	Height of Frame	5.8m
$\theta_{contact}$	Real	Min. Contact Angle	1.744 Radians
S_{min}	Real	Minimum Stroke	0.250m
δ_{dwell}	Real	Maximum dwell delta	1% of Stroke

Table 4.6: Variables for the six bar mechanism

variable defined in the table 4.6, we can now write the expressions related to the solution consistency and robust solution consistency.

Solution Consistency For the set of developed constraints and the variables involved, the solution consistency can be defined by the following expression.

$$\begin{aligned} &\exists a \in d_a, \exists b \in d_b, \exists c \in d_c, \exists d \in d_d, \\ &\exists f \in d_f, \exists g \in d_g, \exists h \in d_h : \\ &\sqrt{a^2+f^2}+b < c+d \\ &\sqrt{a^2+f^2} < (b+c+d) \\ &h < b+c+g \\ &b+h < c+g \end{aligned} \tag{4.44}$$

$$\max \left\{ \begin{array}{l} \theta_2 \in [0, \pi], \max[\max[b \sin \theta_2, f] + i(\theta_2)] \\ \theta_2 \in [\pi, 2\pi], \max[f + i(\theta_2)] \\ \theta_2 \in [0, 2\pi], \theta_4 \in [0, \pi], \max[f + d \sin \theta_{4_{\theta_2}} + i(\theta_2)] \end{array} \right\} < y$$

$h \in [-\infty, 0]$:

$$\max \left\{ \begin{array}{l} \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ |\bar{h}| < \bar{b}, \max[b \cos \theta_2 + a + d \cos \theta_{4_{\theta_2}})] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ |\bar{h}| < \bar{b}, \max[b \cos \theta_2 + a] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ |\bar{h}| > \bar{b}, \max[h + a + d \cos \theta_{4_{\theta_2}})] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ |\bar{h}| > \bar{b}, \max[h + a] \end{array} \right\} < x$$

$h \in [0, +\infty]$:

$$\max \left\{ \begin{array}{l} \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ \bar{h} < \bar{a}, \max[b \cos \theta_2 + a + d \cos \theta_{4_{\theta_2}})] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ \bar{h} < \bar{a}, \max[b \cos \theta_2 + a] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ \bar{h} > \bar{a}, \max[b \cos \theta_2 + \max[h, d \cos \theta_{4_{\theta_2}}]] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ \bar{h} > \bar{a}, \max[b \cos \theta_2 + h] \end{array} \right\} < x$$

$$\theta_3 \in \mathbb{R}$$

$$\theta_4 \in \mathbb{R}$$

$$S_{min} \geq 0.250$$

$$\delta_{dwell} \leq 0.01 S_{min}$$

The above model describes the quantified expressions detailing the solution consistency for the model. The solution consistency lays down the evaluation criteria for the search of the existence of a set such that the mechanism assembles, allows the full rotation of the driver link, fits in the frame, and generates the required motion. This therefore acts as the first evaluation criteria. Once a set is validated through the solution consistency, the next step is to evaluate the set for robust solution consistency.

Robust Solution Consistency The consistency for a robust solution that evaluates the condition for the respect of the assembly and motion generation constraints for the totality of the sets as set down by the design requirements is given by the following expression.

$$\begin{aligned}
&\forall a \in d_a, \forall b \in d_b, \forall c \in d_c, \forall d \in d_d, \\
&\forall f \in d_f, \forall g \in d_g, \forall h \in d_h, \\
&\forall \delta_a \in d_{\delta_a}, \forall \delta_b \in d_{\delta_b}, \forall \delta_c \in d_{\delta_c}, \forall \delta_d \in d_{\delta_d}, \\
&\forall \delta_f \in d_{\delta_f}, \forall \delta_g \in d_{\delta_g}, \forall \delta_h \in d_{\delta_h} : \\
&\sqrt{a^2 + f^2} + b < c + d \\
&\sqrt{a^2 + f^2} < (b + c + d) \\
&h < b + c + g \\
&b + h < c + g \\
&\max \left\{ \begin{array}{l} \theta_2 \in [0, \pi], \max[\max[b \sin\theta_2, f] + i(\theta_2)] \\ \theta_2 \in [\pi, 2\pi], \max[f + i(\theta_2)] \\ \theta_2 \in [0, 2\pi], \theta_4 \in [0, \pi], \max[f + d \sin\theta_{4_{\theta_2}} + i(\theta_2)] \end{array} \right\} < y \\
&\quad h \in [-\infty, 0] : \\
&\quad \max \left\{ \begin{array}{l} \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ |\bar{h}| < \bar{b}, \max[b \cos\theta_2 + a + d \cos\theta_{4_{\theta_2}})] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ |\bar{h}| < \bar{b}, \max[b \cos\theta_2 + a] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ |\bar{h}| > \bar{b}, \max[h + a + d \cos\theta_{4_{\theta_2}})] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ |\bar{h}| > \bar{b}, \max[h + a] \end{array} \right\} < x
\end{aligned} \tag{4.45}$$

$$h \in [0, +\infty] :$$

$$\max \left\{ \begin{array}{l} \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ \bar{h} < \bar{a}, \max[b \cos\theta_2 + a + d\cos\theta_{4_{\theta_2}})] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ \bar{h} < \bar{a}, \max[b\cos\theta_2 + a] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ \bar{h} > \bar{a}, \max[b\cos\theta_2 + \max[h, d\cos\theta_{4_{\theta_2}}]] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ \bar{h} > \bar{a}, \max[b\cos\theta_2 + h] \end{array} \right\} < x$$

$$\theta_3 \in \mathbb{R}$$

$$\theta_4 \in \mathbb{R}$$

$$S_{min} \geq 0.250$$

$$\delta_{dwell} \leq 0.01 S_{min}$$

These expressions can now be transformed through the techniques discussed earlier in section 4.3 for utilization in conjunction with the algorithm developed in section 4.4 to carry out the design space exploration for the sets of feasible solution.

4.6.3.4 Algorithm improvements

Initial simulations with the developed analytical methods, consistencies and its transformations showed that, due to the involvement of cyclic trigonometric functions and multiple occurrences, the algorithm needed improvements in terms of constraint management and interval management. For this purpose, the initial algorithm was improved by adding the sub routines of constraint anteriority for better management of constraints and bisection for the management of intervals.

Constraint anteriority The design problem of the six bar mechanism is unique from the examples treated earlier in a sense that the design problem is governed by a progressive set of constraints instead of a simultaneous set of constraints. In order to take advantage of such provisions in related problems, we introduce the concept of constraint anteriority in the algorithm. Referencing to the constraints discussed earlier, it can be established easily that only those systems which fulfill the initial assembly constraints will work. Likewise, before the complex calculations related to the path generation are to be carried out, it is necessary that the system fits in the frame as well. Lastly, knowing that the position analysis equations arise out of a set of simultaneous equations involving the member lengths and trigonometric relations in terms of θ_2, these equations need to be satisfied for a given configuration of member lengths in order for the design to be validated.

Keeping this information in mind, a constraint hierarchy algorithm has been developed, which allows for progressive evaluation of constraints subject to the validation of earlier constraints. This ability allows the program to search the design space for solution while spending a minimum amount of computational time and effort until necessary. This improves the algorithm considerably. The constraint anteriority algorithm is shown in algorithm. 4.2

Algorithm 4.2 Constraint Anteriority

Require: $C : C = \{c_1, c_2, , , c_n\}, V, D, s_i : s_i \in D$
Ensure: $s_i \models \exists \bar{v} C(\bar{v}, \bar{a})$

```
j ← 1
Check ← true
while j ≤ n  and Check =true do
    if s_i ⊨ ∃v̄ c_j (v̄, ā) then
        Check =true
    else
        Check =false
    end if
    j ← j + 1
end while
```

Interval Bisection An important step in the algorithm is the transformation of the expressions into a computable form through interval analysis as described in section 4.3.1. Converting the variables and constraints through the interval analysis had its advantages in terms of efficiency and ease of approach, but at the same time the interval approach suffers from a number of short comings from a mathematical point of view that have to be accounted for.

One such issue is dependency. In expressions involving multiple occurrences of variables, the approach suffers from the problem of dependency. The dependency problem amplifies the solution domain and therefore adds false domains to the apparent result that may in fact not hold the solution. The problem arises from the fact that in general, each occurrence of a given variable in an interval computation is treated as a different variable. This causes the widening of computed intervals and makes it difficult to compute sharp numerical results of complicated expressions (Hansen & Walster, 2004).

This unwanted extra interval width is called the dependence problem or simply dependence. This problem can be avoided if each variable in a given expression occurs once only. In such an expression, the result is the exact domain bound. This, however, is not the case in most of the complex engineering constraints, which are often implicit in nature and may contain multiple occurrences of one or more variables. A simple example to illustrate the dependency problem is presented as follows:

$$\begin{aligned} x &= [-1, 2] \\ y &= [-1, 2] \\ f_1(x, y) &= (x + y)(x - y) \\ &= [-12, 12] \\ f_2(x, y) &= x^2 - y^2 \\ &= [-4, 4] \end{aligned} \tag{4.46}$$

Despite the fact that mathematically, $f_1(x, y) = f_2(x, y)$, the results obtained by the two equations are vastly different. In fact the result returned by $f_1(x, y)$ is false as it contains imprecise bounds resulting from multiple occurrences of variables x and y. This problem aggravates with the inclusion of trigonometric functions of the type $f(x) = \cos x, f(x) = \sin x$ etc. which occur abundantly in the six bar mechanism problem. In order to address this problem, multiple techniques exit. In this example we use the bisection to integrate a routine in the algorithm to improve the results. The bisection technique is based on the observation that in whatever form we express a function, when we evaluate it with interval arguments; we tend to get narrower bounds on its range as argument widths decrease. One way to compute narrower range-bounds is to subdivide an interval argument and compute the union of results for each subinterval (Hansen & Walster, 2004).

For a given interval $\mathbf{x}$, we divide it into subintervals $\mathbf{x}_i (i = 1, ...m)$ so that

$$\mathbf{x} = \bigcup_{i=1}^{m} \mathbf{x}_i \tag{4.47}$$

This implies that the required function $\mathbf{f}(\mathbf{x})$ needs to be evaluated for each subinterval, and therefore, can be written as:

$$f(\mathbf{x}) \subseteq \bigcup_{i=1}^{m} \mathbf{f}^I(\mathbf{x}_i) \tag{4.48}$$

The superscript "I" on $\mathbf{f}$ emphasizes that because of dependence, the computed value is still not very sharp even if infinite precision interval arithmetic is used. However, each sub interval suffers less than the computation on the total simultaneous bound. Bisection is also more useful in the case of trigonometric functions. Through bisection, the boundary errors related to these functions can be minimized.

A routine for insertion of bisection in algorithm is shown in Algorithm 4.3

Algorithm 4.3 Interval Bisection

Require: $\mathbf{c}_i, \bar{\mathbf{v}} : \bar{\mathbf{v}} \in \mathbf{D}, res$

Ensure: Interval Bisection Over given resolution

1: $res \leftarrow m$ {Number of Required sub interval resolution}
2: $\bar{\mathbf{v}} = \bigcup_{res=1}^{m} \bar{\mathbf{v}}_{res}$
3: $\mathbf{c}_i \leftarrow \varnothing$
4: **for** $res = 1$ to m **do**
5: $\quad \mathbf{c}_i \leftarrow \mathbf{c}_i \cup \mathbf{c}_{i_{res}}(\bar{\mathbf{v}}_{res})$
6: **end for**

4.6.3.5 Results

The algorithm along with the analytical model were programmed in the Mathematica® software. The analytical model was developed to treat the three main requirements of assemblability, fitting and path generation. In order to validate the analytical model, a graphical model of the mechanism was developed for simulation. This was validated for accuracy by simulation with real variables for lengths of the links for $\theta_2 \in [0, 2\pi]$ and the nominal dimensions of $a = 2.0$, $b = 1.0$, $c = 3.0$, $d = 3.0$, $f = 1.0$, $g = 6.0$, $h = 1.0$. The line diagram of the model after programming is shown in Fig. 4.22. The colors blue, green, red and

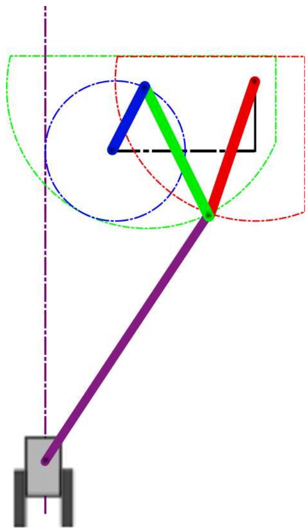

Figure 4.22: Six Bar Mechanism Mathematica Model

magenta represent the moving links R_2, R_3, R_4, R_6 respectively. The line diagram confirms the validity of the analytical model for the example.

This model is then nested within the algorithm for the design space search. The constraints needed for the assembly and positioning are programmed directly from the equations mentioned above. However, in order to evaluate the constraints related to the stroke length, the length of the dwell period and the variation in the position of point G within the dwell stage, further development of the constraints is required.

The position of the point G can be determined by the resolution of the constraints for i, however for finding the variation within the dwell, as well as the information related to the stroke of a given configuration, it is necessary to calculate the positions of G in terms of θ_2 that determine the start and the end of the dwell period as well as the extremities of point G in order to calculate the stroke. This is done through a specific sub routine allowing the numerical derivation of the curve data obtained through the positioning constraints and finding the minimas and maximas of the curve. These points can then be further manipulated to calculate the dwell variation of G within the dwell period. The length of the dwell period in terms of position of θ_2 can also be calculated through the same method. Figure 4.23 shows an example of the curve data obtained for a given BSD for calculating the start and end of the curve data along with the point of the maximum variation within the curve for the intervals. The sub routine generates the curves for the BSD under analysis. This is shown in Figure 4.23 by the two curves which represent the region of the path curve for the given BSD. Any configuration of the design variables will generate a curve that is enclosed by the two curves in the figures. Once these curves have been generated, the sub routine then finds the maximas and minimas of the curves through numerical derivation. This is shown by the points on the curves. Identifying these points then enables the calculation of the minimum and maximum stroke lengths as well as the dwell period and dwell variation for the BSD. This information is then used for the consistency evaluation of the given BSD through the modified algorithm as shown in Algorithm 4.4.

The design model along with the solution consistency, robust solution consistency and associated variables and constraints were used in conjunction with the algorithm for solution. Even with the utilization of the constraint anteriority and interval bisection techniques, the results obtained were not very sharp owing to the problems due to interval dependency arising out of complex trigonometric functions involved in the explicit and implicit equations.

The sharpness of the results obtained from the program depends to a large extent on the results for the calculation of the dependent angles θ_3, θ_4 and θ_6. The results of calculation of these dependant angles is given by the functions in terms of θ_2 which is an independent variable and the sets of lengths of the members and

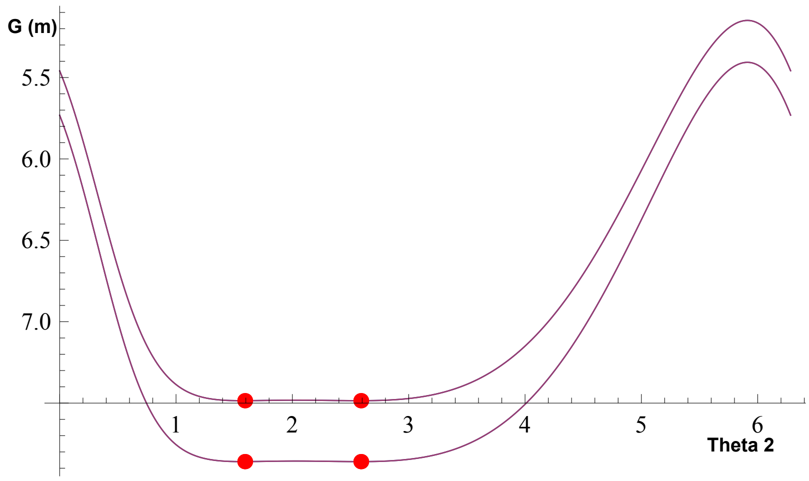

Figure 4.23: Dwell characteristic calculations (Position of point G)

their positioning. As the functions are non-linear, trigonometric and of higher powers, the effects of the interval-dependency greatly increases in the angular calculations. The results for the calculations of the dependant angles therefore contains false domains which introduce a consequent error in the calculations of the positioning of the point G and results in a large interval which cannot be further used for downstream calculations such as acceleration and dynamic analysis of the system.

Figure 4.24 shows a zoomed section of the model showing the results of the configuration of a given position corresponding to a θ_2 position. The radii of the concentric circles show the minimum and maximum values of the set of the lengths the of corresponding members (blue and red dimensions). The members R_2, R_3 and R_4 can be seen in the Figure 4.24. For the given set of members, the range of dependant angles θ_3 and θ_4 is shown by the shaded area. The θ_3 interval is shown by the area shaded in yellow and the interval for θ_4 is shown by the area shaded in green. Similarly, the interval for the position of "G" denoted by i is shown by the area shaded in blue. It can be observed that the effect of the interval dependency increase as the level of the dependant variable increases in depth. The interval calculation of θ_4 has higher bounds than that of the bounds calculated for θ_3. The intersection between the concentric circles of R_3 and R_4 show the actual interval that represents the variation space due to the variation

Algorithm 4.4 Six Bar Design Algorithm Pseudo Code

Require: $\{QV, D, C\}, res, dim, iterations$ $\{res =$Number of Required sub interval resolution, $dim =$Length of DV$\}$
Ensure: Feasible sets of robust solution

$BD \leftarrow D$
$BSD\_list = \{\bigcup_{i=1}^{res}\{d_{i_{v_j}} : j \in \mathbb{N}, j = dim\}\}$
$Exist\_list \leftarrow \varnothing$ {Set of Possible Solutions}
$Robust\_list \leftarrow \varnothing$ {Set of Robust Solutions}
$NoSol\_list \leftarrow \varnothing$ {Space Without Solution}
for $k = 1$ **to** $iterations$ **do**
  for $l = 1$ **to** Length of BSD_list **do**
    Pick BSD_l
    Convert BSD_l to interval based sets
    if constraint anteriority algorithm using bisection of BSD_l= **true then**
      if Consistency of robust solution using bisection of BSD_l= **true then**
        $Robust\_list \leftarrow Robust\_list \cup BSD_l$
      else
        $Exist\_list \leftarrow Exist\_list \cup BSD_l$
      end if
    else
      $NoSol\_list \leftarrow NoSol\_list \cup BSD_l$
    end if
  end for
  $BSD\_list \leftarrow Exist\_list$
end for
Display Results

in both members. Comparing the interval space for θ_3 and θ_4, it becomes clear that both intervals contain false domains. This effect is then amplified in the case of the position of point "G" therefore resulting in loss of the accuracy.

The use of the constraint anteriority techniques reduces the design space for the BSDs which do not contain solutions with the use of little computational effort in comparison to a conventional algorithm. The bisection technique aims to increase the sharpness of the interval results but can increase the computation time considerably in the view of increasing the resolution of the bisection.

Figure 4.25 shows the global variation domain view for the given problem.

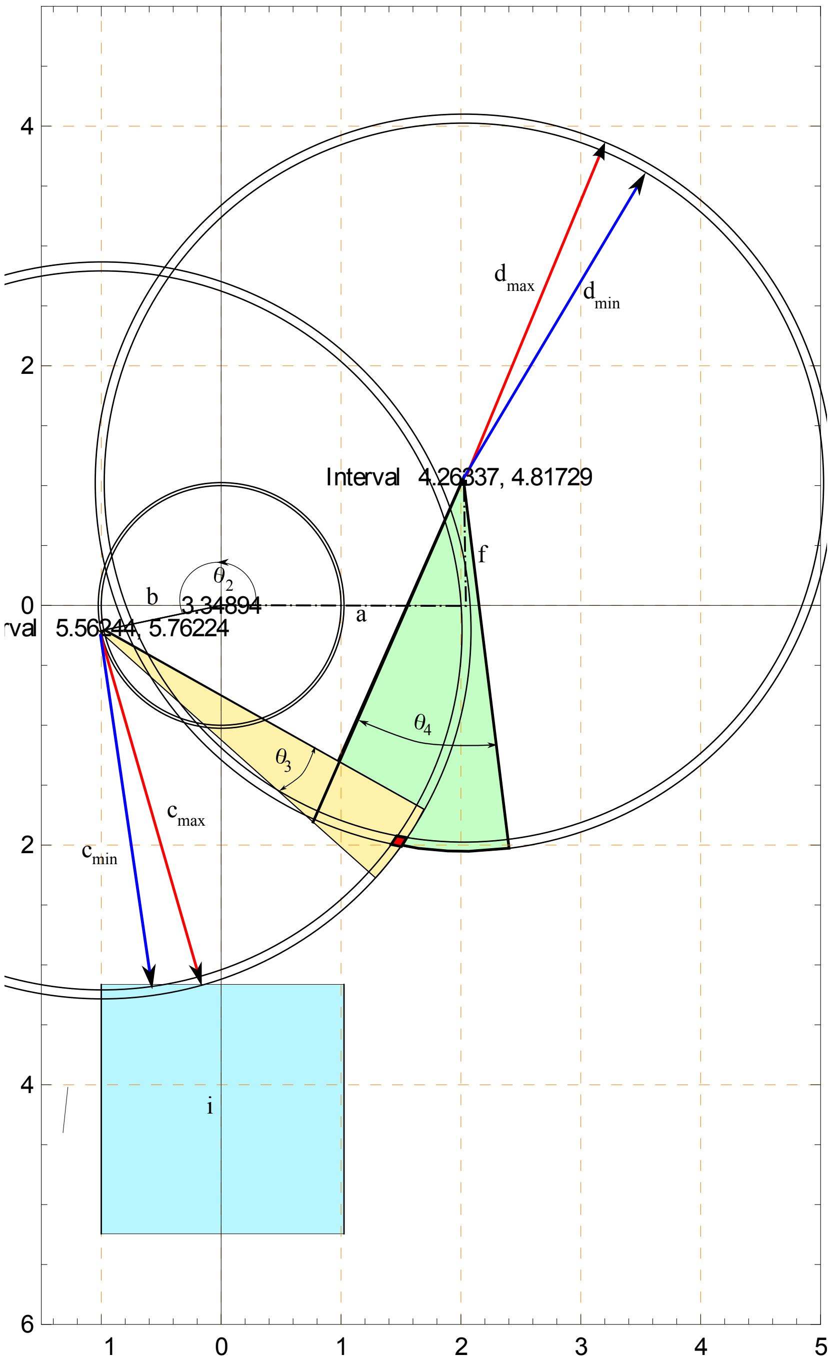

Figure 4.24: Interval Evaluation results

The area in red color corresponds to the intersection of the variation domain of the membersR_1, R_2, R_3, R_4 and R_5. The final variation domain includes the variation in the members R_6 and R_7 and is depicted in blue color. A valid and correct domain, that takes into account the effects of the interval dependency, must correspond to this variation space in order to validate the results.

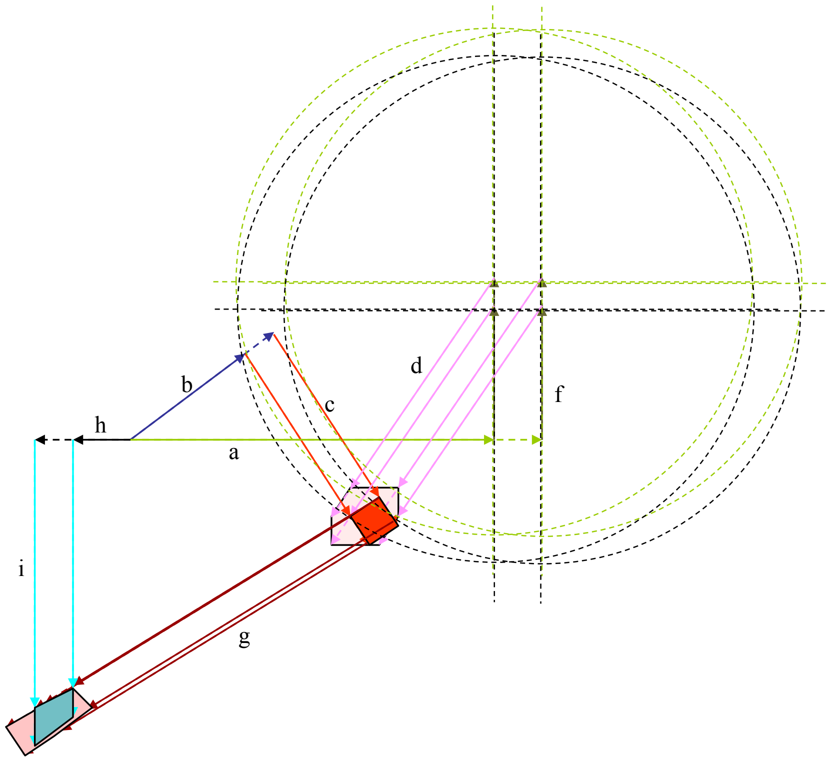

Figure 4.25: Variation domains for the six bar mechanism

4.7 Conclusion and Discussion

This chapter presents a comprehensive overview of the tools, methods, transformations and the algorithms required to apply the set based robust design formalization developed in chapter 3 to the problem of product design and variation management for mechanical products.

In order to transform this formalization in a computable form, it is necessary to provide the means for the transformation of the expression in terms of computable code. This is done by the conversion and transformation of the quantified expressions with the help of box consistency techniques and interval mathematics thus allowing the expression of the existential and universal quantifiers in a computable form.

In addition to the transformation of the formalization through the consistency techniques, it is necessary to provide a space exploration tool that is capable of taking an initially decided set based search space and search this design space in conjunction with the developed consistency techniques for the search of the feasible sets of solutions that correspond to the sets of design variables satisfying the design constraints while simultaneously taking into account the variations in the design model. This is done by the development of an algorithm based on exhaustive search and branch and bound principle to search the design space through the transformed expression expressed in terms of solution consistency and robust solution consistency. The algorithm provides the results in the visual and numeric form in a two dimensional as well as three dimensional space. As the algorithm carries out simultaneous search for the design parameter space as well as the variation space, it provides the designer with the capability of multiple solutions in parallel instead of a point based solution. This algorithm is thus well inline with the set based-design methodology of Sobek and Ward and therefore accomplishes the design with a greater flexibility.

The solutions provided with this set based approach take into account the uncertainties related to the design variables through out the modeling, evaluation, and qualification phase, therefore, the solutions obtained are inherently robust in nature and insensitive to variations in the key design variables.

In order to demonstrate the practical applications of the developed theory, three relevant examples from the field of engineering design towards diverse applications were illustrated. These three examples have allowed verification, testing and qualification of the approach. From a mathematical point of view, the approach has been successfully applied to the problems containing real, discrete and mixed variables and containing multiple constraints of linear, and non linear type. From an engineering point of view, the problems treated belong to the most common mechanical engineering problems encountered such as mechanical design, dimension design and path generation. Two out of the three examples treated have been validated with the traditional methods, validating the results obtained by the algorithm.

The work done in the example 3 has however prompted improvements in the algorithm. This improvement has been done in the form of two additional routines built in the code for handling constraint anteriority for better management of computational effort and the interval bisection method for increasing the sharpness of the resultant intervals. This development has improved the results but in spite of these improvements the results were not found to be satisfactory. The main cause for this is the inherent problem associated to the interval arithmetics of interval dependency. This arises due to the multiple occurrence of the same variables inside an expression compounded with the complex implicit expressions of trigonometric nature.

One solution to over come this problem is to increase the sub interval resolution in the bisection algorithm. It was however noted that in order for bisection technique to work and obtain sharp results, the intervals had to be split into sub intervals of very high resolutions. This is very intensive as interval splitting becomes costly in terms of computational effort and time and becomes infeasible. If a function depends on n intervals and each of these intervals is divided into half, then the function needs to be evaluated with $2n$ sets of different arguments. Considering that this technique itself is nested within a $m \times n \times o$ loop where m is the recursion depth, n is the number of variables and o is the number of cuts administered by the branch and bound algorithm, the exhaustive rapidly becomes prohibitive for a problem of even small dimensions. This can be compounded with the inclusion of trigonometric functions, which require a high resolution of interval bisection.

Another way to avoid the interval dependency issue is to factorize the problem in a way such that the multiple occurrences of the variables in the constraints is avoided. This is possible for a problem of a very simple type but even for a basic problem, this is not possible and the chances of having implicit constraints with multiple variable occurrences remains high. Another promising technique is to develop an affective variable domain splitting strategy specific to each design problem where selective variable domains are bisected for higher resolution to find a compromise between result sharpness and computational effort and time while keeping the other variables constant.

Chapter 5

Application to Tolerance Analysis

Tolerance analysis is an important step in the product design which ensures the design verification and validation. Tolerance analysis attempts to estimate the assembly tolerance stack-up i.e. the effect of individual tolerances on the assembly response function, and thereby qualifying the assembly as per the requirements set by the designers in view of the design requirements (Maropoulos & Ceglarek, 2010). It forms one of the key steps in the digital product verification by evaluating the product performance with the given set of tolerances assigned.

The formalization of the tolerance analysis in the product design phase and its expressive formulation has been discussed in chapter 3. In order to implement the formalization in a computable form, it is necessary to select and develop tools, methods and transformations proper to the field of tolerance analysis. Using these tools, the developed formalization can be transformed for application to examples in mechanical product design.

This chapter therefore presents the work carried out in the identification of a multi step process of transformation of the formalization towards a final algorithm, consisting of different methods and tools, that can be used to carry out the tolerance analysis of a mechanical assembly. This requires the selection and decision of proper tools to interpret the product geometry, followed by an appropriate computational model in terms of the tolerance analysis formalization discussed to carry out tolerance analysis.

Eventually, an overconstrained mechanism assembly is developed and presented as an example for testing and validating the tolerance analysis application developed in the chapter.

5.1 Considerations for the Application of Tolerance Analysis Formalization

The tolerance analysis problem can be divided into three main steps:

- The models representing the product geometry
- The mathematical relations for calculating the stack and modeling the assembly response analytically
- The development of the solution techniques or analysis methods to carry out the tolerance analysis

In order to apply the formalization developed in chapter 3, it is necessary to consider the appropriate tools and methods for each of these steps.

The first step deals with the representation of the product geometry and behavior. The main objective of this step is to transform the geometric product deviations and its consequent behavior which may be the result of either real manufacturing process or a virtual simulation. Such a model should capture and translate the geometrical deviations in a harmonized and analytical manner that can be passed onto the next step for evaluation. The work presented in this chapter focuses on the representation in 3D. In terms of the formalization proposed earlier, this representation will populate the variables in the model.

The second step is concerned with the development of a model of the assembly response function. This function is highly dependent upon the product or assembly under consideration. The complexity of the function depends on the degree of freedom in the assembly as well as the number of components. This step generates a parametric model, which captures the interaction between the assembly parts and their features while taking into account the information provided by the geometrical transformation in the previous step. In terms of the tolerance analysis formalization, this step results in a set of constraints.

The last step in tolerance analysis takes the above two steps and applies suitable techniques to evaluate the consistency of the model in terms of deviations. Different techniques may be applied for this purpose, however, two popular methods are: Worst case tolerance analysis and Statistical tolerance analysis. This is an important step in terms of the transformation of the tolerance analysis formalization. The application of the developed approach based on the quantified expressions to tolerance analysis, that govern the respect of the assemblability of the mechanism and the respect of the functional requirements along with the interaction of the variables involved, can be realized in a computable expression. Eventually, in order to provide a complete approach, an algorithm is necessary which brings together all the methods and performs the tolerance analysis for an assembly.

The following sections address these identified areas and present an overview of the tools searched, developed and retained for the transformation of the tolerance analysis formalization. The concepts and the state of the art of the tolerance analysis process have been discussed in the section 2.3. Tolerance analysis formalization has been discussed earlier in the section 3.5.

5.2 Representation of the Geometric Variation

As discussed in the previous section, the starting point for any tolerance analysis problem is to define a method through which the geometric behavior of a part can be translated into a quantifiable format. In the CAD environment, a model is represented by ideal dimensions known as nominal dimensions. The nominal dimension is the representation of the ideal representation of the part model geometry. Due to the variations associated with manufacturing process, it is not possible to attain this nominal dimension in a repetitive manner. In reality, any specific dimension might vary within a defined range due to setup errors, tool wear and many other factors. In order to account for these factors and to ensure the desired behavior of an assembly in spite of variations, the component features are assigned a parametric zone within which the value of the feature (situation and intrinsic) lie.

The approach, used in this chapter, is a parameterization of deviations from theoretic geometry. The real geometry of parts is apprehended by a variation of the nominal geometry. The substitute surfaces model these real surfaces. This parameterization of variations is detailed in the section 3.5.1, and enables us to define a variations parametric space, in which each coordinate system axis represents a parametric variable.

The mathematical formulation of tolerance synthesis takes into account not only the influence of geometric deviations on the geometric behavior of the mechanism and on the geometric product requirements, but also the influence of the types of contacts on the geometric behavior. All these physical phenomena are modeled by convex hulls (compatibility hull, interface hull and functional hull), discussed in the theory of tolerance analysis in section 3.5.3, which are defined in the variations parametric space. A convex hull or a convex polytope (Bisztriczky *et al.*, 1994; Ziegler, 1994) may be defined as a finite set of points, as the intersection of a set of half-spaces, or as a region of n-dimensional space enclosed by hyperplanes.

With this description by convex hulls, a mathematical expression of the admissible deviations of parts is detailed in the section 3.5.3.

5.2.1 Explanation of geometrical description

For an explanation of the geometrical description in further text, an example is presented in the form of a simple assembly (figure 5.1 a). In this case, two reference datum A and B have been defined for the example (figure 5.1 b). However, the suitable choice of the selection of the specification and references rests with the designer.

In order to illustrate the application of different variations on the components, we take the component 1 from the assembly. The CAD model of the component is defined with its tolerance specification (figure 5.1 b). Due to different variables discussed earlier, at the time of fabrication, the component geometry no longer conforms to the specified dimensions or the nominal surface (figure 5.2 a) and takes the form of a non ideal surface (figure 5.2 b).

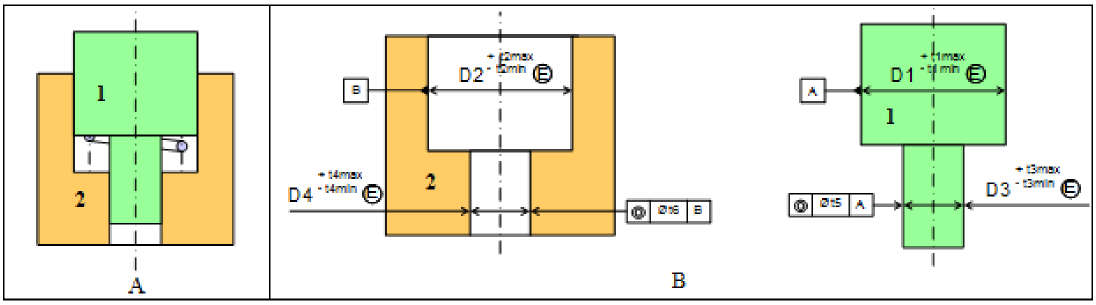

Figure 5.1: Sample assembly for tolerance analysis.

It is not possible to extract every detail of the non ideal surface, therefore it is approximated with a surface that can parameterize the deviations from the nominal surface. This surface is known as the substitute surface. It may form due to situation deviations or due to intrinsic variations. Figures 5.2 e and 5.2 f illustrate and differentiate between the concepts of intrinsic and situation deviation. A change in the diameter of the substitute surface due to the result of the imperfections on surface "a" is considered an intrinsic deviation as it changes one of the main specification of the primitive "cylinder". On the other hand, considering surface "b", we can distinguish the situation deviation taking place. The deviation may, in its own self, be decomposed into position and orientation variation. Position variation is defined as the displacement of the substitute surface representing the non ideal surface from the nominal or ideal surface. Orientation variation on the other hand is the measure of the angular deviation of the substitute surface from its nominal surface. The gaps between the components of an assembly can also be defined into the gaps according to the degree of contact and the gaps according to the degree of freedom (figure 5.2 g). This distinction of gaps and contacts has been detailed by Dantan & Ballu (2002); Dantan *et al.* (2005).

5.2.1.1 Geometric description in 1D

As the first step, the individual components are analyzed carefully and their geometry is expressed in the form of variable part features i.e. each fixed nominal

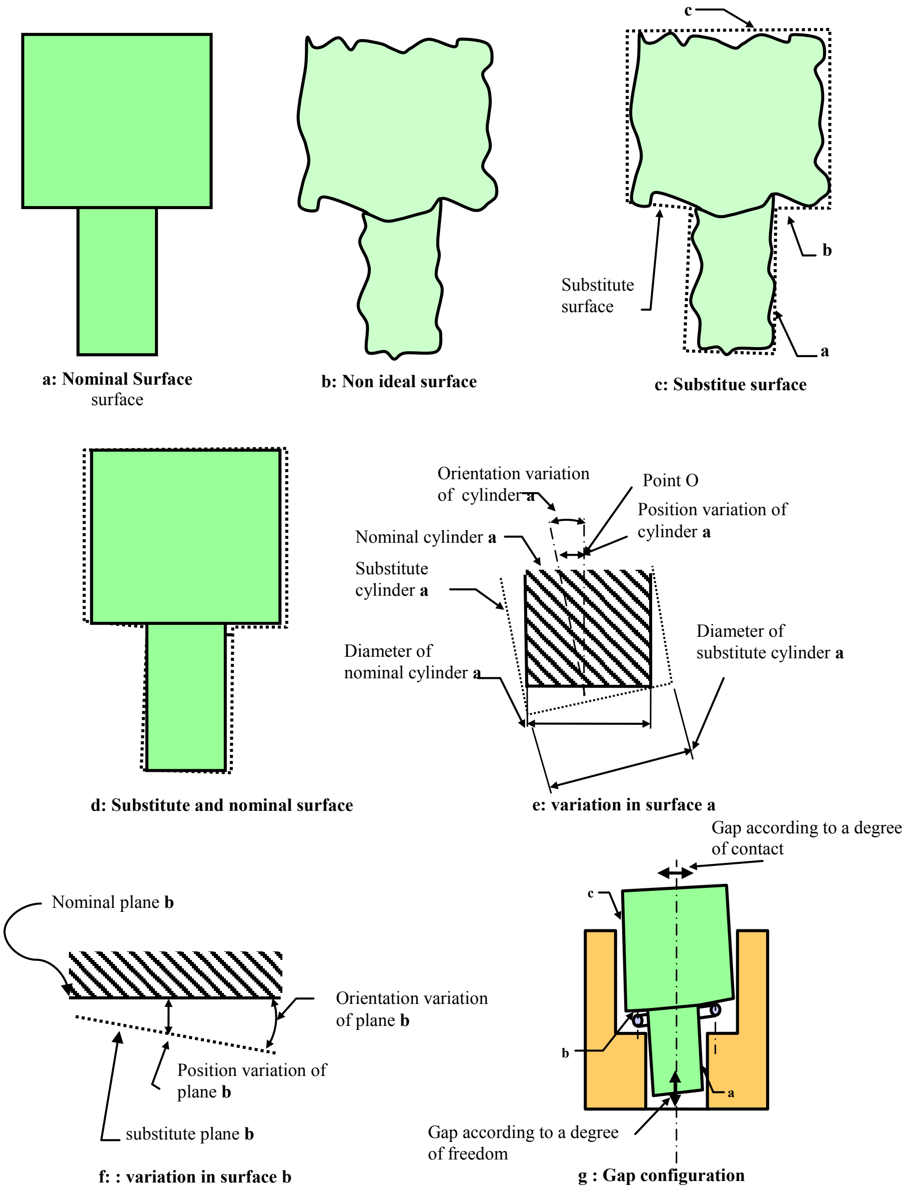

Figure 5.2: Definition of situation deviation, intrinsic deviation and gaps

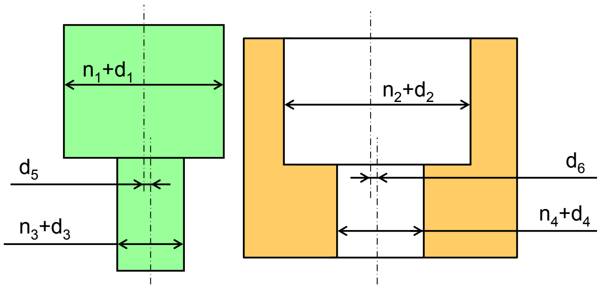

Figure 5.3: Part deviations

dimension n_i is assigned a corresponding deviation variable d_i which defines the allowable parametric variation to that nominal dimension (figure 5.3). In addition to the nominal dimensions, deviation variables are also assigned to the geometric tolerance features such as coaxiality etc.

5.2.1.2 Geometric description in 3D

The application of the formalization attempts to apply the developed work to 3D tolerance analysis. In order to do so, an appropriate model to represent the variation of the real entity from the ideal entity in 3D is to be selected. It can be described by the following manners:

- With the help of the vectors (Chase & Parkinson, 1991; Pasupathy *et al.*, 2003),
- By the torsors of the small displacements. (Ballot & Bourdet, 1997; Bourdet *et al.*, 1996; Dantan & Ballu, 2002; Teissandier *et al.*, 1999)
- By matrices (Desrochers, 1999; Gupta & Turner, 1993; Roy & Li, 1999)
- By a kinematic approach (Kyung & Sacks, 2003)
- By stream of variations (SOVA) (Huang *et al.*, 2007a,b)

The formalization developed is model independent. For the sake of illustration, in this chapter, the model chosen for the geometric deviation is the Small

Displacement torsors (SDT) (Bourdet *et al.*, 1996) (Dantan & Ballu, 2002). The small displacement torsor is used for modeling the geometric deviation of mechanical parts. The components of a small displacement torsor can also be seen as different parameters for orientation and location. A torsor d consisting of a rotational part r and a translational part t is given as:

$$d = \left\{ \begin{array}{c} r \\ t \end{array} \right\}, r = \left[\begin{array}{c} \alpha \\ \beta \\ \gamma \end{array} \right], t = \left[\begin{array}{c} u \\ v \\ w \end{array} \right] \tag{5.1}$$

In this chapter, SDT is used to model the variation and deviation. SDT therefore models the variables (section 3.5.1)by two main torsor types i.e., deviation torsors and gap torsors.

Deviation torsor: The deviation torsor can be used to model situation deviation as defined in section 3.5.1. The situation deviation defines the variation (position and orientation) between the nominal geometry of part k and a substitute surface ka . A transformation $\{d_{k/ka}\}$ is associated to the substitute surface k_a of a part.

$$\{d_{k/ka}\} = \left\{ \begin{array}{c} r_{k/ka} \\ t_{k/ka} \end{array} \right\}, \tag{5.2}$$

In general, $\{d_{k/ka}\}$ contains three translational parameters and three rotational parameters. However, as the elementary surfaces usually have some invariances, some of the $\{d_{k/ka}\}$ parameters could be reduced to 0 (Dantan & Ballu, 2002). Considering the example presented in Figure 5.1, and taking the z-axis to be in line with the symmetric axis of the assembly, the situation deviation of part “1” at surface “a” is given by the following torsor:

$$d_{1/1a} = \left\{ \begin{array}{c} r_{1/1a} \\ t_{1/1a} \end{array} \right\}, r_{1/1a} = \left[\begin{array}{c} \alpha_{1/1a} \\ \beta_{1/1a} \\ 0 \end{array} \right], t_{1/1a} = \left[\begin{array}{c} u_{1/1a} \\ v_{1/1a} \\ 0 \end{array} \right] \tag{5.3}$$

similarly, the situation deviation of the part “1” at surface “b” is given by the following torsor:

$$d_{1/1b} = \left\{ \begin{array}{c} r_{1/1b} \\ t_{1/1b} \end{array} \right\}, r_{1/1b} = \left[\begin{array}{c} \alpha_{1/1b} \\ \beta_{1/1b} \\ 0 \end{array} \right], t_{1/1b} = \left[\begin{array}{c} 0 \\ 0 \\ w_{1/1b} \end{array} \right] \tag{5.4}$$

The intrinsic deviations of the substitute surface are specific to its type. For instance, intrinsic variation of a substitute cylinder is the radius variation between the substitute cylinder and nominal cylinder.

In the same way, the gap can also be modeled by torsors (Bourdet *et al.*, 1996; Dantan & Ballu, 2002). Two types of gaps (or displacements) are distinguished: The displacements according to the degree of freedom and the displacements according to the degree of contact. The displacements according to the degree of contact are due to an eventual void between the surfaces.

Gap Torsor: While describing a gap torsor, gaps according to the degree of contacts are denoted by small letters in the mathematical expressions and gaps according to the degree of freedom are expressed by capital letter or marked infinity. Considering the example presented in the figure 5.1, and taking the z-axis to be in line with the symmetric axis of the assembly, the gap torsor between the substitution surfaces of the part "1" and part "2" at surface "a" is given by the following torsor:

$$d_{1a/2a} = \left\{ \begin{array}{c} r_{1a/2a} \\ t_{1a/2a} \end{array} \right\}, r_{1a/2a} = \begin{bmatrix} \alpha_{1a/2a} \\ \beta_{1a/2a} \\ \Gamma_{1a/2a} \end{bmatrix}, t_{1/1a} = \begin{bmatrix} u_{1a/2a} \\ v_{1a/2a} \\ W_{1a/2a} \end{bmatrix} \tag{5.5}$$

5.3 Constraint Expression Via Hulls

Having chosen the model for the translation of the product geometric information in terms of parametric information, the set of variables for the tolerance analysis problem is populated. The next step deals with the generation of a product analytical model. It is generated by expressing the topological loops of the assembly. For this purpose, the deviation and gap torsors are defined for all the surfaces through which the assembly topological loops pass. Using the definitions of hulls as described in section 3.5.3, the constraints that model the assembly can now be formulated.

The constraints are divided into three categories: The group of constraints from the compatibility hull; the group of constraints from the interface hull;

and the group of constraints arising from the functional hull. The set of these constraints forms the global constraint model for the tolerance analysis problem.

A mechanism can be classified into any of two main categories according to their degrees of freedom:

- Iso-constrained mechanisms
- Over-constrained mechanisms

Given their impact on the constraint model formulation for the problem of tolerance analysis, a brief discussion of these two types is provided as discussed by Ballu *et al.* (2008).

Iso-constrained mechanism : Iso-constrained mechanisms are those mechanisms in which geometric deviations do not lead to assembly problems; the deviations are independent and the degrees of freedom catch the deviations. When considering small deviations, functional deviations may be expressed by linear functions of the deviations. These functional deviations may be linked to the geometric ones by any of the methods described in section 5.2.1.2. Commercial software, with statistical approach, introduces gap deviations as random variables with a mean value equal to zero. The distributions (standard deviations for Gaussian distributions) of the gaps are defined from the maximum material dimensions.

In fact, worst gaps are dependent of the dimension deviations. As the deviations are very small, a linear relation links gaps and dimension deviations. The assembly response function in the case of an iso-constrained mechanism can be expressed in a explicit equation.

Over-constrained Mechanism : Considering over-constrained mechanisms is much more complex. Assembly problems occur and the expression of the functional deviations is no more linear. Depending on the value of the manufacturing deviations:

- The assembly is feasible or not

- The worst configuration of contacts is not unique for a given functional deviation

For each over-constrained loop, events on the deviations have to be determined:

- Events ensuring assembly
- Events corresponding to the different worst configurations of contacts

As there are different configurations, the expression of the functional deviation cannot be linear. It is linear, only for a particular configuration, it means that for each event (i.e. each configuration), a specific model has to be defined.

In the case of most iso-constrained mechanisms, the assembly response function is explicit in terms of the individual tolerances. But, in the case of over-constrained mechanisms, the assembly response is more complex and maybe expressed in implicit form as shown in figure 5.4.

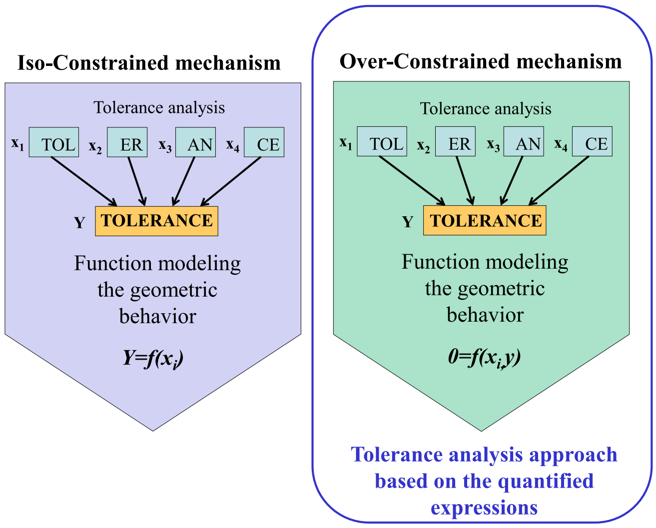

Figure 5.4: Iso-constrained & over-constrained mechanisms

With respect to the statistical methods commonly used, we can notice that these methods are generally limited to iso-constrained mechanisms. Moreover, gaps within the mechanisms are not taken into account or are considered as random variables. Hence, we would like to point out the fact that a good modeling of over-constrained mechanisms and gaps greatly improves the result on the tolerances.

In order to apply the formalization of tolerance analysis, especially to see its performance in the implicit assembly response functions, the example selected is that of an over-constrained mechanism. The example is presented in section 5.5.

5.4 Development Of Analysis Methods

The goal of an analysis method is to provide the techniques to estimate the affect of the allotted individual tolerances on the assembly response function. A concise state of the art on tolerance analysis has been discussed in section 2.3. The general objective of the techniques is to utilize optimization, statistical, and probabilistic approaches to evaluate the model of the assembly response in the presence of deviations to ascertain if it remains within the acceptable limits or not.

According to the analysis objectives, two techniques have generally been used: (1) worst case tolerance analysis (WCTA) and (2) statistical tolerancing analysis (STA). In worst case tolerancing (i.e., 100% acceptance rate), the aim is to find the worst possible part, in terms of functional characteristics. If the worst part in a series of manufactured parts complies with the functional characteristics, it is logical to conclude that all of the parts manufactured will comply. Worst case methods give results that are overly pessimistic. In statistical tolerance analysis, it is assumed that variations of individual contributing features are independent and are given some probability distribution. This in turn allows computing some cumulative probability that the product will meet its design tolerance requirements at the end of the manufacturing process. Computations are most often performed using Monte-Carlo simulations (Kamali Nejad, 2009).

In order to transform the formalization of the tolerance analysis, we will discuss these two techniques separately in the following sections and utilize them to carry out the tolerance analysis of the assembly example.

Dantan *et al.* (2005) present a mathematical formulation of tolerance synthesis to simulate the influences of geometric deviations on the geometric behavior of the mechanism using the quantifier notion (existential and universal quantifiers) and model the phenomena by convex hulls (compatibility hull, interface hull and functional hull) which are defined in a parametric space. With this description using convex hulls, a mathematical expression of the admissible deviations of parts integrates the quantifier notion. With this approach, some rules are formalized to determine the modifier (maximum material condition or minimum material condition) function from the type of quantifier. Using the virtual gauge, and the quantifier notion, worst-case tolerance analysis can be performed.

Kamali Nejad *et al.* (2009a,b) develop an integrated numerical technique for the tolerance analysis problem of a virtually manufactured part with a 3D approach. This approach is based on the concept of the SDT (Bourdet *et al.*, 1996), Model of Manufactured Part MMP (Vignat *et al.*, 2010) and virtual gauge (Dantan & Ballu, 2002) to perform worst case and statistical tolerance analysis of a manufactured part for the respect of the functional tolerance specification stipulated during the manufacturing stage. SDT is used to model the deviations in 3D space, and MMP is used to model the deviations arising in a part due to different operations in the machining phases such as:

- Errors induced due to datum errors
- Errors due to fixture components
- Errors due to machining deviations depending upon the machining, fixing and setting up process

This results into successive intermediate virtual parts (MWP), which at the end of the modeling process creates a virtual manufactured part (MMP). The MMP contains the information about the deviations generated during the complete operations.

Once a virtual manufactured part is produced, the different parameters involved in the MMP are assigned ranges. Three different strategies regarding the parameter variation range are adopted: dependent parameters, independent parameters, and variation zone. The range for the dependent parameters is taken from the data obtained from the production and qualification of a sufficient quantity of parts resulting into dependent expressions between the parameters which can then be used as the range for the components of the MMP. In the case of independent parameters, it can be drawn from a suitable probability distribution dataset to simulate the components of the MMP. A variation zone can also be used to set up the limits of the components of the MMP by which, the variation range is bounded by the limits of the 3D variation zone of the feature that it represents.

With the allotted parameter ranges, Kamali Nejad (2009) provides the methods for tolerance analysis with the MMP and the use of virtual gauges (Dantan & Ballu, 2002). Two approaches have been provided: Worst case tolerance analysis with the help of optimization algorithms, and stochastic tolerance analysis with the help of Monte Carlo simulation. The worst case based tolerance approach is based on a multilayered problem. The inner nested layer finds the most constricting gap according to the virtual gauge for the part to be analyzed and maximizes its values to establish the existence of a configuration that respects the functional tolerance. The outer global layer finds the worst possible part in terms of tolerance that a multistage machining process might produce. If such a part complies with the functional tolerance, all other manufactured part would also comply (Kamali Nejad, 2009). The interval based worst case approach makes use of the interval arithmetic to assign values in terms of intervals to the parameters using the developed approach and solves the problem in terms of intervals with the help of a combined approach using MMP and the Jacobian-Torsor model. Lastly, these methods are compared with a stochastic approach based on the Monte carlo simulation to simulate the virtual parts. The optimization techniques, used in the research work, are genetic algorithm and sequential quadratic programming from built in Matlab® routines for finding the optimized solutions.

The work presented by Kamali Nejad (2009) brings into light, the possible application of torsor based geometric model for implication in the product life cycle from manufacturing process simulation to part qualification via tolerance analysis. The work concerned with the tolerance analysis of a virtually machined part, lies in a similar domain to the more generic tolerance analysis formalization, presented in this research work. This research work aims to formalize an expression for the gap based tolerance analysis for the assemblies consisting of multiple parts for the respect of specified functional conditions, whereas the work presented by Kamali Nejad *et al.* (2009b) presents the tolerance analysis for a single virtually machined part for the respect of the functional tolerances. This approach involves optimization for worst case tolerance analysis and Monte-Carlo simulation. These formalizations, expressed in terms of a double layered optimization system, can be expressed within the context of the tolerance analysis formalization developed in the earlier chapters using the quantified expressions provided by the first order logic.

5.4.1 Approach for the worst case tolerance analysis

The scope of the tolerance analysis formalization in this thesis is restricted to:

- Expressing the tolerance analysis problem in terms of logical and mathematical expression as proposed in chapter 3

- Selection of the appropriate tools for modeling the geometric behavior

- Development and application of the techniques, for transformation of the quantifier notion in an algorithmic form, by using optimization and statistical approaches, allowing application of the problem to overconstrained assemblies.

The resolution of the quantified expressions, resulting from the formalization, falls under the general resolution problem of the $QCSP$. The following text looks at the problem of the $QCSP$ and its context in the tolerance analysis problem.

5.4.1.1 QCSP

The concept of the $QCSP$ has been presented and discussed in section 3.2.3. For tolerancing application, the $QCSP$ is used to check the model. The algorithm takes a QCSP with a set X of existentially or universally quantified variables in a given order, and computes the arc consistent sub-domains in case the problem is arc consistent or returns FALSE in case the problem is not arc consistent.

5.4.1.2 Formulation of tolerance analysis for QCSP solver

In this section, we propose the method which enables us to transform the logical conditions for the worst case tolerance analysis problems as laid down in the sections 3.5.4.1 and 3.5.4.2 using the QCSP principles described earlier.

$$\begin{aligned}
&\forall x_1, \forall x_2, \forall x_3, ..., \forall x_n;\\
&\exists x_{n+1}, ..., \exists x_m;\\
&c_1, ..., c_p
\end{aligned} \tag{5.6}$$

Where:

- $x_1, x_2 ....., x_n$ are the variables representing each part deviation (S, I)
- $x_{n+1}, ..., x_m$ are the variables representing each gap between parts G
- The mathematical representation of geometric specifications is a set of intervals which limit each part deviation like vectorial tolerancing: $x_i \in D(x_i)$ with x_i a part deviation and $D(x_i)$ its tolerance interval, $D(x_1) \times .... \times D(x_n) = H_{specification}$
- The mathematical representation of interface constraints is a set of inequations which limit each gap: $x_j \in D(x_j)$ with x_j a gap and $D(x_j) : x_j \geq 0$, $D(x_{n+1}) \times ... \times D(x_m) = H_{interface}$
- The mathematical representation of the compatibility equations is a set of constraints: $c_1, ..., c_p$

Therefore, the mathematical expression of tolerance analysis for assembly requirement is: "For all acceptable deviations (deviations which are inside tolerances), there exists a gap configuration such that the assembly requirement (interface constraints) and the compatibility equations are verified".

The expressive power of QCSP integrates the notion of quantifier in terms of mathematical expression for tolerance and allows us to model the mathematical formulation of tolerance analysis for assembly requirement with a vectorial tolerancing model.

However, there remains limitation, in the QCSP expression power, to be addressed in terms of implicit universal quantification of the variables which do not have the direct property of projectability. The lack of ability of universal quantification of variables which do not possess the property of projectabililty means that other means need to be adopted to circumvent this limitation. An instance of this problem occurs during the universal quantification of the gaps they depend upon (S and I).

5.4.2 Statistical tolerance analysis based on formalization and Monte Carlo simulation

In order to ensure the robustness of design, it is necessary to simulate and study the effect of variations on assembly due to manufacturing variations. Variations may take place in any random pattern concurrently in the concerned dimension and may affect the assemblability and function of the assembly. In the following section; the example of tolerance analysis, developed through the transformation of sections 3.5.4.1 and 3.5.4.2, with the help of optimization techniques will be modified and integrated with the Monte Carlo simulation to obtain an algorithm. This algorithm performs the tolerance analysis of a mechanical assembly from a sample population of components generated by Monte Carlo simulation.

The tolerance analysis method adopted in this report is based on the application of Monte Carlo simulation to the solution of a CSP in order to calculate the probability of assembly and functioning of a given assembly. In order to develop the algorithm, it is first necessary to gain a general idea of the Method of Monte Carlo simulation. Afterwards, a flow chart involving the application of the Monte Carlo Simulation for 3D tolerance analysis will be discussed.

5.4.2.1 Monte Carlo simulation

Monte Carlo simulation is a popular method for performing statistical variation analysis. Fig. 5.5 graphically shows how the process works.

The output distribution is a function of the distributions of the input variables and the assembly function. Thousands of samples of the input variables are combined to get a reliable measure of the output distribution (Nigam & Turner,

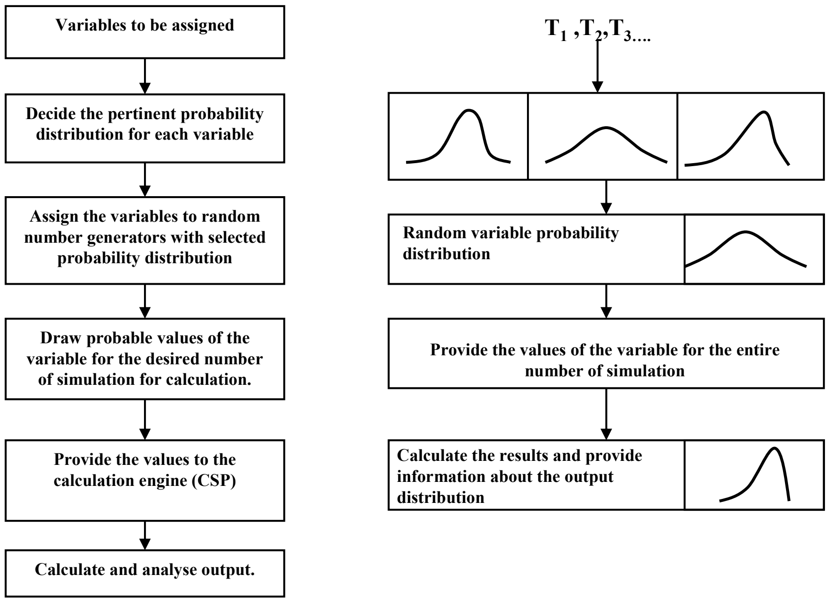

Figure 5.5: Monte carlo simulation

1995). In this study, Monte Carlo simulation is used to simulate the geometric variations of the mechanism. It can be used successfully in the situation where the assembly function is not available explicitly. However, the sample size should be sufficiently large in order to ensure a stable and consistent result.

For this specific case, the objective of the simulation is to ascertain the probability of the assembly of the mechanism and the probability of the respect of the functional conditions for a given tolerance specification for a mechanism.

Let P_A be the probability of the assembly for a given tolerance specification. This specifies the respect of the assembly requirements. Let P_{FR} be the probability of respect of the functional requirements of an assembly. Let AR be the event that the assembly requirements for a given assembly are respected. Let FR be the event that the functional requirements of an assembly are fulfilled. Then the probability of the assembly and the probability of the respect of the functional requirements can be expressed as:

$$\begin{aligned} P_A &= Prob(AR) \\ P_{FR} &= Prob(AR \cap FR) \end{aligned}$$

The Monte Carlo simulation along with the developed formalization and the transformation enables us to calculate these probabilities. The failure of an assembly and the failure with respect to the functional requirement in parts per million (*ppm*)s can be used as a measure at the end of the simulation to represent the parts in million, that fail to assemble or that respect the functional requirements.

5.4.2.2 Transformation for statistical tolerance analysis

In this section, we propose the method to transform the logical conditions specified for the statistical tolerance analysis problems (sections 3.5.4.1 and 3.5.4.2) using the QCSP principles described earlier.

Assemblability of the mechanism: The condition of the assemblability of the mechanism in section 3.5.4.1 describes the first essential condition to be fulfilled by the assembly in order to successfully validate the functional conditions for a robust design. In order to apply the above condition into a computable form using Monte Carlo simulation, equation 3.46 needs to be transformed into a mathematical expression that translates into the following:

"For each sample (Monte Carlo simulation) of acceptable deviations (deviations which are inside tolerance limit), there exists a gap configuration such as the assembly requirement (interface constraints) and the compatibility equations are verified"

Respect of functional condition (robust solution): The condition for the existence of a robust solution is given in tolerance theory (section 3.5.4.2). This condition describes the second subsequent condition to be fulfilled by the assembly configurations validated by the condition of the assemblability. The evaluation of the second condition confirms, that, for a given configuration, the assembly will assemble and conform to the functional conditions. In order to apply the above condition to a computable form using Monte Carlo simulation, equation 3.47 is to be transformed into a mathematical expression that translates into the following:

"For each sample (Monte Carlo simulation) of acceptable deviations (deviations which are inside tolerances), the worst case of functional characteristics must be respected for the functional requirements such as the interface constraints and the compatibility equations to be verified"

A diagram describing the developed algorithm for tolerance analysis is shown in figure 5.6. The main steps in the algorithm are explained via respective pseudocodes in the following text.

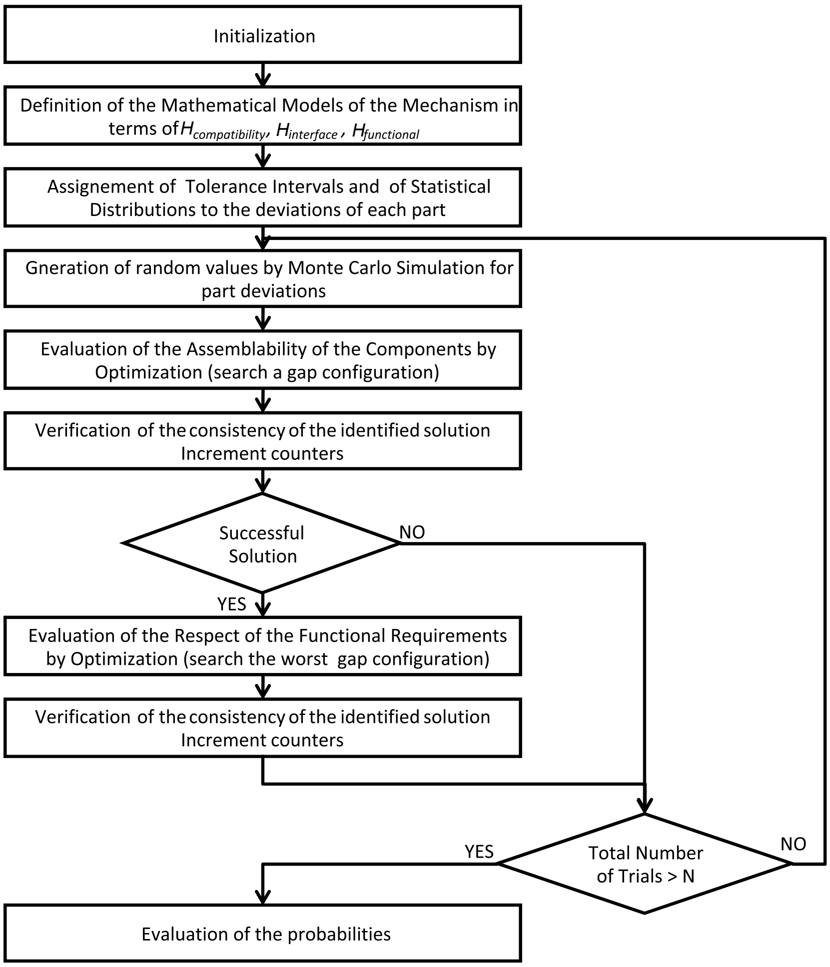

Figure 5.6: Algorithm for statistical tolerance analysis for the formalization

The main principle, behind the algorithm, is to simulate and evaluate the influence of the manufacturing deviations on the assembly and then evaluate the assembly requirement and the functional requirement expressed through the formalization transformation. In order to achieve this, Monte Carlo simulation and quantifier notion are used to simulate the deviations and to quantify the variables in the geometric behavior model respectively. This process is repeated recursively for a large sample of deviations to establish a resultant distribution of assembly probability. This probability is a measure of the assembly success or failure for a given assembly consisting of sub components.

The parametric model, of the assembly is defined by the set of variables from the standard notation described in section 3.5.1 and the set of constraints consisting of these variables obtained from compatibility, interference and functional hulls as explained in sections 3.5.3 and 5.3. These constraints take the general form as expressed in equations 3.38-3.44.

The part deviations (S, I) are then evaluated recursively within the algorithm. Monte Carlo simulation is used to generate random variables simulating the part deviations with all the generated deviations being within the $H_{specification}$ which is the Hull of specifications as defined by the designer.

A sample of part deviations generated by Monte Carlo simulation is noted:

$$S' = \{s'_1, s'_2, ..., s'_n\} \tag{5.7}$$

$$I' = \{i'_1, i'_2, ..., i'_o\} \tag{5.8}$$

For any given instance of iteration, the part deviations generated by Monte Carlo simulation should satisfy the set of constraints:

$$(S', I') \in D_{specification} \tag{5.9}$$

The pseudo code for Monte Carlo Simulation is shown in algorithm 5.1.

Algorithm 5.1 Monte Carlo Simulation Algorithm

Require: $S, I, H_{Specification}, No\_of\_simulations$
Ensure: $S', I' \in H_{specification}$

```
Assign probability distribution p(s_j), p(i_k) to all s ∈ S andi ∈ I
for l = 1 to No_of_simulations do
    for m = 1 to n do {n = Size of S}
        Generate s'_m from p(s_m)
    end for
    for c = 1 to o do {o = Size of I}
        Generate i'_c from p(i_c)
    end for
end for
```

Assemblability condition transformation: To evaluate the assemblability of each sample (instance of part deviations), we verify if there exists an admissible gap configuration of the mechanism, such that the assembly requirement (interface constraints) and the compatibility equations are verified. This check is performed with the help of equation 3.46. In the algorithm, for an instance of iteration, its mathematical form becomes:

$$\begin{aligned} &\exists g \in: \{g \in G : (S, G, I) \in H_{Compatibility} \cap H_{Interface}\} \\ &(S', I', G, Fc) \in H_{Compatibility} \cap H_{Interface} \end{aligned} \tag{5.10}$$

Depending on this decision process, it may be desirable:

- To determine whether a solution exists, (verify the consistency of the Constraint Satisfaction Problem)
- To find at least one solution; to compute the space of all solutions of the Constraint Satisfaction Problem
- To find an optimal solution relative to a given cost function which respects all constraints (C)

In our case, the goal is not to find a particular solution but to validate the consistency of the variables over the set of constraints. This can be achieved by using the in-built Mathematica® function "Exists". This function is a black box function usable for basic problems. However, with the increase in the complexity of the problem, this option no longer remains feasible. In order to find an application to more complex problems, we have transformed the expression into an optimization problem.

The optimization problem is formulated to check the existence of the gaps in the assembly for the given configuration and samples generated by the Monte Carlo simulation. This is done through optimization, by minimizing all Gaps, with respect to the compatibility and interface constraints. To do that, a numerical algorithm is used to find the global minimum of sum of the gaps subject to the constraints. By this method, a minimum solution (worst case) is obtained for the gap configuration such that the assembly requirement is verified. The aim is to find if there are values of gaps such that all the constraints are satisfied with gaps having values of more than 0 (there is no interference). The result of this step effectively establishes whether the individual parts with the simulated deviations are able to assemble.

The expression 3.46 and 5.10 can be written mathematically as:

$$\begin{aligned} &(S', I') \in D_{specification} \\ &(\min(G : S', I' \in H_{Compatibilty} \cap H_{Interface})) \end{aligned} \tag{5.11}$$

This optimization problem includes non-linear constraints (the constraints are a logical combination of equalities, inequalities, and domain specifications. Equalities and inequalities may be nonlinear. An example of non linear constraint is detailed in the following section), therefore we use Numerical Nonlinear Global Optimization techniques. Numerical algorithms for constrained nonlinear optimization can be broadly categorized into gradient-based methods and direct search methods. Gradient-based methods use first derivatives (gradients) or second derivatives (Hessians). Direct search methods do not use derivative information. They use functions that implement several algorithms for finding constrained global optima. The methods are flexible enough to cope with functions that are not differentiable or continuous and are not easily trapped by local optima. The implemented algorithms include a tolerance for accepting constraint violations. Therefore, we add a step to check the consistency of the identified solution.

Algorithm 5.2 presents the pseudo-code of this transformation.

Algorithm 5.2 Assemblablity of Mechanism Algorithm

Require: $H_{Compatibility}, H_{Interface}, S, I, Monte\_Carlo\_Algo$

Ensure: $G : S', I' \in H_{Compatibilty} \cap H_{Interface}$

1: Generate samples $S', I' \in H_{Compatibilty}$ through Monte Carlo Simulation
2: $assemb\_test = \min(G : S', I' \in H_{Compatibilty} \cap H_{Interface})$ {Minimization of Gaps}
3: **if** $assemb\_test$ =**true then**
4: **return** Assembly Possible
5: **else**
6: **return** Assembly Impossible
7: **end if**

Moreover, the check of the solution existence can be made by hull-consistency algorithm. The key idea is to generalize constraint-consistency (Bordeaux & Monfroy, 2006) criterion to a higher level where the set of all constraints is seen as a single global constraint. Hence, it must guarantee arc consistency at the bounds of the variable domains for this single global constraint.

Functional condition verification: To evaluate the respect of the functional requirements for each sample (instance of part deviations) that assembles, we verify if for all admissible gap configuration of the mechanism, there exists functional characteristics such that the functional requirements are verified. This check is performed with the help of the equation 3.47. In the algorithm, for an instance of iteration, its mathematical form becomes:

$$\begin{aligned} &\forall g \in \{g \in G : (S, G, I) \in H_{Compatibility} \cap H_{Interface}\} \\ &\exists Fc \in H_{functional} \\ &: (S', I', G, Fc) \in H_{Compatibility} \cap H_{Interface} \cap H_{functional} \end{aligned} \tag{5.12}$$

Depending on this decision process it may be desirable either to determine the space of all solutions of the Constraint Satisfaction Problem, or to find an optimal solution relative to a given cost function which respects all constraints (C). For examples, pertaining to elementary level, the check of the solution is made possible by using the built-in Mathematica® function "ForAll". However, with the increase in complexity of the assembly, the usage of this function becomes time consuming. Two alternative approaches to translate relations 3.47 and 5.12 have been formalized which are:

1. Optimization of worst-case values

2. Interval propagation with the help of interval arithmetic to reduce the domain of the variables (functional characteristics)

We use the optimization approach to find the worst case values of the functional characteristics which is given by the following expression:

$$\begin{aligned}
&(S', I') \in D_{specification} \\
&Fc_{\min} \wedge Fc_{\max} \in H_{functional} : \\
&Fc_{\max} = \max(Fc(G, S', I')), Fc_{\min} = \min(Fc(G, S', I')) \\
&S.T. \\
&C_{comp} = 0 \\
&C_{int} \geq 0 \\
&Fc_{\min} \leq C_{fonc} \geq Fc_{\max}
\end{aligned} \tag{5.13}$$

The transformation effectively describes the consistency check required for the evaluation of the respect of the functional conditions as described by the logical expression for the robust solution in the tolerance analysis theory. The boundary evaluation of the functional conditions, with the previous optimization for the minimization of the gaps, makes sure that the components would assemble without interference while maintaining the respect of the functional condition, thereby validating the specifications for the given configuration of the assembly.

For a given configuration of allowable deviations and nominal dimensions for the assembly, this process is repeated iteratively with the Monte Carlo simulation for the number of runs necessary for the assembly and desired confidence interval to obtain the required probability of assembly and respect of the functional conditions.

Algorithm 5.3 presents the pseudo-code of this transformation.

Algorithm 5.3 Functional Condition Verification Algorithm

Require: $H_{Compatibility}, H_{Interface}, S, I, Monte\_Carlo\_Algo, assemb\_test$

Ensure: $G, Fc : S', I' \in H_{Compatibilty} \cap H_{Interface} \cap H_{Functional}$

1: $Fc\_test_{\min} = \min(G, Fc : S', I' \in H_{Compatibilty} \cap H_{Interface} \cap H_{Functional})$
2: $Fc\_test_{\max} = \max(G, Fc : S', I' \in H_{Compatibilty} \cap H_{Interface} \cap H_{Functional})$
3: **if** $Fc\_test_{\min} \wedge Fc\_test_{\max}$ =**true then**
4: **return** Respect of functional conditions
5: **else**
6: **return** Functional conditions violated
7: **end if**

The above two transformations run in conjunction with the Monte Carlo simulation, inside an algorithm for the given number of simulations, to provide the basis to assess the respect of the assemblability conditions and the functional conditions as proposed by the formalization.

5.4.2.3 Results

The algorithm has been programmed to return the results at the end of the simulation in the form of probability of assembly of the components and the probability of the respect of the functional conditions. The algorithm can also return the results in terms of histogram and percentile distribution.

5.5 Application

The approach, developed for tolerance analysis for 3D mechanical assemblies, has been applied and validated over different models with and without GD & T specifications. For the sake of brevity and clarity of application, a simple over-constrained mechanism (as shown in figures 5.7 and 5.9)is taken as an example.

Figure 5.9 illustrates the different views of the case study mechanism. Two main parts are assembled by three guides. The contact between the guides and one of the parts (Part 2) is fixed, and the contact between the guides and the other part (Part1) is floating. The functional characteristic (Fc) is coaxiality

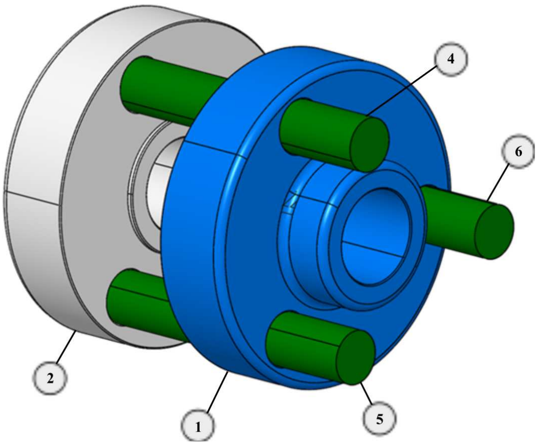

Figure 5.7: Over Constrained Translation Assembly

between the center holes of the two parts. The specifications of the case study are provided in figure 5.8.

This example includes 32 part deviation variables, 24 gap variables and 6 functional characteristic variables. The model is based on 3 assembly requirement topological loops therefore 18 assembly requirement compatibility equations (linear equations) and 3 functional requirement topological loops therefore 18 functional requirement compatibility equations (linear equations). There are 6 non interference constraints (6 non linear inequations, 6 fixed joint constraints (12 linear equations), and 2 functional requirement constraints (2 non linear inequations)).

The first step is to define the assembly behavior model which can then be used with the help of the hulls (defined in the formalization) to develop the constraints related to the assembly. In order to do so, a primary links diagram is made to identify the contact relationships in the assembly. Figure 5.10 shows the preliminary links diagram presenting the contacts.

In figure 5.10, the Blue ellipse depicts component 1, the gray ellipse depicts component 2, whereas, the green circles (4,5,6) depict the cylindrical guides. Cyan circles (a,b,c,d) in the figure show the contact surfaces between the assembly components. The global origin is the center of the inner side of the bore on part 2.

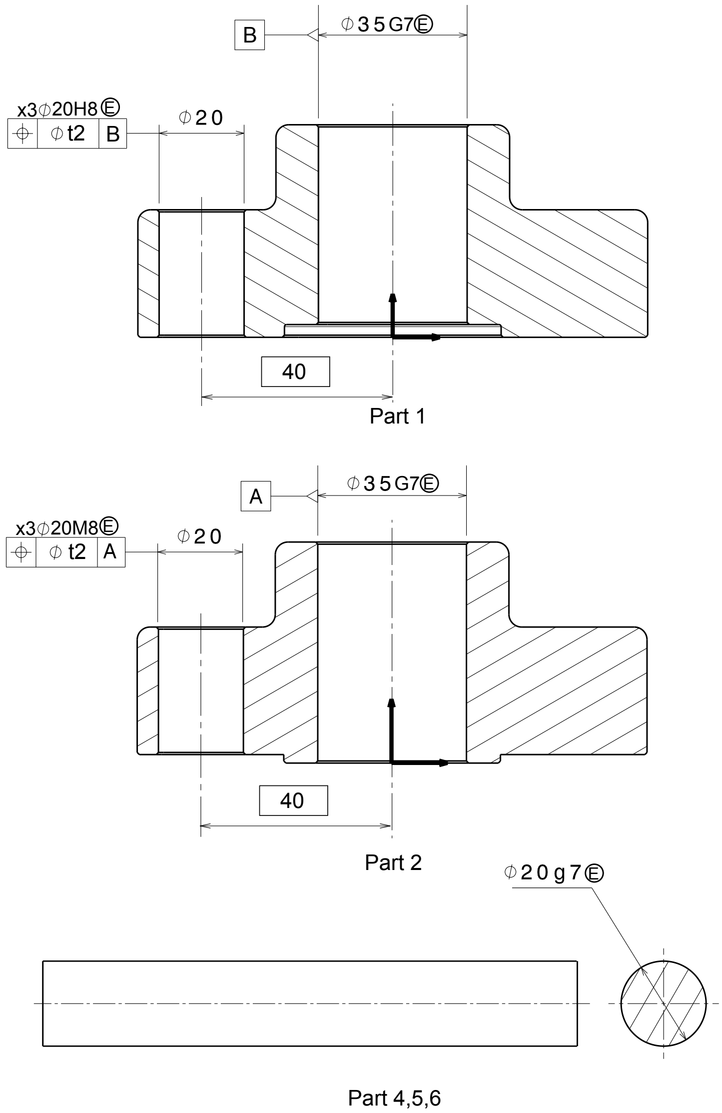

Figure 5.8: Over Constrained Translation Assembly-Specifications

The next step is to develop the topological loops passing through these links using the hulls defined in the theory. These loops are expressed by the sum of deviation and gap torsors contained within a given loop.

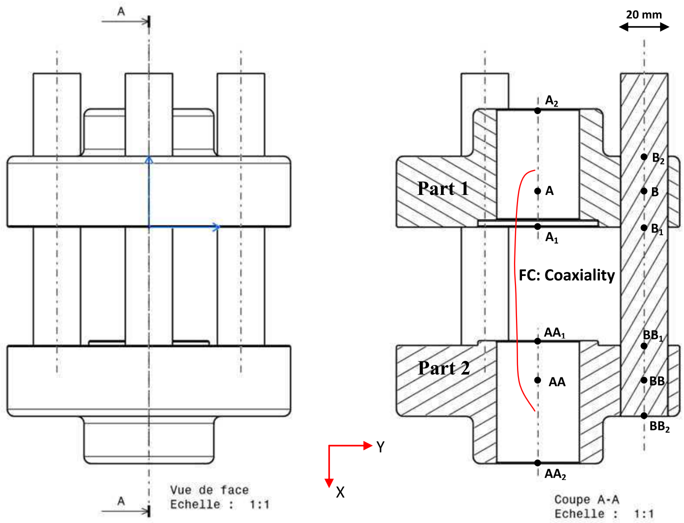

Figure 5.9: Over Constrained Translation Assembly-Projections

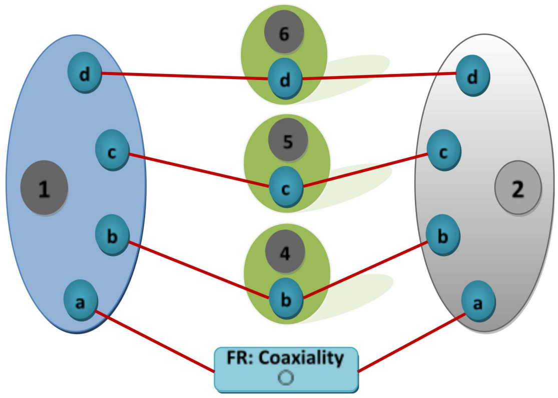

Figure 5.10: Links diagram for over constrained Translation Assembly example

5.5.1 Geometric description

The algorithm developed is capable of performing 1D, 2D and 3D tolerance analysis of any given mechanical assembly. In this example, the 3D geometric behavior model (developed with help of SDT as discussed in previous sections) is used as a mathematical model for the Algorithm.

The SDT $D_{k/ka,M}$ defines the part deviation (position and orientation) between the substitute surface ka and the nominal geometry of part k, expressed

at point M. The components of a SDT can also be seen as different parameters for orientation and position:
$D_{k/ka,M} = \{a_{Dk/ka}, b_{Dk/ka}, g_{Dk/ka}, u_{Dk/ka,M}, v_{Dk/ka,M}, w_{Dk/ka,M}\}$. $a_{Dk/ka}$, $b_{Dk/ka}$, $g_{Dk/ka}$ are the rotation deviation parameters and $u_{Dk/ka,M}, v_{Dk/ka,M}, w_{Dk/ka,M}$ are the position deviation parameters expressed at the point M.

In the same way, the gap can also be modeled by SDT. The SDT "$G_{ka/ib}$", M defines the gap (position and orientation) between the substitute surface ka of part k and the substitute surface ib of part i.

As mentioned in section 3.5.1, the vector Sd is defined by all situation deviation parameters of all surfaces. For the case study, the vector Sd is given as:

$$\begin{aligned} Sd = \{& bd11a, gd11a, vd11aA, wd11aA, bd11b, gd11b, vd11bB, wd11bB, \\ & bd11c, gd11c, vd11cC, wd11cC, bd11d, gd11d, vd11dD, wd11dD, bd22a, \\ & gd22a, vd22aAA, wd22aAA, bd22b, gd22b, vd22bBB, wd22bBB, bd22c, \\ & gd22c, vd22cCC, wd22cCC, bd22d, gd22d, vd22dDD, wd22dDD\} \end{aligned} \quad (5.14)$$

The vector I is defined as all intrinsic deviation parameters of all surfaces. For the case study, the vector I includes the diameter of each cylinder:

$$I = \{d1b, d1c, d1d, d4, d5, d6\} \quad (5.15)$$

In the same way, the vectors G and Fc are defined:

$$\begin{aligned} G = \{& ag2b4b, ag1b4b, bg1b4b, gg1b4b, ug2b4bBB, ug1b4bB, \\ & g1b4bB, wg1b4bB, ag2c5c, ag1c5c, bg1c5c, gg1c5c, ug2c5cCC, \\ & ug1c5cC, vg1c5cC, wg1c5cC, ag2d6d, ag1d6d, bg1d6d, \\ & gg1d6d, ug2d6dDD, ug1d6dD, vg1d6dD, wg1d6dD\} \end{aligned} \quad (5.16)$$

$$\begin{aligned} Fc = \{& afc1a2a, bfc1a2a, gfc1a2a, ufc1a2aA, \\ & vfc1a2aA, wfc1a2aA\} \end{aligned} \quad (5.17)$$

The constraint model is the set of equalities and inequalities that results from the expressions of different hulls (as defined in section 3.5.3). The constraint model for the example in terms of compatibility hull, interface hull and functional hull is discussed in the next section.

5.5.2 Geometric behavior and constraint model

The geometric behavior of the mechanism is expressed by the composition relations of small displacements in the various loops of the assembly graph (Ballot & Bourdet, 1997). The sum of situation deviations and gaps along a loop of an assembly graph (figure 5.10) must be equal to zero:

$$\begin{aligned}
AR1 \quad &: \quad \left\{D_{1/1d}\right\} + \left\{G_{1d/6d}\right\} + \left\{G_{6d/2d}\right\} + \left\{D_{2d/2}\right\} + \left\{D_{2/2c}\right\} \\
&\quad + \left\{G_{2c/5c}\right\} + \left\{G_{5c/1c}\right\} + \left\{D_{1c/1}\right\} = 0 \qquad (5.18) \\
AR2 \quad &: \quad \left\{D_{1/1d}\right\} + \left\{G_{1d/6d}\right\} + \left\{G_{6d/2d}\right\} + \left\{D_{2d/2}\right\} + \left\{D_{2/2b}\right\} \\
&\quad + \left\{G_{2b/4b}\right\} + \left\{G_{4b/1b}\right\} + \left\{D_{1b/1}\right\} = 0 \qquad (5.19) \\
AR3 \quad &: \quad \left\{D_{1/1c}\right\} + \left\{G_{1c/5c}\right\} + \left\{G_{5c/2c}\right\} + \left\{D_{2c/2}\right\} + \left\{D_{2/2b}\right\} \\
&\quad + \left\{G_{2b/4b}\right\} + \left\{G_{4b/1b}\right\} + \left\{D_{1b/1}\right\} = 0 \qquad (5.20) \\
FR1 \quad &: \quad \left\{D_{1/1b}\right\} + \left\{G_{1b/4b}\right\} + \left\{G_{4b/2b}\right\} + \left\{D_{2b/2}\right\} + \left\{D_{2/2a}\right\} \\
&\quad + \left\{G_{2a/fc}\right\} + fc + \left\{G_{fc/1a}\right\} + \left\{D_{1a/1}\right\} = 0 \qquad (5.21) \\
FR2 \quad &: \quad \left\{D_{1/1c}\right\} + \left\{G_{1c/5c}\right\} + \left\{G_{5c/2c}\right\} + \left\{D_{2c/2}\right\} + \left\{D_{2/2a}\right\} \\
&\quad + \left\{G_{2a/fc}\right\} + fc + \left\{G_{fc/1a}\right\} + \left\{D_{1a/1}\right\} = 0 \qquad (5.22) \\
FR3 \quad &: \quad \left\{D_{1/1d}\right\} + \left\{G_{1d/6d}\right\} + \left\{G_{6d/2d}\right\} + \left\{D_{2d/2}\right\} + \left\{D_{2/2a}\right\} \\
&\quad + \left\{G_{2a/fc}\right\} + fc + \left\{G_{fc/1a}\right\} + \left\{D_{1a/1}\right\} = 0 \qquad (5.23)
\end{aligned}$$

An observable fact in equations 5.18-5.23 is the absence of the deviation torsors related to the cylindrical slides.

5.5.2.1 Compatibility Hull ($H_{compatibility}$)

The composition relations define the compatibility equations and the variables between the situation deviations and the gaps. To convert these torsor based expressions into constituent component expressions, first each torsor is displaced to a common origin in the loop and then each relation in terms of torsors yields the compatibility equations in terms of the torsor components.

Figure 5.11 shows the set of equations generated through the compatibility hull with the torsor displacement at the global origins where required. The set of compatibility equations, obtained by the application of the composition relation to all possible loops, constitutes a system of linear equations. The equations are linear because they are result of sums of situation deviations and gaps. The intrinsic deviations do not influence these equations (Dantan & Ballu, 2002).

$$
\begin{aligned}
&-\mathrm{ag1c5c} + \mathrm{ag1d6d} + \mathrm{ag2c5c} - \mathrm{ag2d6d} = 0\\
&-\mathrm{bd11c} + \mathrm{bd11d} + \mathrm{bd22c} - \mathrm{bd22d} - \mathrm{bg1c5c} + \mathrm{bg1d6d} = 0\\
&-\mathrm{gd11c} + \mathrm{gd11d} + \mathrm{gd22c} - \mathrm{gd22d} - \mathrm{gg1c5c} + \mathrm{gg1d6d} = 0\\
&240\sqrt{3}\,\mathrm{bd11c} + 240\sqrt{3}\,\mathrm{bd11d} - 240\sqrt{3}\,\mathrm{bd22c} - 240\sqrt{3}\,\mathrm{bd22d} + 240\sqrt{3}\,\mathrm{bg1c5c} +\\
&\quad 240\sqrt{3}\,\mathrm{bg1d6d} + 240\,\mathrm{gd11c} - 240\,\mathrm{gd11d} - 240\,\mathrm{gd22c} + 240\,\mathrm{gd22d} + 240\,\mathrm{gg1c5c} -\\
&\quad 240\,\mathrm{gg1d6d} - \mathrm{ug1c5cC} + \mathrm{ug1d6dD} + \mathrm{ug2c5cCC} - \mathrm{ug2d6dDD} = 0\\
&-240\sqrt{3}\,\mathrm{ag1c5c} - 240\sqrt{3}\,\mathrm{ag1d6d} + 240\sqrt{3}\,\mathrm{ag2c5c} + 240\sqrt{3}\,\mathrm{ag2d6d} - 300\,\mathrm{gd22c} +\\
&\quad 300\,\mathrm{gd22d} - \mathrm{vd11cC} + \mathrm{vd11dD} + \mathrm{vd22cCC} - \mathrm{vd22dDD} - \mathrm{vg1c5cC} + \mathrm{vg1d6dD} = 0\\
&-240\,\mathrm{ag1c5c} + 240\,\mathrm{ag1d6d} + 240\,\mathrm{ag2c5c} - 240\,\mathrm{ag2d6d} + 300\,\mathrm{bd22c} - 300\,\mathrm{bd22d} - \mathrm{wd11cC} +\\
&\quad \mathrm{wd11dD} + \mathrm{wd22cCC} - \mathrm{wd22dDD} - \mathrm{wg1c5cC} + \mathrm{wg1d6dD} = 0\\
&\qquad\vdots\\
&-\mathrm{afc1a2a} + \mathrm{ag1d6d} - \mathrm{ag2d6d} = 0\\
&-\mathrm{bd11a} + \mathrm{bd11d} + \mathrm{bd22a} - \mathrm{bd22d} - \mathrm{bfc1a2a} + \mathrm{bg1d6d} = 0\\
&-\mathrm{gd11a} + \mathrm{gd11d} + \mathrm{gd22a} - \mathrm{gd22d} - \mathrm{gfc1a2a} + \mathrm{gg1d6d} = 0\\
&240\sqrt{3}\,\mathrm{bd11d} - 240\sqrt{3}\,\mathrm{bd22d} + 240\sqrt{3}\,\mathrm{bg1d6d} - 240\,\mathrm{gd11d} + 240\,\mathrm{gd22d} - 240\,\mathrm{gg1d6d} -\\
&\quad \mathrm{ufc1a2aA} + \mathrm{ug1d6dD} - \mathrm{ug2d6dDD} = 0\\
&-240\sqrt{3}\,\mathrm{ag1d6d} + 240\sqrt{3}\,\mathrm{ag2d6d} + 300\,\mathrm{gd22d} - \mathrm{vd11aA} + \mathrm{vd11dD} + \mathrm{vd22aAA} - \mathrm{vd22dDD} -\\
&\quad \mathrm{vfc1a2aA} + \mathrm{vg1d6dD} = 0\\
&240\,\mathrm{ag1d6d} - 240\,\mathrm{ag2d6d} - 300\,\mathrm{bd22d} - \mathrm{wd11aA} + \mathrm{wd11dD} + \mathrm{wd22aAA} - \mathrm{wd22dDD} -\\
&\quad \mathrm{wfc1a2aA} + \mathrm{wg1d6dD} = 0
\end{aligned}
$$

Figure 5.11: Compatibility Equations for the example

5.5.2.2 Interface Hull $H_{Interface}$

An interface constraint limits the geometric behavior of the mechanism and characterizes non-interference constraint or press-fit constraint (Giordano, 1993; Roy & Li, 1999). These constraints define the interface hull in Gap $\times$ Intrinsic space. $H_{Interface}$ is the interface hull for assembly. Dantan & Ballu (2002) define three types of contacts between assembly components: floating contacts, slipping contacts, and fixed contacts. In the case of floating contact, depending upon the type of contact, the interface constraint results in inequalities defined in $Gap \times Intrinsic$ space. In the example presented, the features in floating contact are considered as rigid, so there is no possible interference. The relative displacement is bounded in one direction according to the normal of the contact at each point.

In the case of slipping and fixed contacts, the interface constraint specifies that a contact is enforced because of the presence of a mechanical action. This results in equations defined in $Gap \times Intrinsic$ space. Indeed, no relative displacement is possible for a fixed contact, and no relative displacement according to the degrees of contact is possible for a slipping contact.

For joints 1/4, 1/5, 1/6, the gap is limited by the non-interference constraint.The set of inequations, generated through the interface hull via Mathematica®, is given as follows:

$$\begin{aligned}(100gg1b4b + vg1b4bB)^2 &+ (-100bg1b4b + wg1b4bB)^2 \\ &\leq (d1b - d4)^2 \end{aligned} \quad (5.24)$$

$$\begin{aligned}(-100gg1b4b + vg1b4bB)^2 &+ (100bg1b4b + wg1b4bB)^2 \\ &\leq (d1b - d4)^2 \end{aligned} \quad (5.25)$$

$$\begin{aligned}(100gg1c5c + vg1c5cC)^2 &+ (-100bg1c5c + wg1c5cC)^2 \\ &\leq (d1c - d5)^2 \end{aligned} \quad (5.26)$$

$$\begin{aligned}(-100gg1c5c + vg1c5cC)^2 &+ (100bg1c5c + wg1c5cC)^2 \\ &\leq (d1c - d5)^2 \end{aligned} \quad (5.27)$$

$$\begin{aligned}(100gg1d6d + vg1d6dD)^2 &+ (-100bg1d6d + wg1d6dD)^2 \\ &\leq (d1d - d6)^2 \end{aligned} \quad (5.28)$$

$$\begin{aligned}(-100gg1d6d + vg1d6dD)^2 &+ (100bg1d6d + wg1d6dD)^2 \\ &\leq (d1d - d6)^2 \end{aligned} \quad (5.29)$$

These constraints represent the non interference constraints (Germain, 2007) for the guide-Hole system.

5.5.2.3 Functional Hull $H_{Functional}$

The functional hull defines the control for the respect of the required functional characteristics and provides the basis for a performance evaluation of the assembly. In the example model, the functional constraint is defined as the co-axiality of parts 1 and 2. This requirement is governed by the topological loops containing the functional conditions. The resolution of the functional condition topological loop gives us the constraints on the functional Hull in a manner similar to the resolution of the assembly requirement loops as expressed earlier.

The set of equations and inequations, resulting from the transformation of the hulls, form the behavior model of the assembly for the given configuration. Therefore, it allows further development of the example. Using the set of equations and inequations, the next step is the optimization technique that evaluates the constraint model.

5.5.3 Optimization

The algorithm is not fully detailed; to illustrate the non linear optimization problem, we just detail one of them: the maximization of one functional characteristic, which is shown in figure 5.12.

The maximization of functional characteristic is used in the case study to verify if, for all of the gap configurations, the functional requirement is respected or not. If the worst value of the functional characteristic respects the functional requirement then, for all the other values of the functional characteristic, the functional requirement is respected.

In our case study, according to figure 5.10, the functional characteristic is coaxiality which must not be more than the functional requirement. Therefore a numerical algorithm is used to find the global maximum of the functional characteristic subject to the constraints. In another case, the aim might be to find the minimum value. It depends on the type of the functional characteristic.

5.5.4 Assembly Example Results

Using the algorithm (developed for the statistical tolerance analysis in the figure 5.6), the constraint model (developed in section above), and the transformation of the tolerance analysis formalization (from chapter 3), multiple simulations for the tolerance analysis of the assembly were run with different domain ranges for the deviation parameters. The results of the simulation were found to be satisfactory for the nominal design dimensions of the assembly as well as for the tolerance intervals for the deviation variables in terms of the probability of assembly and for the respect of the functional characteristic.

Maximization of $(100\,gfc1a2a + vfc1a2aA)^2 + (-100\,bfc1a2a + wfc1a2aA)^2$

S.T.

$$-ag1c5c + ag1d6d + ag2c5c - ag2d6d == 0$$
$$-bd11c + bd11d + bd22c - bd22d - bg1c5c + bg1d6d == 0$$
$$-gd11c + gd11d + gd22c - gd22d - gg1c5c + gg1d6d == 0$$
$$240\sqrt{3}\;bd11c + 240\sqrt{3}\;bd11d - 240\sqrt{3}\;bd22c - 240\sqrt{3}\;bd22d + 240\sqrt{3}\;bg1c5c + 240\sqrt{3}\;bg1d6d + 240\,gd11c - 240\,gd11d - 240\,gd22c + 240\,gd22d + 240\,gg1c5c - 240\,gg1d6d - ug1c5cC + ug1d6dD + ug2c5cCC - ug2d6dDD == 0$$
$$-240\sqrt{3}\;ag1c5c - 240\sqrt{3}\;ag1d6d + 240\sqrt{3}\;ag2c5c + 240\sqrt{3}\;ag2d6d - 300\,gd22c + 300\,gd22d - vd11cC + vd11dD + vd22cCC - vd22dDD - vg1c5cC + vg1d6dD == 0$$
$$-240\,ag1c5c + 240\,ag1d6d + 240\,ag2c5c - 240\,ag2d6d + 300\,bd22c - 300\,bd22d - wd11cC + wd11dD + wd22cCC - wd22dDD - wg1c5cC + wg1d6dD == 0$$
$$-ag1b4b + ag1d6d + ag2b4b - ag2d6d == 0$$
$$-bd11b + bd11d + bd22b - bd22d - bg1b4b + bg1d6d == 0$$
$$-gd11b + gd11d + gd22b - gd22d - gg1b4b + gg1d6d == 0$$

•

•

•

$$240\,ag1c5c - 240\,ag2c5c - 300\,bd22c - wd11aA + wd11cC + wd22aAA - wd22cCC - wfc1a2aA + wg1c5cC == 0$$
$$-afc1a2a + ag1c5c - ag2c5c == 0$$
$$-bd11a + bd11c + bd22a - bd22c - bfc1a2a + bg1c5c,$$
$$-gd11a + gd11c + gd22a - gd22c - gfc1a2a + gg1c5c == 0$$
$$-240\sqrt{3}\;bd11c + 240\sqrt{3}\;bd22c - 240\sqrt{3}\;bg1c5c - 240\,gd11c + 240\,gd22c - 240\,gg1c5c - ufc1a2aA + ug1c5cC - ug2c5cCC + 240\sqrt{3}\;ag1c5c - 240\sqrt{3}\;ag2c5c + 300\,gd22c - vd11aA + vd11cC + vd22aAA - vd22cCC - vfc1a2aA + vg1c5cC == 0$$
$$240\,ag1c5c - 240\,ag2c5c - 300\,bd22c - wd11aA + wd11cC + wd22aAA - wd22cCC - wfc1a2aA + wg1c5cC == 0$$

Figure 5.12: Optimization for the example assembly

Tolerance analysis was performed using normal distribution, for each displacement of the extreme points of each hole, having a zero mean and dimension specific standard deviations derived from the specified tolerance. The program also calculates the worst case values of the gaps "g" for which the assembly conditions and functional requirements were respected. The results are shown in table 5.1

5.5.5 Error Control and Revalidation

The algorithm was programmed in Mathematica® and the optimization was done using the built in optimization functions in the program. An important step in the use of the black box functions (built into a program) is the reliability and

	Configuration		**Results**			
No.	Nominal dimensions (mm)	Standard deviation	Number of accepted samples	Number of errors	Respect probability of AR	Respect probability of FR
1	20 & 19.5	0.2/6	9996	4	9977/9996	9594/9996
2	20 & 19.8	0.2/6	8898	1102	5327/8898	5327/8898
3	20 & 19.8	0.3/6	8786	1214	1990/8786	1989/8786
4	20 & 19.8	0.5/6	9440	560	335/9440	323/9440
5	20 & 19.8	0.005/6	10000	0	9991/10000	9991/10000

Table 5.1: Results for tolerance analysis of the over-constrained mechanism

the revalidation of the results obtained through these functions. It is therefore necessary to include a revalidation routine for the results obtained in order to ascert their validity.

The revalidation assures the quality of the built-in functions as well as the quality of the algorithm and its validation. It has two steps. The first step is to program a function that revalidates the results of the optimization carried out by the built-in functions. This step helps to ensure that the results calculated by the optimization satisfy the constraint model, therefore validating the results of the calculation in an implicit way.

The second step is to program an internal error control and management system which allows reporting and management of internal exceptions of the algorithm in order to improve the algorithm results. For this purpose, a subroutine was programmed to manage and filter the instances during the algorithm iterations. This prevents the failure in optimization due to failure of built-in functions and separates such instances from the final results.

5.6 Conclusion

This chapter provides an insight into the practical application of the tolerance analysis theory developed using the quantified expressions from FOL and QCSP. The theory of tolerance analysis, developed in chapter 3, provides a necessary tool box through which the tolerance analysis problem can be expressed using a first order logic. A symbolic logical model can be constructed to express the relationships between the geometric variables and the additional consideration of the gaps. The usage of FOL enables explicit quantification of the variables and allows the specification of the assembly and functional conditions among them. The theory therefore provides the basic building blocks for the definition, expression and formalization of the tolerance analysis problem within a logical paradigm, empowering the designer and engineer to develop an expression which can later on be converted into a computable form.

In order to apply the logical framework, it is necessary to determine the methods, techniques and tools, that allow conversion of the expressions into computable form. This chapter provides an overview of the process for the choice, utilization, and development of the necessary elements that permit this application.

The first step deals with the question of linking the geometric features of the assembly under analysis in the form of a computable model that can be analyzed regarding assembly condition and the required functional conditions. The work done to describe the geometry into a computable model has been developed to benefit from the earlier research work in the field of geometric behavior modeling of the assemblies carried out by Dantan & Ballu (2002); Dantan *et al.* (2003b, 2005). The geometric modeling approach is based on modeling by the parameterization of the component. Instead of an explicit model, the model is based on a set of equations and inequations which provides the necessary constraints for product functioning and part compatibility.

The resulting model is dependent not only on the nominal dimensions of the assembly but also on the required functional characteristics and its respect as well as the respect of gaps. The 3D geometric behavior description used, has been adopted for use in the tolerance analysis theory. This geometric behavior description relies on the SDT method developed by Ballot & Bourdet (1997); Bourdet *et al.* (1996); Germain (2007).

The most important step is the application integration through transformation of the logical framework. This integration allows the SDT based geometric behavior model to be computed for the design requirements. For this purpose, an algorithm is developed that translates and transforms the logical framework into a computable algorithm based on the boundary level optimization of the quantified expressions. This optimization provides the necessary consistency to enable the gap based quantified expression evaluation and reduction for conformity to the assembly constraints as well as the subsequent consistency evaluation and validation for the respect of functional conditions for the assemblies that respect the assemblability constraints. Through this transformation and integration with Monte Carlo simulation, the work presented provides a complete framework; starting from the logical description of the problem, definition and development of the geometric behavior, and eventual transformation into the numerical problem that can be solved for a tolerance analysis problem.

The resulting framework is modular in nature and therefore permits integration with the robust design theory and application. This results in a total integrated framework of concurrent design decision making on the mechanical, manufacturing, performance, as well as assembly aspects of a mechanical product. The algorithm has also been shown to be tested as an add-on in other situations and programs such as tolerance allocation algorithms using heuristic techniques (genetic algorithms).

The applicability of the algorithm has been demonstrated with the help of an over constrained example. As discussed earlier, the tolerance analysis of over constrained assemblies requires a special attention due to unavailability of the explicit assembly response function. The example, discussed in this chapter, shows the expressive power of the developed approach and its capability to perform tolerance analysis of the over constrained systems. It also allows us to perform tolerance analysis without calculating the Minkowsky sums.

Chapter 6

Conclusion And Perspectives

6.1 Conclusion

This research provides a generalized framework for the definition, modeling, expression and interaction, of the key design variables and their variations, for the mechanical product design, using the philosophy of set based design developed by (Ward *et al.*, 1995). The developed framework integrates the notion of concurrent engineering design while considering the design parameters in terms of sets, instead of point based design, as well as the notion of noise (variation/uncertainty), arising from different sources, enabling an expression of flexible and robust design. The research also sets down the complete approach from an abstract expression of the framework description to the operational application to examples in the mechanical engineering design of products in the embodiment design phase. Successive chapters discuss: the definition of a common framework for the expression of concurrent product design problem with the management of variation, its transformation for the set based robust design and tolerance analysis applications , within an integrated design context through appropriate tools, methods and techniques.

The research brings together the advances made in the domain of the design philosophy in mechanical engineering (Scaravetti, 2004; Sobek *et al.*, 1999), descriptive logical languages and QCSP in artificial intelligence and computer science (Benhamou & Goualard, 2000; Bordeaux & Monfroy, 2006; Russell & Norvig, 2003), tolerance specification, analysis and synthesis techniques from the domain of industrial engineering (Dantan *et al.*, 2003a; Dantan & Ballu, 2002; Dantan *et al.*, 2005), tools and methods from applied mathematics to propose a solution that enables the application of concurrent integrated design practices to the product design.

The concept of the set based design is a well grounded industrial design technique practiced by popular industries. It involves thinking in terms of ranges and sets through out the product design phase instead of a point based design. The most important characteristic of the set based design, that separates this design approach from other approaches, is the flexibility and inherent robustness that arises from following such an approach. This results in an integrated management of variations arising from different dependent and/or independent sources within the global context of product design and results into products that are robust.

Various research works exist that describe this approach from a philosophical point of view in a qualitative aspect. Many research works are based on these seminal research works and provide examples of application of set based design to specific examples by the utilization of applied tools. There is however, a need for a framework, that permits the designers to express the principles of the set based concurrent design in a formal syntax, that can then be understood and transformed into specific applications by selection of appropriate tools and methods.

This thesis is based on the seminal work in decision based design, specifically in the philosophy of the set based design. It proposes a complete framework, permitting a designer to express the design needs in terms of design and noise variables simultaneously, while keeping into account the constraints and the design domain. This novel framework deals with the design problem at the embodiment phase and delineates a syntax that defines the set based interaction between the design requirements and the design model.

An abstract expression, that can take into account the concept of variability, sets and ranges, needs to have a quantification strategy for the variables involved. This strategy is proposed in this research work through the expression of the general problem in terms of first order logic. The first order logic provides a generalization and extension of the earlier research proposed in the domain of tolerance specification and synthesis by Dantan et el. These works provide the basis for the quantification of the variables in a tolerancing problem within the product design phase. Using this approach as a starting point, this thesis develops an integrated formalization, which generalizes and extends the earlier tolerance based description to a generic product design paradigm, and enables the designer to define a quantified model based on his design requirements and the variables involved. The framework provides: formal expression for the starting global design space, which include both the design and variation space; an expression enabling the statement of the design constraints in terms of a logical model along with the conditions for its quantified validity. Lastly the notion of the quantified variables proposed by Dantan & Ballu (2002) applicable to the tolerancing problem is harmonized, developed and extended to a general definition of set based design problem.

The thesis demonstrates the application of the developed approach in the domain of set based robust design and tolerance analysis. Two main chapters in the thesis are dedicated to a detailed discussion and subsequent demonstrations that outline the process of application of the framework to the above two steps in the design.

Chapter 4 presents a detailed overview of the fundamental stages that need to be addressed in order to apply the developed formalization of set based robust design in terms of a computable algorithm. The core areas discussed are: the tools, methods, and means to transform the quantified expressions (which result from the utilization of the formalization) into a computable form; a generic format for constraint adoption for application purposes; a proper algorithm that allows the usage of proposed consistencies in conjunction with a constraint model to search the design space for sets of robust solutions. The chapter therefore presents the choice of tools taken for each of the above areas in order to carry out the transformation. For consistency transformation, the chapter provides an interval arithmetic based transformation for the conversion of the quantified expressions in terms of interval bound expressions. This transformation is based on:

- The notion of QCSP (Bordeaux & Monfroy, 2006) (developed in the domain of applied mathematics)

- Application of interval arithmetic (Benhamou *et al.*, 2004; Gervet, 1997; Hansen & Walster, 2004) to the mechanical engineering design constraints

- The adaption and application of box consistency developed in the domain of computer science (Benhamou *et al.*, 1999) to problems in mechanical engineering

The developed transformation is then used for the search of robust solutions in design and variation space.

Lastly, an algorithm has been developed and presented which integrates the developed consistency evaluation techniques and performs a tree based branch and bound search of design and variation space. The algorithm filters the initial design space for the sets of robust solutions and reduces the space by rejecting the spaces without the solution. As the consistency techniques use the sets of design variables instead of a fixed points and simultaneously integrate the associated noise and variation, therefore the solutions returned are inherently robust. The developed algorithm can be validated for adherence to the philosophy of the set based design as defined in Sobek *et al.* (1999). The algorithm is capable of carrying out set based manipulations, the domain filtering for the intersections of feasible regions while analyzing the constraints in order to evaluate multiple possible solutions simultaneously and integrating the uncertainty at the same time.

The process is demonstrated with four examples. All examples start from the abstract expression of the design intent in the form of logical expressions with the definition of the variables involved, the design space and the constraints, and successively show the different steps towards the final application in a computable form with the results. The examples are validated through analytical methods that include boundary analysis as well as domain discretization. The results obtained through the examples establish the validity of the expression as well as the tools and methods developed to enable the transformation. There are, however, tool related considerations and effects, on the performance of the approach, visible in the example of the six bar mechanism, which highlight the areas of further exploration, discussed in the following section on perspectives.

In chapter 5, a discussion on the application of the developed tolerance analysis formalization to the variation and validation phase of tolerance analysis in the product design is discussed. The mathematical quantifiers expressed in the earlier formulation by Dantan & Ballu (2002) are homogenized within the framework of FOL and the fundamental questions relating to the application of such formalization in a computable form are dealt with. The chapter provides a discussion of three main areas related to the application: The tools for the representation of geometrical description; the expression of constraints to model the product; the transformation of the the quantified tolerance analysis expressions arising from the FOL expressions.

For the sake of homogeneity with some earlier works (Dantan & Qureshi, 2009), the concept of SDT for modeling the deviations is adopted. The modeling of the product geometric behavior through hulls as defined by Dantan *et al.* (2005) is used in conjunction with deviation and gap torsors to develop a constraint model of the product. The chapter then develops and proposes tolerance analysis techniques based on the transformation of the quantified expressions through optimization techniques for the worst case and the statistical tolerance analysis of the assemblies. The statistical tolerance analysis is carried out via Monte Carlo simulation and is validated for an example of over constrained mechanism. The successful application to an over-constrained mechanism shows the capability of the approach for the tolerance analysis of complex assemblies which lack an explicit assembly response function.

6.2 Perspectives

Currently the interaction between the design space and the variation space in this research work is carried out through the definition of design variables and noise variables separately. However, in the case of variations, that are associated to a given design variable, these relationships may be expressed with the help of predefined predicates that establish the relationship between the variation and the design space without the need for a declaration of the independent noise variables. This requires an extended definition of uniform predicates that can co-relate the interaction on a higher level between the design space and the variation space. This can be further categorized into sub-phases for the application to robust design and tolerance analysis therefore improving greatly the expression that is already developed.

In terms of applicability of the framework, currently the phases of the robust design and tolerance analysis of the mechanisms are considered. The framework however is expressive and promising and this formalism can be extended to apply to other functions such as process planning and selection, tolerance allocation, and other branches of industrial and mechanical engineering design.

The first order logic decision making is limited to the crisp bivalence (i.e. every declarative sentence expressing a proposition is either true or false). This results in the validation and rejection of the design space following the conversion through the box consistency and may result into a loss of information towards the frontiers in a constraint specific manner. This aspect can be improved through the possibility of integration of the many-valued logic such as fuzzy logic which can assign degrees of truth during the validation process, thereby improving the perception of the search space in a constraint sensitivity appreciation context.

In chapter 4, the transformation of the quantifier consistency for the existence of the solution and the existence of a robust solution have been done in the research work via the help of box consistency techniques based on the usage of interval arithmetic. The technique works well for simpler problems without multiple occurrences of the variables. However, as the constraint system complexity increases, multiple occurrences of the variables in a given expression increases and the technique starts to suffer from interval dependency problem. This is aggravated by the presence of trigonometric expressions. This issue has been dealt with the addition of the bisection function which in turn is computationally intensive depending upon the bisection resolution. Interval boundary validation techniques can be integrated in the algorithm in order to compensate for this shortcoming.

As seen in Chapter 4, the strategy for domain splitting and discrete variable management has an important effect on algorithm performance. The variable splitting and management strategies can therefore be explored in the context of mixed constraints (consisting of continuous and discrete variables) to improve the algorithm and avoid combinatorial explosion.

The algorithm developed for the purpose of tolerance analysis, presented in the Mathematica® software is based on deviation instances generated through Monte Carlo simulation. The optimization functions used in the program are built-in optimization functions provided by the software. In order to validate the results of the optimization, a revalidation of the constraint system for each result was carried out. This process in conjunction with the higher number of simulations required for Monte Carlo simulations may result into higher computational effort. This can be alleviated by using another approach such as Form-Sorm which requires significantly fewer number of iterations for a result of comparable satisfaction.

Appendix A

Appendix 1

A.1 Description of FOL terms and symbols

Symbol	$\longrightarrow$	Description
$\Rightarrow$	$\longrightarrow$	Implication
$\vee$	$\longrightarrow$	OR
$\wedge$	$\longrightarrow$	AND
$\Leftrightarrow$	$\longrightarrow$	Equivalence
$\forall$	$\longrightarrow$	Universal Quantifier
$\exists$	$\longrightarrow$	Existential Quantifier
$\models$	$\longrightarrow$	Satisfies

Table A.1: Syntex of First-order logic specified in Backus-Naur form. Adopted from (Russell & Norvig, 2003)

A.2 Constraints For Flange Coupling Example

A.2.1 Nomenclature

The following symbols and abbreviations are used within the model:

A_t = Tensile Stress Area
$b_{A/C}$ = Bolt head length across corners
C_1 = Torsion moment in bolt due to preload
D = Outside diameter of flange
d = nominal diameter of the shaft/ hub internal diameter
D_1 = Bolt circle diameter
D_2 = Hub outside diameter
d_2 = Pitch diameter of thread
d_n = Bolt nominal diameter
d_{ts} = Diameter of stress area
f_1 = Friction coefficient between the bolt and the flange
$F_{0_{min}}$ = Minimum bolt tightening torque
F_b = Tension load in each Bolt
i = Number of bolts
L = Length of hub
m_b = Minimum bolt center distance from edge
p = Pitch of thread
p_b = tool clearance
r_m = Mean radius of surface
S_p = Proof Strength of bolt
T = Torque to be transmitted
T_{hub} = Torque capacity based on shear of flange at the outside hub diameter
$T_{friction}$ = Torque transmission capacity due to friction
$T_{b_{shear}}$ = Torque transmitted through bolts in shear
$T_{b_{bearing}}$ = Torque capacity based on bearing of bolts
t = Thickness of flange

$$\alpha_s = \text{Accuracy factor of tightening tool}$$
$$\sigma_y = \text{Bolt yield strength}$$
$$\sigma_b = \text{Design stress in bolts}$$
$$\sigma eq_{max} = \text{Von Mise stress}$$
$$\sigma_{b_{max}} = \text{Max. tensile stress in bolt}$$
$$\mu = \text{Coefficient of friction between flange surfaces}$$
$$\tau_f = \text{Design shear stress in flange}$$
$$\tau_s = \text{Design shear stress in shaft}$$
$$\tau_{b_{max}} = \text{Max. torsional stress in the bolt}$$

A.2.2 Constraints for flange coupling design

The shaft size dictates the main bore of the Flange coupling and can be found by the following relation:

$$d = \left(\frac{16T}{\eta\pi\tau_s}\right)^{1/3} \tag{A.1}$$

Where η is a factor that takes care of the reduced strength of shaft due to keyway, T is the torque to be transmitted and d is the nominal diameter of the shaft/hub internal diameter. Also, torque in terms of power can be found out as:

$$T = \frac{9550P}{N} \tag{A.2}$$

where P is the power in kW, N are the revolutions per minute of the prime mover and T is in N-m. The outside hub diameter of the coupling has to be calculated such that the torque is transmitted safely without shear of flange at the outside hub diameter. This can be expressed as:

$$T_{hub} = t(\pi D_2)\tau_f \frac{D_2}{2} \tag{A.3}$$

$$T_{hub} \geq T \tag{A.4}$$

The Friction torque transmission capacity of the flange coupling based on the concept of the friction force acting at the mean radius of the surface is given by:

$$T_{friction} = i\mu F_b r_m \tag{A.5}$$

$$T_{friction} \geq T \tag{A.6}$$

where $r_m = \frac{D+d}{2}$ and F_b is tension load in each bolt. The length of the hub is generally calculated from the relation $L = 1.5d$ Other miscellaneous dimensional constraints related to the flange dimensions are given as follows:

$$D_1 \geq D_2 + 2b_{A/C} + 2m_b \tag{A.7}$$

$$D \geq D_1 + 2b_{A/C} + 2m_b \tag{A.8}$$

Where $b_{A/C}$ denotes the bolt head length across corners and m_b denotes the clearance between the bolt center and the edge of the hub.

The maximum shear stress calculated in the hub if it is considered as a hollow shaft should be less than allowable shear stress for the flange material:

$$\tau_{f_{calculated}} = \frac{T16}{\pi}\left(\frac{D_2}{D_2^4 - d^4}\right) \tag{A.9}$$

$$\tau_{f_{calculated}} \leq \tau_f \tag{A.10}$$

A.2.2.1 Bolt design

The bolt design for the coupling is a necessary part of the design of the coupling and therefore will be part of the example. The bolts used in this example are ISO metric standard bolts and their required mechanical, thread, dimensional properties can be found from the relevant ISO standards. The type of bolts and number of bolts to be used in the flange coupling are an important factor relevant to all the design requirements of the coupling.

The minimum number of bolts should not be less than three i.e. $i_m = 3$ and $i \geq i_m$. As the bolts are under torsional as well as tensile stress therefore the Von Mises criterion for the combined loading of the bolt is calculated. the Von Mises criterion is given by the following equation (Yvars *et al.*, 2009):

$$\sigma eq_{max} = \sqrt{\sigma^2_{b_{max}} + 3\tau^2_{b_{max}}} \tag{A.11}$$

Where:

$$\tau_{b_{max}} = 16\frac{C_1}{(\pi d_{ts}^3)} \tag{A.12}$$

$$C_1 = F_b(0.16p + 0.583d_2f_1) \tag{A.13}$$

$$d_ts = d_n - 0.938194p \tag{A.14}$$

$$\sigma_{max} = \frac{F_b}{A_t} \tag{A.15}$$

$$F_b = \alpha_s F_{0_{min}} \tag{A.16}$$

$$F_{0_{min}} = 0.75A_tS_p \tag{A.17}$$

The inequality to be respected in terms of Von Mises criterion is:

$$0.9\sigma_y \geq \sigma eq_{max} \tag{A.18}$$

In order to allow the ease of assembly and disassembly of the flange, the dimensional constraints related to the tools and the shape of bolt heads should be taken into account to ensure a considerable distance between the bolt heads. This should allow their tightening by an appropriate tool. TO do so, the allowable clearances for the bolts have to be respected such that the relevant tool can be used to tighten and loosen the bolts. The distances between the bolt centers can be found out by the following equation:

$$s = 2\pi\frac{D_1}{i} \tag{A.19}$$

The clearance between the bolts can be calculated by the taking into account the maximum distance across corners of the bolt head..

$$s_c = 2\pi\frac{D_1}{i} - b_{A/C} \tag{A.20}$$

This clearance should be larger than the minimum clearance required for the tool to work. These clearance values can be found from the tables. The resulting inequality becomes:

$$s_c \geq p_b \tag{A.21}$$

A.3 Constraints For Six Bar Mechanism

A.3.1 General Considerations

It is considered that the links are free from surface defects. The pin joints are perfectly aligned and free of friction and geometrical errors. In addition, following general considerations also apply:

- For reference related to nomenclature, please refer to Fig. 4.21
- The symbols a, b, c, d, f, g, h denote the length of the links $R_1, R_2, R_3, R_4, R_5, R_6, R_7$ and are therefore treated as scalar quantities
- R_1, R_5, R_7 Are the components that constitute the fixed link
- All the system should be contained within the frame
- The directions of position vectors are chosen as to define their angles
- The angle of a vector is always measured at its root not at its head
- All angles are measured in counter clockwise direction from horizontal
- A complex mechanism can be solved by sub division of the mechanism into sub mechanisms

In order to develop a geometrical and kinematic model for the mechanism, the constraints can be divided into three main categories: the constraints arising from the assembly conditions, the constraints arising from the framing constraints, and the constraints arising from the kinematic positioning. The following text develops and describes the analytical model by addressing these categories.

A.3.2 Assemblability Constraints

The first set of constraint that needs to be evaluated in case of kinematic chain with limited degree of freedom is to ensure if:

- Successfully Assemble with the given lengths of the members.

- Allow Free rotation of the driver link R_2

In order to ensure the above conditions, the Grashof criteria ensures the assemblability and rotation of the driver link in terms of lengths of the members of the mechanism involved. In our case, we have a six bar mechanism with one slider link. Therefore, in order to totally ascertain if the links will work, we need to have the maximum possible number of assemblability constraints possible. To develop assemblability constraints, we break down the mechanism into a four bar mechanism and a five bar mechanism. The resulting mechanisms can then be further constrained for assembly individually, thus increasing the number of constraints.

For this purpose, we propose dissociation of the system $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8$ with lengths a, b, c, d, f, g, h, i into two sub systems which are:

- Five Bar chain R_2, R_3, R_6, R_8, R_7 chain resulting from the conversion of the slider to the revolving link
- Four bar chain $R_2, R_3, R_4, (R_5, R_1)$

In order to find out the assemblability constraints and rotation constraints for the whole assembly, we will find out the intersection of the constraints arising from the sub systems defined above. For this, we will now develop the individual constraints for the sub systems. Figure A.1 is presented below for the purpose of reference.

Assemblability and rotation Constraints for Chain $R_2, R_3, R_4, (R_5, R_1)$
Refer to the Fig. A.1 (C). This chain is a classic four bar mechanism and Grashof criteria for assemblability of mechanism can be established from the following equations (Norton, 1999):

$$\sqrt{a^2 + f^2} + b \quad < \quad c + d \tag{A.22}$$

Where, a, f, b, c, d are lengths of the respective links. Equation(A.22) translates the rotation constraints for the link b which is the driver link. Similarly, equation (A.23) (Norton, 1999) translates the constraints related to the successful assembly of the system.

$$\sqrt{a^2 + f^2} \quad < \quad (b + c + d) \tag{A.23}$$

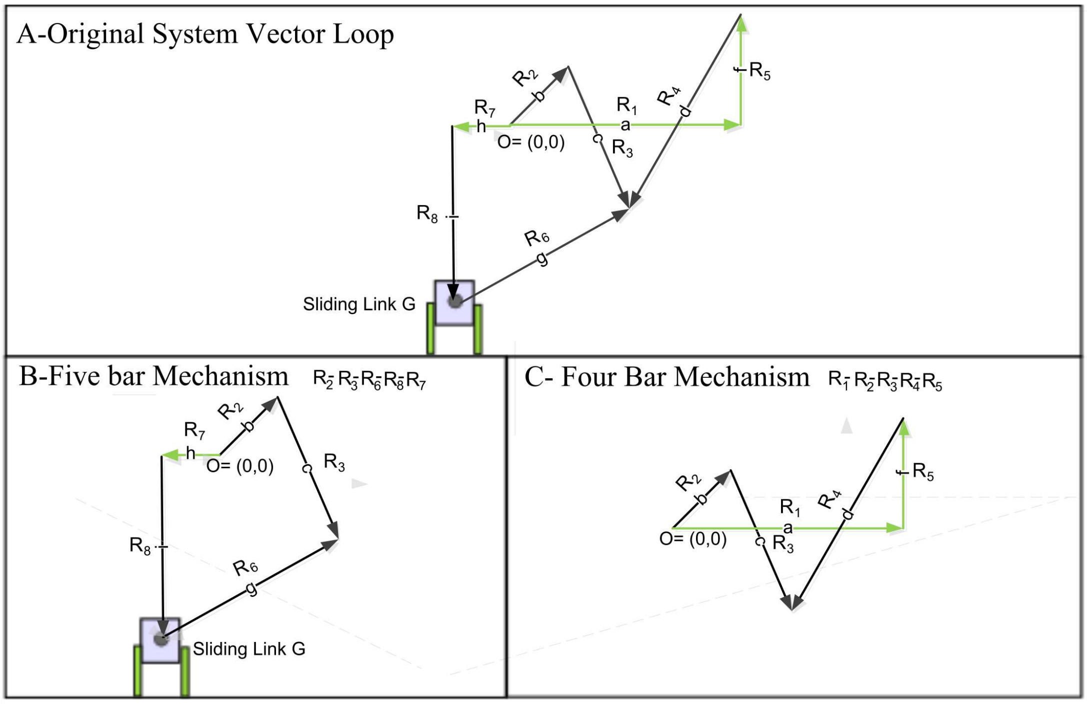

Figure A.1: Dissociation of the 6 bar mechanism

Assemblability and Rotation constraints for Chain R_2, R_3, R_6, R_8, R_7 This chain is a five bar mechanism as shown in Figure A.1(B)with a slider link. The constraints for assembly for this sub-mechanism depends on the extreme configurations of the links R_7, R_2, R_3, R_6 and $i = 0$. Therefore, for $\theta_2 = 0$, The assembly constraints to be respected becomes:

$$h + b \quad < \quad g + c \tag{A.24}$$

Where h, b, g and c are the lengths of the respective links. Similarly the second extreme position occurs at $\theta_2 = \pi$. For this configuration of R_7, R_2, R_3, R_6, the constraint to be respected are:

$$h \quad < \quad b + c + g \tag{A.25}$$

Equation A.24 and A.25 translate the constraints related to the assembly of the sub-mechanism.

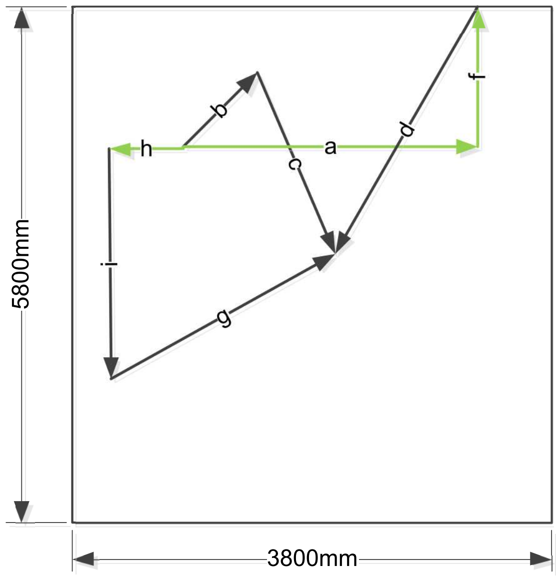

Figure A.2: Frame Constraints

A.3.2.1 Fitting and Framing Constraints

Since the mechanism is to be enclosed in a frame with specific dimensions therefore additional constraints related to the framing arise. These constraints need to be formalized, therefore, in this part, we will describe the necessary constraints that govern the design of the mechanism so that it fits in the frame.

The elementary drawing of the system is shown by figure A.2. In order to write the constraints for such a system, we will assume the following

- All the system should be contained within the frame
- The maximum width of the mechanism in any configuration should be less than the width of the frame.
- The maximum vertical length of the mechanism in any configuration should be less than the height of the frame.

Vertical framing constraints Let y be the vertical maximum dimension of the frame. Then the following constraints will ensure that the system does not exceed this maximum dimension:

$$\max \left\{ \begin{array}{l} \theta_2 \in [0,\pi], \max[\max[b\sin\theta_2, f] + i(\theta_2)] \\ \theta_2 \in [\pi, 2\pi], \max[f + i(\theta_2)] \\ \theta_2 \in [0, 2\pi], \theta_4 \in [0,\pi], \max[f + d\sin\theta_{4_{\theta_2}} + i(\theta_2)] \end{array} \right\} < y \tag{A.26}$$

(A.26) presents the constraints which make the framing of the mechanism with the given dimension y possible.

Horizontal Framing constraints Let x be the maximum horizontal dimension of the frame. Then the following constraints will ensure that the system does not exceed this maximum dimension. Two global assumptions concerning the value of h are made

$$h \in [-\infty, 0] \tag{A.27}$$

$$h \in [0, +\infty] \tag{A.28}$$

In order to treat these two different cases, two different sets of constraints based on these cases are developed.

Constraints for $h \in [-\infty, 0]$ These set of constraints assume that the offset position of the slider is to the negative side of the global coordinate system:

$$\max \left\{ \begin{array}{l} \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ |\bar{h}| < \bar{b}, \max[b\cos\theta_2 + a + d\cos\theta_{4_{\theta_2}})] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ |\bar{h}| < \bar{b}, \max[b\cos\theta_2 + a] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ |\bar{h}| > \bar{b}, \max[h + a + d\cos\theta_{4_{\theta_2}})] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ |\bar{h}| > \bar{b}, \max[h + a] \end{array} \right\} < x \tag{A.29}$$

Constraints for $h \in [0, \infty]$ These set of constraints assume that the offset position of the slider is to the positive side of the global coordinate system:

$$\max \left\{ \begin{array}{l} \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ \bar{h} < \bar{a}, \max[b \cos\theta_2 + a + d \cos\theta_{4_{\theta_2}})] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ \bar{h} < \bar{a}, \max[b \cos\theta_2 + a] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [3\pi/2, 2\pi] \cup [0, \pi/2], \\ \bar{h} > \bar{a}, \max[b \cos\theta_2 + \max[h, d \cos\theta_{4_{\theta_2}}]] \\ \\ \theta_2 \in [0, 2\pi], \theta_4 \in [\pi/2, 3\pi/2], \\ \bar{h} > \bar{a}, \max[b \cos\theta_2 + h] \end{array} \right\} < x \tag{A.30}$$

The above two formulations lay down the constraints related to the assurance of framing of mechanism within the maximum horizontal dimension x.

A.3.2.2 Path generation constraints

In order to write the constraints related to the motion generation, it is necessary to describe a vector loop model of the mechanism in order to describe the position of the mechanism in function of the driver crank position given in terms of angle. For nomenclature, the symbols and numbering as shown in figure. 4.21 are used.

For solving the vector position analysis of the system, the mechanism will be divided into two four bar systems. First system comprising of the links $a(ground), b, c$ and d only which will be solved for the angles, and then once θ_4 has been determined, the remaining system will be solved as a crank slider mechanism. comprising of links $a(ground), d, g$ and p

Position analysis of system a, b, c, d for θ_3 & θ_4 The closed vector loop equation of the first system can be written as:

$$\overrightarrow{R_2} + \overrightarrow{R_3} - \overrightarrow{R_4} - \overrightarrow{R_5} - \overrightarrow{R_1} = 0 \tag{A.31}$$

Replacing the vectors by corresponding Euler identity, we get:

$$be^{j\theta_2} + ce^{j\theta_3} - de^{j\theta_4} - fe^{j\theta_5} - ae^{j\theta_1} = 0 \tag{A.32}$$

θ_2 is independent and we need to find the expressions which define θ_3 and θ_4 in terms of θ_2 such that:

$$\theta_3 = m\{a, b, c, d, \theta_2\} \tag{A.33}$$

$$\theta_4 = n\{a, b, c, d, \theta_2\} \tag{A.34}$$

Where m and n are functions of the given variables as shown in equations A.33-A.34. θ_1=0 as it is considered to be the ground link. Substituting the euler identity in A.32:

$$\begin{aligned} b(\cos\theta_2 + j\sin\theta_2) + c(\cos\theta_3 + j\sin\theta_3) - d(\cos\theta_4 + j\sin\theta_4) \\ -f(\cos\theta_5 + j\sin\theta_5) - a(\cos\theta_1 + j\sin\theta_1) = 0 \end{aligned} \tag{A.35}$$

Separating real and imaginary parts of the equation A.35, we get the real part

$$b\cos\theta_2 + c\cos\theta_3 - d\cos\theta_4 - f\cos\theta_5 - a\cos\theta_1 = 0 \tag{A.36}$$

and the imaginary part

$$bj\sin\theta_2 + cj\sin\theta_3 - dj\sin\theta_4 - fj\sin\theta_5 - aj\sin\theta_1 = 0 \tag{A.37}$$

$\because \theta_1 = 0 \& \theta_5 = \frac{\pi}{2} \therefore$ equation. A.36 and A.37 become:

$$b\cos\theta_2 + c\cos\theta_3 - d\cos\theta_4 - a = 0 \tag{A.38}$$

$$bj\sin\theta_2 + cj\sin\theta_3 - dj\sin\theta_4 - fj = 0 \tag{A.39}$$

Dividing equation A.39 by j

$$b\sin\theta_2 + c\sin\theta_3 - d\sin\theta_4 - f = 0 \tag{A.40}$$

Writing the equation A.40 and equation A.38 in terms of θ_3:

$$c\cos\theta_3 = d\cos\theta_4 + a - b\cos\theta_2 \tag{A.41}$$

$$c\sin\theta_3 = d\sin\theta_4 + f - b\sin\theta_2 \tag{A.42}$$

Squaring and adding equation A.41 and equation A.42 and simplifying we get:

$$\begin{aligned} a^2 + f^2 - c^2 + b^2 + d^2 - 2ab\cos\theta_2 - 2bd\cos\theta_2\cos\theta_4 \\ -2bd\sin\theta_2\sin\theta_4 - 2bf\sin\theta_2 + 2ad\cos\theta_4 + 2df\sin\theta_4 = 0 \end{aligned} \tag{A.43}$$

Dividing by $2bd$ and letting $\frac{(a^2+f^2-c^2+b^2+d^2)}{2bd} = k_1$, equation A.43 becomes:

$$k_1 - \frac{a}{d}\cos\theta_2 - \frac{f}{d}\sin\theta_2 + \cos\theta_4(\frac{a}{b} - \cos\theta_2) + \sin\theta_4(\frac{f}{b} - \sin\theta_2) = 0 \tag{A.44}$$

Let:

$$
\begin{aligned}
A &= (-\frac{a}{d}\cos\theta_2 - \frac{f}{d}\sin\theta_2) && \text{(A.45)}\\
B &= (\frac{a}{b} - \cos\theta_2) && \text{(A.46)}\\
C &= (\frac{f}{b} - \sin\theta_2) && \text{(A.47)}
\end{aligned}
$$

Equation A.44 becomes:

$$k_1 + A + B\cos\theta_4 + C\sin\theta_4 = 0 \tag{A.48}$$

The above equation can now be solved for the obtaining an equation for θ_4 through half angle identity. The solution of equation A.48 can be written as:

$$
\begin{aligned}
\theta_4 &= 2\arctan\left(\frac{-2C \pm \sqrt{-4(k_1 + A - B)(K_1 + A + B) + 4C^2}}{2(K_1 + A - B)}\right) && \text{(A.49)}\\
\theta_3 &= 2\arctan\left(\frac{-2F \pm \sqrt{-4(k_2 + D - E)(K_2 + D + E) + 4F^2)}}{2(K_2 + D - E)}\right) && \text{(A.50)}
\end{aligned}
$$

Where

$$
\begin{aligned}
k_2 &= \frac{(a^2 + f^2 + b^2 + c^2 - d^2)}{(2bc)} && \text{(A.51)}\\
D &= (-\frac{a}{c}\cos\theta_2 - \frac{f}{c}\sin\theta_2) && \text{(A.52)}\\
E &= (\cos\theta_2 - \frac{a}{b}) && \text{(A.53)}\\
F &= (\sin\theta_2 - \frac{f}{b}) && \text{(A.54)}
\end{aligned}
$$

As can be seen from the results, each of θ_3 and θ_4 have two solutions. These two solutions correspond to the two circuits of the mechanism (Norton, 1999). For a mechanism that fulfills all the constraints set out in the earlier section of assemblability and rotation, the solution to the equations A.49 and A.50 should be real. An imaginary solution shows that there is no solution for a given value of lengths of the members and the position of the crank for a given value of θ_2, indicating a configuration failure.

Position analysis of θ_6 Having developed the expression for θ_3 and θ_4, we can now proceed to calculate the expression for the position of the point G. For this purpose, its is necessary to calculate and expression for θ_4. This necessitates the vector loop analysis for d, f, g, h. The vector loop equation for this system can be written as:

$$\overrightarrow{R_4} - \overrightarrow{R_5} - \overrightarrow{R_6} - \overrightarrow{R_7} = 0 \tag{A.55}$$

Replacing the individual vectors by Euler identity, we get:

$$de^{j\theta_4} - ge^{j\theta_6} - he^{j\theta_7} - ie^{j\theta_8} = 0 \tag{A.56}$$

Substituting the Euler identity, we get:

$$\begin{aligned} d(\cos\theta_4 + j\sin\theta_4) - g(\cos\theta_6 + j\sin\theta_6) - h(\cos\theta_7 + j\sin\theta_7) \\ -i(\cos\theta_8 + j\sin\theta_8) = 0 \end{aligned} \tag{A.57}$$

Separating the real and imaginary components and equating to zero, two equations are obtained from equation A.55:

$$d\cos\theta_4 - g\cos\theta_6 - h\cos\theta_7 - i\cos\theta_8 = 0 \tag{A.58}$$

$$dj\sin\theta_4 - gj\sin\theta_6 - hj\sin\theta_7 - ij\sin\theta_8 = 0 \tag{A.59}$$

As θ_7=0 and $\theta_8 = \frac{3\pi}{2}$ and j can be canceled from equation A.59, the resulting equations become:

$$d\cos\theta_4 - g\cos\theta_6 - h = 0 \tag{A.60}$$

$$d\sin\theta_4 - g\sin\theta_6 - i = 0 \tag{A.61}$$

Solving the above two equations for the vertical position of point G, equation A.61 can be written as:

$$i = d\sin\theta_4 - g\sin\theta_6 \tag{A.62}$$

The vertical position should account for the vertical offset of the link d, therefore, the final equation becomes:

$$i = d\sin\theta_4 - g\sin\theta_6 + f \tag{A.63}$$

In order to solve for θ_6, equation A.60 can be written as:

$$\theta_6 = \arccos\left(\frac{d\cos\theta_4 - h}{g}\right) \tag{A.64}$$

The value of θ_4 has already been determined earlier and the x-axis value being fixed, the link position and subsequently the position of point G can be determined exactly with help of equations A.63 and A.64

References

ABOUEL NASR, E. & KAMRANI, A. (2007). *Computer-Based Design and Manufacturing An Information-Based Approach*. Springer, New York, USA. xi, 51, 62, 63

BALLOT, E. & BOURDET, P. (1997). Computation method for the consequences of geometric errors in mechanisms. In *Proc. of the CIRP Int. Seminar on Computer-Aided Tolerancing*. 112, 183, 207, 213

BALLU, A., PLANTEC, J.Y. & MATHIEU, L. (2008). Geometrical reliability of overconstrained mechanisms with gaps. *CIRP Annals - Manufacturing Technology*, **57**, 159 – 162. 186

BENHABIB, B. (2003). *Manufacturing Design, Production, Automation, and Integration*. Marcel Dekker, New York, USA. 7, 64, 67

BENHAMOU, F. & GOUALARD, F. (2000). Universally quantified interval constraints. In *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming*, CP 02, 67–82, Springer-Verlag, London, UK. 215

BENHAMOU, F. & OLDER, W.J. (1997). Applying interval arithmetic to real, integer, and boolean constraints. *The Journal of Logic Programming*, **32**, 1 – 24. 125

BENHAMOU, F., MCALLESTER, D. & VAN HENTENRYCK, P. (1994). Clp(intervals) revisited. In *Proceedings of the 1994 International Symposium* *on Logic programming*, ILPS '94, 124–138, MIT Press, Cambridge, MA, USA. 125

Benhamou, F., Goualard, F., Granvilliers, L. & Puget, J.F. (1999). Revising hull and box consistency. In *INT. CONF. ON LOGIC PROGRAMMING*, 230–244, MIT press. 27, 28, 107, 125, 218

Benhamou, F., Goualard, F., Languenou, E. & Christie, M. (2004). Interval constraint solving for camera control and motion planning. *ACM Trans. Comput. Logic*, **5**, 732–767. 218

Beyer, H.G. & Sendhoff, B. (2007). Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, **196**, 3190 – 3218. 78, 92

Bhide, S., Ameta, G., Davidson, J. & Shah, J. (2007). Tolerance-maps applied to the straightness and orientation of an axis. In *Models for Computer Aided Tolerancing in Design and Manufacturing*, 45–54. 84

Bisztriczky, T., McMullen, P., Schneider, R. & Weiss, A. (1994). Polytopes: Abstract, convex and computational. In *Proceedings of the NATO Advanced Study Institute, Scarborough, Ontario, Canada*. 180

Bordeaux, L. & Monfroy, E. (2006). Beyond np: Arc-consistency for quantified constraints. In P. Van Hentenryck, ed., *Principles and Practice of Constraint Programming - CP 2002*, vol. 2470 of *Lecture Notes in Computer Science*, 17–32, Springer Berlin / Heidelberg. 15, 17, 27, 75, 99, 107, 125, 200, 215, 217

Bourdet, P., Mathieu, L., Lartigue, C. & Ballu, A. (1996). The concept of small displacement torsor in metrology. In *Advanced mathematical tool in metrology*, vol. 40, Advances in mathematics for applied sciences. 37, 83, 183, 184, 185, 189, 213

Chang, T.S. & Ward, A.C. (1995). Conceptual robustness in simultaneous engineering: A formulation in continuous spaces. *Research in Engineering Design*, **7**, 67–85. 71

Chase, K.W. & Parkinson, A.R. (1991). A survey of research in the application of tolerance analysis to the design of mechanical assemblies. *Research in Engineering Design*, **3**, 23–37. 11, 82, 183

Chen, H.M. (2004). *The computational complexity of quantified constraint satisfaction*. Ph.D. thesis, Ithaca, NY, USA, adviser-Kozen, Dexter. 98

Chen, W. (2007). Design optimization under uncertainty, status & promises. Panel Presentation, 7th World Congress on Structural and Multidisciplinary Optimization. 2, 78, 80

Chen, W., Allen, J.K., Tsui, K.L. & Mistree, F. (1996). A procedure for robust design: Minimizing variations caused by noise factors and control factors. *ASME Journal of Mechanical Design*, **118**, 478–485. 9, 10, 18, 77, 79, 80

Chenouard, R. (2007). *Resolution par satisfaction de contraintes appliquée à la decision en conception architecturale*. Ph.D. thesis, ParisTech Arts et Metiers. 16, 107, 127

Chenouard, R., Granvilliers, L. & Sebastian, P. (2009). Search heuristics for constraint-aided embodiment design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **23**, 175–195. 74

Committee on Theoretical Foundations for Decision Making in Engineering Design (2001). *Theoretical Foundations for Decision Making in Engineering Design*. National Academy Press, Washington, D.C. USA. xi, 67, 68, 70, 122

Cruz, J. & Barahona, P. (2001). Global hull consistency with local search for continuous constraint solving. In P. Brazdil & A. Jorge, eds., *Progress in Artificial Intelligence*, vol. 2258 of *Lecture Notes in Computer Science*, 77–83, Springer Berlin. 125

Cruz, J. & Barahona, P. (2003). Maintaining global hull consistency with local search for continuous csps. In *Global Optimization and Constraint Satisfaction*, vol. 2861 of *Lecture Notes in Computer Science*, 178–193, Springer Berlin. 125

Dantan, J., Mathieu, L. & Ballu, A. (2003a). Geometrical product requirement: uncertainty and expression. CIRP Seminar on Computer Aided Tolerancing. 20, 81, 84, 110, 216

Dantan, J.Y. (2000). *Synthèse des spécifications géométriques - modélisation par calibre à mobilités internes*. Ph.D. thesis, l'Université Bordeaux I, Bordeaux, France. 37, 38

Dantan, J.Y. & Ballu, A. (2002). Assembly specification by gauge with internal mobilities (gim)–a specification semantics deduced from tolerance synthesis. *Journal of Manufacturing Systems*, **21**, 218 – 235. 13, 16, 20, 23, 37, 46, 57, 84, 89, 93, 99, 110, 111, 112, 114, 117, 118, 181, 183, 184, 185, 189, 190, 208, 213, 216, 217, 219

Dantan, J.Y. & Qureshi, A.J. (2009). Worst-case and statistical tolerance analysis based on quantified constraint satisfaction problems and monte carlo simulation. *Computer-Aided Design*, **41**, 1 – 12. 93, 99, 109, 117, 219

Dantan, J.Y., ANWER, N. & MATHIEU, L. (2003b). Integrated tolerancing process for conceptual design. *CIRP Annals - Manufacturing Technology*, **52**, 135 – 138. 57, 213

Dantan, J.Y., Mathieu, L., Ballu, A. & Martin, P. (2005). Tolerance synthesis: quantifier notion and virtual boundary. *Computer-Aided Design*, **37**, 231 – 240. 57, 84, 93, 95, 99, 110, 112, 114, 181, 188, 213, 216, 219

Davidson, J. & Shah, J. (2003). Using tolerance-maps to represent material condition on both a feature and a datum. In *Proc. 8th CIRP Seminar on Computer Aided Tolerancing*, 8th international CIRP Seminar on Computer Aided Tolerancing, Charlotte, North Carolina, USA. 84

DESROCHERS, A. (1999). Modeling three dimensional tolerance zones using screw parameters. In *CD-ROM proceedings of 25th ASME Design Automation Conference*, 25th Design Automation Conference, ASME,. 83, 183

EAST, D. & TRUSZCZYNSKI, M. (2006). Predicate-calculus-based logics for modeling and solving search problems. *ACM Trans. Comput. Logic*, **7**, 38–83. 125

EVBUOMWAN, N.F.O., SIVALOGANATHAN, S. & JEBB, A. (1996). A survey of design philosophies, models, methods and systems. *Journal of Engineering Manufacture*, **210: 301-320**. 62

FINCH, W.W. & WARD, A.C. (1997). Set-based system for eliminateing infeasible designs in engineering problems dominated by uncertainty. In *Proc. ASME 1997 Design Engineering Technical Conf.*. 8, 73, 74, 95

GENT, I.P., NIGHTINGALE, P. & STERGIOU, K. (2005). QCSP-solve: A solver for quantified constraint satisfaction problems. In *In Proc. of Int. Joint. Conf. on Artificial Intelligence IJCAI*, 138–143, Morgan Kaufmann. 28, 98

GERMAIN, F. (2007). *Tolérancement Statistique Tridimensionnel, Intégration En CFAO*. Ph.D. thesis, L'Université de Savoie. 209, 213

GERO, J. (1990). Design prototypes: A knowledge representation scheme for design. *AI Magazine*, **11**, 26–36. 6, 62, 63, 70

GERVET, C. (1997). Interval propagation to reason about sets: definition and implementation of a practical language. *CONSTRAINTS*, 191–244. 125, 218

GIORDANO, D., M.AND DURET (1993). Clearance space and deviation space, application to three-dimensional chain of dimensions and positions. In *CIRP Seminar on Computer Aided Tolerancing*, ENS Cachan, France. 84, 113, 208

GIORDANO, M., SAMPER, S. & PETIT, J. (2005). Tolerance analysis and synthesis by means of deviation domains, axi-symmetric cases. In *9th CIRP International Seminar on Computer-Aided Tolerancing*, Arizona State University, Tempe, Arizona, USA. 84

Gu, X., Renaud, J.E., Ashe, L.M., Batill, S.M., Budhiraja, A.S. & Krajewski, L.J. (2002). Decision-based collaborative optimization. *Journal of Mechanical Design*, **124**, 1–13. 69, 70

Gupta, S. & Turner, J.U. (1993). Variational solid modeling for tolerance analysis. *IEEE Computer Graphics and Applications*, **13**, 64–74. 83, 183

Hans-Georg, B. & Bemhard, S. (2007). Robust optimization - a comprehensive survery. *Computer Methods in Applied Mechanics and Engineering*, **196**, 3190–3218. 52, 80

Hansen, E. & Walster, W.G. (2004). *Global Optimization Using Interval Analysis 2e.*. Marcel Dekker, 2nd edn. 166, 167, 218

Hong, Y.S. & Chang, T.C. (2002). A comprehensive review of tolerancing research. *International Journal of Production Research*, **40**, 2425–2459. 11, 81, 82

Huang, W., Lin, J., Bezdecny, M., Kong, Z. & Ceglarek, D. (2007a). Stream-of-variation modeling—part i: A generic three-dimensional variation model for rigid-body assembly in single station assembly processes. *Journal of Manufacturing Science and Engineering*, **129**, 821–831. 183

Huang, W., Lin, J., Kong, Z. & Ceglarek, D. (2007b). Stream-of-variation (sova) modeling—part ii: A generic 3d variation model for rigid body assembly in multistation assembly processes. *Journal of Manufacturing Science and Engineering*, **129**, 832–842. 183

Hurley, P.J. (2008). *A Concise Introduction to Logic*. Thomson Higher Education, United States. 16, 94, 95

Kamali Nejad, M. (2009). *Propositions de résolution numérique des problèmes d'analyse de tolérance en fabrication : approche 3D*. Ph.D. thesis, Université Joseph-Fourier - Grenoble I. 188, 190

Kamali Nejad, M., Vignat, F. & Villeneuve, F. (2009a). Simulation of the geometrical defects of manufacturing. *The International Journal of Advanced Manufacturing Technology*, **45**, 631–648, 10.1007/s00170-009-2001-3. 189

Kamali Nejad, M., Vignat, F. & Villeneuve, F. (2009b). Tolerance analysis in manufacturing using the MMP Comparison and evaluation of three different approaches. CIRP Conference on Computer aided Tolerancing. 189, 190

Kyung, M.H. & Sacks, E. (2003). Nonlinear kinematic tolerance analysis of planar mechanical systems. *Computer-Aided Design*, **35**, 901 – 911. 183

Liker, J., Sobek, D., Ward, A. & Cristiano, J. (1996). Involving suppliers in product development in the united states and japan: evidence for set-based concurrent engineering. *IEEE Transactions on Engineering Management*, **43**, 165–178. 8, 73

Malak, R.J., Aughenbaugh, J.M. & Paredis, C.J.J. (2009). Multi-attribute utility analysis in set-based conceptual design. *Computer-Aided Design*, **41**, 214–227. 7, 64, 71, 75

Mamoulis, N. & Stergiou, K. (2004). Algorithms for quantified constraint satisfaction problems. In *In proceedings of CP-2004*, 752–756, Springer. 27, 99, 107, 125

Maropoulos, P. & Ceglarek, D. (2010). Design verification and validation in product lifecycle. *CIRP Annals - Manufacturing Technology*, **59**, 740 – 759. 56, 64, 76, 177

Martin, J. & Simpson, T. (2006). A methodology to manage system-level uncertainty during conceptual design. *Journal of Mechanical Design*, **128**, 959–968. 54

Massa, F., Lallemand, B. & Tison, T. (2009). Fuzzy multiobjective optimization of mechanical structures. *Computer Methods in Applied Mechanics and Engineering*, **198**, 631 – 643. 17, 95

McDowell, D., Panchal, J., Choi, H., Seepersad, C., Allen, J. & F., M. (2010). *Integrated Design of Multiscale, Multifunctional Materials and Products*. Elsevier, Oxford, UK. 6, 54, 61, 62

MISTREE, F., SMITH, W., BRAS, B. & MUSTER, A. (1990). Decision-based design: A contemporary paradigm for ship design. *Transactions, Society of Naval Architects and Marine Engineers*, **98**, 565–597. 5, 64, 69, 90, 91

NIGAM, S. & TURNER, J. (1995). Review of statistical approaches to tolerance analysis. *Computer-Aided Design*, **27**, 6–15(10). 11, 82, 83, 193

NIKOLAIDIS, E., GHIOCEL, D.M. & SINGHAL, S. (2008). *Engineering Design Reliability Applications*. Taylor and Francis Group, Newy York USA. 71

NORTON, R.L. (1999). *Design of Machinery: An introduction to the synthesis and analysis of mechanisms and machines*. McGraw-Hill, 2nd edn. 229, 235

NORTON, R.L. (2005). *Machine Design: An Integrated Approach (3rd Edition)*. Prentice Hall. 153

PAHL, G. & BEITZ, W. (1996). *Engineering Design-A Systematic Approach*. Springer-Verlag, London. 6, 51, 63, 70, 80

PARSONS, M.G., SINGER, D.J. & SAUTER, J.A. (1999). A hybrid agent approach for set-based conceptual design. In *Proceedings of the 10th International Conference on Computer Applications in Shipbuilding*. 74

PARSONS, S. & DOHNAL, M. (1992). Qualitative, semiqualitative and interval algebras, and their application to engineering problems. *Engineering Applications of Artificial Intelligence*, **5**, 553 – 559. 28, 126, 127

PARUNAK, H.V.D., WARD, A.C., FLEISHER, M. & SAUTER, J. (1997). A marketplace of design agents for distributed concurrent set-based design. In *ISPE International Conference on Concurrent Engineering: Research and Applications (ISPE/CE97)*, ISPE. 73

PASUPATHY, K.T.M., MORSE, E.P. & WILHELM, R.G. (2003). A survey of mathematical methods for the construction of geometric tolerance zones. *Journal of Computing and Information Science in Engineering*, **3**, 64–75. 183

PETERS, S. & WESTERSTAHL, D. (2006). *Quantifiers in Language and Logic*. Oxford university Press, United States. 18, 97, 100

PHADKE, M. (1989). *Quality Engineering using robust Design*. Prentice Hall, Englewood Cliffs, NJ. 7, 9, 64, 76

QURESHI, A.J., DANTAN, J.Y., BRUYERE, J. & BIGOT, R. (2010). Set based robust design of mechanical systems using the quantifier constraint satisfaction algorithm. *Engineering Applications of Artificial Intelligence*, **23**, 1173 – 1186. 99

REKUC, S. (2005). *Eliminating Design Alternative Under Interrval-Based Uncertainty*. Master's thesis, George W. Woodrruff School of Mechanical Engineerring, Georgia Institute of Technology. 54, 72

ROY, U. & LI, B. (1999). Representation and interpretation of geometric tolerances for polyhedral objects. ii.: Size, orientation and position tolerances. *Computer-Aided Design*, **31**, 273 – 285. 83, 113, 183, 208

ROY, U., LIU, C. & WOO, T. (1991). Review of dimensioning and tolerancing: representation and processing. *Computer-Aided Design*, **23**, 466 – 483. 11, 82

RUSSELL, S. & NORVIG, P. (2003). *Artificial Intelligence, A Modern Approach*. Prentice Hall, United States. 15, 18, 96, 215, 223

SÉBASTIAN, P., CHENOUARD, R., NADEAU, J. & FISCHER, X. (2007). The embodiment design constraint satisfaction problem of the bootstrap facing interval analysis and genetic algorithm based decision support tools. *International Journal on Interactive Design and Manufacturing*, **1**, 99–106, 10.1007/s12008-007-0014-4. 8, 74

SCARAVETTI, D. (2004). *Formalisation Préalable d'un Problème de Conception, Pour l'aide a la Décision en Conception Préliminaire*. Ph.D. thesis, ParisTech Arts et Métiers. xi, 7, 64, 65, 66, 215

SCHUËLLERA, G. & JENSENB, H. (2008). Computational methods in optimization considering uncertainties - an overview. *Computer Methods in Applied Mechanics and Engineering*, **198**, 2–13. 52, 80

Scott, M. & Antonsson, E. (2000). Using indifference points in engineering decisions. In *In: Proceedings of the ASME/DETC2000/DTM, Baltimore, Maryland*, 10–13. 142

Shigley, J., Mischke, C. & Budynas, R.A. (2003). *Mechanical Engineering Design)*. McGraw-Hill. 153

Simpson, T.W., Rosen, D., Allen, J.K. & Mistree, F. (1998). Metrics for assessing design freedom and information certainty in the early stages of design. *Journal of Mechanical Design*, **120**, 628–635. 74

Sobek, D., Ward, A. & Liker, J. (1999). Toyota's principles of set-based concurrent engineering. *Sloan Management Review*, **40**, 67–83. 8, 72, 73, 80, 215, 218

Soderberg, R. & Johannesson, H. (1999). Tolerance chain detection by geometrical constraint based coupling analysis. *Journal of Engineering Design*, **10**, 5–24(20). 112

Suh, N. (1990). *The Principles of Design*. Oxford University Press, New York. 70

Suh, N. (2001). *Axiomatic Design, Advances and Applications*. Oxford University Press, New York. 5, 6, 7, 61, 64, 69, 70

Taguchi, G. (1978). Off-line and on-line quality control systems. *Proc. of International Conference on Quality Control*. 7, 64, 79

Taguchi, G. (1987). *System of experimental design*. Unipub/Kraus International Publication, New York. 1, 2, 7, 9, 64, 76, 79

Teissandier, D., Couétard, Y. & Gérard, A. (1999). A computer aided tolerancing model: proportioned assembly clearance volume. *Computer-Aided Design*, **31**, 805 – 817. 84, 183

Thornton, A.C. (1996). The use of constraint-based design knowledge to improve the search for feasible designs. *Engineering Applications of Artificial Intelligence*, **9**, 393 – 402. 16, 92, 105

TSANG, E. (1993). *Foundations of Constraint Satisfaction*. Academic Press Limited, United Kingdom. 91

UEBELHART, S. (2006). *Non-Deterministic Design and Analysis of Parameterized Optical Structures during Conceptual Design*. Ph.D. thesis, Massachusetts Institute of Technology. xi, 69, 70

ULLMAN, D.G. (2003). *The Mechanical Design Process*. McGraw-Hill, Newy York USA. 6, 61, 68, 71

VAREILLES, E. (2005). *Conception et approches par propagation de contraintes : contribution à la mise en oeuvre d'un outil d'aide interactif*. Ph.D. thesis, L'institut national polytechnique Toulouse, France. 28, 74, 126, 127

VAREILLES, L., ALDANONDO, M. & GABORIT, P. (2009). How to take into account piecewise constraints in constraint satisfaction problems. *Engineering Applications of Artificial Intelligence*, **22**, 778 – 785. 74, 137

VERGER, G. & BESSIERE, C. (2006). Blocksolve: A bottom-up approach for solving quantified csps. In *Proc. of 12th International Conference, CP 2006*, 635–649, Springer. vii, 98

VIGNAT, F., VILLENEUVE, F. & KAMALI NEJAD, M. (2010). Analysis of the deviations of a casting and machining process using a model of manufactured parts. *CIRP Journal of Manufacturing Science and Technology*, **2**, 198 – 207, sustainable Development of Manufacturing Systems. 189

VLAHINOS, A., KELLY, K., D'ALEO, J. & STATHOPOULOS, J. (2003). Effect of material and manufacturing variations on membrane electrode assembly pressure distribution. *ASME Conference Proceedings*, **2003**, 111–120. xi, 77, 79, 80

WARD, A. & SEERING, W.P. (1993a). The performance of a mechanical design 'compiler'. *Journal of Mechanical Design*, **115**, 341–345. 8, 73

WARD, A., LIKER, J.K., CRISTIANO, J.J. & SOBECK, D.K.I. (1995). The second toyota paradox: How delaying decisions can make better cars faster. *Sloan Management Review*, **36**, 43–61. xi, 8, 71, 72, 73, 215

Ward, A.C. (1989). *A Theory of Quantitative Inference for Artifact Sets, Applied to a Mechanical Design Compiler*. Ph.D. thesis, Massachusetts Institute Of Technology. 17, 74

Ward, A.C. & Seering, W.P. (1993b). Quantitative inference in a mechanical design 'compiler'. *Journal of Mechanical Design*, **115**, 29–35. 8, 73

Whitney, D. (2004). *Mechanical Assemblies, their Design, Manufacture, and Role in Product Development*. Oxford University Press, New York, USA. 55

Wood, K. & Antonsson, E. (1989). Computations with imprecise paramteres in engineering design: background and theory. *ASME Journal of Mechanics, Transmissions and Automation in Design*, **111**, 616–625. 142

Yannou, B., Simpson, T.W. & Barton, R.R. (2003). Towards a conceptual design explorer using metamodeling approaches and constraint programming. *ASME Conference Proceedings*, **2003**, 605–614. 74, 93

Yannou, B., Troussier, N., Chateauneuf, A., Boudaoud, N. & Scaravetti, D. (2007). Preliminary dimensioning, when and how to use design exploration, robust and reliable design approaches. CPI 2007. 5th International conference on integrated design and prodution. 16, 74, 93, 142

Yannou, B., Troussier, N., Chateauneuf, A., Boudaoud, N. & Scaravetti, D. (2009). Dimensioning a product in preliminary design through different exploration techniques. *International Journal of Product Development*, **9**, 140+. 8, 70, 93

Yvars, P., Lafon, P. & Zimmer, L. (2009). Optimization of mechanical system: contribution of constraint satisfaction method. In *40th International Conference on Computers and Industrial Engineering (CIE), 2010*. 8, 151, 226

Yvars, P.A. (2008). Using constraint satisfaction for designing mechanical systems. *International Journal on Interactive Design and Manufacturing*, **2**, 161–167. 8, 16, 74, 91, 93

Yvars, P.A. (2009). A csp approach for the network of product lifecycle constraints consistency in a collaborative design context. *Engineering Applications of Artificial Intelligence*, **22**, 961 – 970, artificial Intelligence Techniques for Supply Chain Management. 93

Ziegler, G.M. (1994). *Lectures on Polytopes. Graduate texts in Mathematics*, vol. 0152. Springer. 180

Gestion de Variation Pendant Conception Produit par Logique Formelle

Résumé: Ces travaux sont situés sur les domaines de la conception mécanique, l'informatique, et les mathématiques appliqués pour proposer une solution globale de gestion des variations pendant le processus de la conception d'un produit afin de rechercher l'ensemble des solutions robustes. Ils visent plus particulièrement à répondre à la question : "Comment exprimer et intégrer la variabilité admissible des paramètres du produit (espace des solutions), et sur le tolérancement de celles-ci : les variations admissibles (tolérances) simultanément pendant la phase de la conception d'un produit mécanique le plutôt possible ? Pour répondre à cette question, cette thèse présente, une formalisation généralisée pour définir, modéliser et exprimer la problème de la gestion des variations dans le processus de la conception du produit utilisant la philosophie de la conception basée sur les ensembles "Set Based Design" et la logique formelle. Cette formalisation permet la prise en compte simultanée des incertitudes et variations lors de la définition d'un produit. Les travaux aussi développent et expose les outils pour appliquer la formalisation développé au recherche des ensembles des solutions robustes des système mécanique et l'analyse des tolérances des mécanismes hyperstatiques.

Mots Clés: Conception Robuste, logique formelle, analyse des tolérances, gestion de variation, conception ensembliste

Variation Management in Product Design Phase Via Logic

Resume: This work is oriented towards the variation management within the embodiment design phase of a product. It focuses on management and integration of design parameter variation and manufacturing variations in design phase. In this work, a generalized framework for definition, modeling, expression and interaction, of the design variables and the variations, for the mechanical product design, using the philosophy of set based design developed and logic has been developed. The developed framework integrates the notion of concurrent engineering design while considering the design parameters in terms of sets, instead of point based design, as well as the notion of the noise (variation/uncertainty), arising from different sources, enabling an expression of flexible and robust design. The research also sets down the complete approach from abstract expression of the framework description to the operational application to examples in the mechanical engineering design of products in the embodiment design phase. This is achieved through definition of a common framework for expression of concurrent product design problem with management of variation, its transformation for the set based robust design and tolerance analysis applications , within an integrated design context through appropriate tools, methods and techniques.

Mots Clés: Robust design, Tolerance analysis, First Order Logic, Variation Management, Set Based Design

ARTS ET MÉTIERS ParisTech

ParisTech INSTITUT DES SCIENCES ET TECHNOLOGIES PARIS INSTITUTE OF TECHNOLOGY