

Titre: Etude de la propagation des erreurs mécaniques dans les assemblages grâce à une approche vectorielle basée sur les torseurs
Title:

Auteur: Bruno Dufour
Author:

Date: 1997

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Dufour, B. (1997). Etude de la propagation des erreurs mécaniques dans les assemblages grâce à une approche vectorielle basée sur les torseurs [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/8637/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/8637/>
PolyPublie URL:

Directeurs de recherche: Christian Mascle, & Éric Dupinet
Advisors:

Programme: Non spécifié
Program:

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI[®]
800-521-0600

UNIVERSITÉ DE MONTRÉAL

**ÉTUDE DE LA PROPAGATION DES ERREURS
MÉCANIQUES DANS LES ASSEMBLAGES
GRÂCE À UNE APPROCHE VECTORIELLE BASÉE
SUR LES TORSEURS**

**BRUNO DUFOUR
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL**

**MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
JANVIER 1997**



**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

395 Wellington Street
Ottawa ON K1A 0N4
Canada

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-42903-2

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

**ÉTUDE DE LA PROPAGATION DES ERREURS MÉCANIQUES
DANS LES ASSEMBLAGES GRÂCE À UNE APPROCHE
VECTORIELLE BASÉE SUR LES TORSEURS**

présenté par: DUFOUR Bruno

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. MAYER René, Ph. D., président

M. MASCLE Christian, Ph. D., membre et directeur de recherche

M. DUPINET Éric, Ph. D., membre et codirecteur de recherche

M. HOUMMADY Abdellah, Ph. D., membre et Président d'ACT inc.

REMERCIEMENTS

Je voudrais remercier tout particulièrement le docteur Éric Dupinet pour son support scientifique au cours des mois passés dans la mise au point des méthodes d'analyse présentées dans ce mémoire. De plus, j'aimerais affirmer ma profonde gratitude à madame Nathalie Malo pour son aide lors de la préparation de ce travail. Je remercie le docteur Christian Mascle pour le support administratif qu'il m'a donné lors de la finalisation de ce mémoire. Je remercie monsieur Serge Gravelle pour son aide au cours de la rédaction du programme informatique utilisé pour l'analyse des mécanismes.

RÉSUMÉ

Il est un fait bien connu que les mécanismes sont très sensibles à l'accumulation des défauts géométriques de leurs composantes. En effet, ces défauts affectent les coûts de fabrication et conduisent souvent à des variations de performances des produits et ce, de façon imprévisible.

Plusieurs études des effets des accumulations d'erreurs mécaniques ont été réalisées depuis quelques années. Un grand nombre de ces études ont été effectuées à l'aide des méthodes du calcul différentiel, malgré que nous ayons constaté qu'il existe d'importantes contraintes à son application pour l'analyse des erreurs mécaniques et des jeux. Aussi, ce travail présente-t-il une nouvelle approche mathématique facilitant l'étude des défauts géométriques et des jeux dans les systèmes mécaniques tridimensionnels. Nous avons développé cette procédure d'analyse de manière à ce qu'elle puisse être automatisable.

La méthode d'analyse proposée est basée sur la théorie des torseurs et la mathématique des exponentielles vectorielles. Ces outils mathématiques permettent d'intégrer efficacement les tolérances dimensionnelles, les tolérances géométriques et les jeux dans un même modèle d'accumulation. L'analyse des erreurs mécaniques et des jeux nécessite la compréhension de plusieurs facteurs d'importances. En effet, la réussite d'un assemblage mécanique et l'assurance d'un fonctionnement optimal demandent une analyse adéquate incluant les défauts géométriques et les jeux, les déformations surfaciques et volumiques, ainsi que les efforts externes agissant sur le mécanisme. Certaines hypothèses simplificatrices ont été effectuées pour développer la méthode d'analyse vectorielle:

1. nous considérons que les mécanismes étudiés sont constitués d'un ensemble de pièces et de liaisons mécaniques formant une ou plusieurs boucles fermées;

2. nous considérons les déformations volumiques dues à des efforts comme absentes ou négligeables;
3. nous considérons les déformations surfaciques dues à des efforts comme absentes ou négligeables;
4. nous considérons les composantes mécaniques comme des corps indéformables (compte tenu des hypothèses 2 et 3);
5. nous considérons les composantes mécaniques comme étant en situation réelle et possédant des défauts géométriques (tolérances dimensionnelles et tolérances géométriques);
6. nous proposons que les liaisons possèdent des jeux;
7. nous définissons une condition fonctionnelle comme un espace contrôlé entre deux composantes;
8. nous considérons que la forme d'une surface est parfaite.

La validation de la méthodologie est réalisée à l'aide du générateur d'équations à quatre membrures régulièrement employée en milieu académique. Nous calculons l'erreur de sortie connaissant les tolérances dimensionnelles sur chacune des membrures. Nous comparons les résultats obtenus par notre nouvelle approche avec ceux obtenus grâce à la méthode différentielle. Les résultats confirment la validité de notre nouvelle méthode d'analyse. En effet, les différences numériques observées entre les résultats des deux méthodes sont négligeables.

Un variateur mécanique de vitesse est analysé afin de démontrer les possibilités pratiques de l'emploi de la nouvelle méthode. Sur ce modèle, nous calculons les jeux fonctionnels du mécanisme connaissant le tolérancement dimensionnel, le tolérancement géométrique ainsi que la condition fonctionnelle imposée.

En résumé, ce mémoire présente une nouvelle approche basée sur certaines théories vectorielles, pour analyser les problèmes d'accumulation des défauts

géométriques et des jeux. Les qualités systématiques et procédurales de cette nouvelle méthode d'analyse permettent d'envisager avec réalisme l'informatisation de cette démarche.

Plusieurs travaux de recherche et de développement peuvent être entrepris pour compléter les concepts mis en place dans ce mémoire:

- Création d'une procédure d'optimisation des produits mécaniques.
- Introduction des concepts de logique floue pour définir un tolérancement adéquat en fonction de multiples facteurs techniques.
- Résolution d'un problème complet, car nous n'avons présenté que l'analyse partielle d'un cas pratique.
- Création d'un logiciel intégrant les concepts de ce mémoire.
- Développement de la méthodologie pour l'analyse multi-cycles.
- Introduction de l'effet des déformations dans la méthodologie.
- Introduction d'une méthode d'estimation des efforts dans le mécanisme.

ABSTRACT

It is a historical fact that mechanisms are sensitive to their geometrical errors. These errors affect fabrication costs and often lead to unpredictable performance variations.

Some studies have been conducted on the effects of mechanical error accumulation. A lot of them are based on differential computation although, we have observed difficulties in their application in the analysis of mechanical errors and clearances. For this reason, we propose a new mathematical model to facilitate the analysis of clearances and mechanical errors in 3D mechanisms. In addition, this new model is appropriate for the computation analysis of mechanical errors.

Our new mathematical model is based on the screw theory and the vectorial exponential mathematics. These two mathematical tools easily integrate dimensional tolerances, geometrical tolerances and gaps in a unique accumulation model. A good understanding of some factors is essential to a valid mechanical errors analysis. These factors include geometrical errors, gaps, superficial deformations, volumic deformations and external forces acting on the mechanism. Some hypothesis are proposed:

1. We consider that mechanisms are constituted from groups of parts and mechanical links organized through one or more functional cycles.
2. We consider that volumic deformations are negligible.
3. We consider that superficial deformations are negligible.
4. We consider that mechanical components are undeformable shapes.
5. We consider that mechanical components have real dimensions including tolerances.
6. We propose that each joint has a clearance.
7. We define that a functional condition has a spacing between two parts or has

a backlash.

8. We consider that each surface has a perfect form.

The methodology validation is realized using the four bar linkage equation generators. We calculate the output error knowing the dimensionnal errors of each bar. The results are compared with the results generated by our new model. These results confirm the validity of the new methodology.

A mechanical speed variator is analyzed in order to demonstrate practical use of the new methodology. On this model, we compute fonctionnal gaps knowing dimensionnal tolerances, geometrical tolerances and the fonctionnal condition.

To conclude, this thesis proposes a new methodology based on some vectorial theories used to analyze mechanical errors acumulation problems. This new method will permit to computarize effectively this kind of mechanical analysis.

Some works could be done to complete this research:

- Create an optimization procedure for mechanical products.
- Solve a complete engineering problem.
- Build a software using this new methodology.
- Upgrade the methodology to deal with more than one fonctionnal cycle.
- Introduce the forces effects and the deformations effects in the methodology.

TABLE DES MATIÈRES

	page
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES TABLEAUX	xii
LISTE DES FIGURES	xiii
LISTE DES SIGLES ET ABRÉVIATIONS	xiv
LISTE DES ANNEXES	xix
INTRODUCTION	1
1.0 PROBLÉMATIQUE	4
2.0 REVUE BIBLIOGRAPHIQUE	8
3.0 FONDEMENTS MATHÉMATIQUES DE LA MÉTHODOLOGIE D'ANALYSE	12
3.1 Fondements mathématiques	13
3.1.1 Déplacement d'un point et champ de déplacement	13

3.1.2	Les torseurs de petits déplacements	17
3.2	Tolérances et Jeux	25
4.0	MODÈLE D'ACCUMULATION DES ERREURS ET DES JEUX DANS UN MÉCANISME	36
4.1	Conditions fonctionnelles et <i>assemblabilité</i>	38
4.2	Boucles fonctionnelles multiples	42
4.3	Référentiel proposé	43
4.4	Résumé de la procédure d'analyse	43
5.0	VALIDATION DE LA MÉTHODOLOGIE	46
5.1	Définition du problème	47
5.2	Expérimentation	56
6.0	ANALYSE D'UN CAS PRATIQUE	63
6.1	Étude préliminaire du mécanisme	63
6.2	Développements symboliques	70
6.2.1	Torseurs d'erreurs et de jeux	70
6.2.2	Accumulation vectorielle des erreurs	74
6.3	Résultat de l'étude	82
6.3.1	Analyse des résultats	92
7.0	LOGICIEL DE CALCUL PAR TORSEUR	95
	CONCLUSION	96
	BIBLIOGRAPHIE	100
	ANNEXES	104

LISTE DES TABLEAUX

	page
Tableau 3.1 Liaisons cinématiques standards	34
Tableau 6.1 Valeurs numériques des tolérances associées à la demi-boucle 1 .	83
Tableau 6.2 Valeurs numériques des tolérances associées à la demi-boucle 2 .	83
Tableau 6.3 Valeurs numériques des dimensions nominales associées à la demi-boucle 1	84
Tableau 6.4 Valeurs numériques des dimensions nominales associées à la demi-boucle 2	84

LISTE DES FIGURES

	page
Figure 1.1 Boucles d'un mécanisme	6
Figure 3.1 Modèle du mouvement infinitésimal d'un corps	14
Figure 3.2 Illustration d'un corps se déplaçant durant une période dt	17
Figure 3.3 Transport du torseur $\{T\}$ du point a au point b	23
Figure 3.4 Changement de référentiel du torseur $\{T\}$ du réf. A au réf. B ...	24
Figure 3.5 Représentation d'une tolérance dimensionnelle par un torseur ...	26
Figure 3.6 Tolérances dimensionnelles et tolérances géométriques modélisées par les torseurs de petits déplacements	30
Figure 3.7 Solides virtuels soumis à un effort F	32
Figure 3.8 Modélisation d'un jeu par torseur de petits déplacements	33
Figure 5.1 Générateur d'équations à quatre membrures	46
Figure 5.2 Générateur d'équations, boucles et torseurs d'erreurs	48
Figure 5.3 Schéma des vecteurs composants les vecteurs d'erreurs totaux. ...	52
Figure 5.4 Essai pour ΔR_x à $+0.05$	59
Figure 5.5 Erreurs dépassant les cinq degrés	61
Figure 6.1 Aperçu du variateur mécanique de vitesse	66
Figure 6.2 Schéma cinématique du variateur mécanique de vitesse	67
Figure 6.3 Diagramme des liaisons du variateur mécanique de vitesse	68
Figure 6.4 Diagramme explosé des liaisons du variateur mécanique de vitesse	69
Figure 6.5 Représentation du vecteur B sur les dents de l'engrenage	74
Figure 6.6 Projection des erreurs et des jeux sur la contrainte fonctionnelle B	76
Figure 6.7 Cas extrêmes des déviations engendrées par les tolérances géométriques	87

LISTE DES SIGLES ET ABRÉVIATIONS

m_{old}	vecteur associé au point $m_{old} \in (S)$, exprimé dans R_0
m_{new}	vecteur associé au point $m_{new} \in (S)$, homologue du point m_{old} , après transformation, exprimé dans R_0
t	vecteur de translation du solide (S) exprimé dans R_0
t	vecteur déplacement définissant la translation du point m_1 au point m_1'
t	vecteur de translation quelconque
$[A]$	matrice régulière orthogonale directe exprimant une rotation du solide (S) autour d'un point ou d'un axe
$\exp()$	opérateur exponentiel vectoriel
$R()$	opérateur de rotation vectorielle (employé par le logiciel)
$T()$	opérateur de translation vectorielle (employé par le logiciel)
α	angle de rotation autour de l'axe du vecteur unitaire u
β	angle de rotation autour de l'axe du vecteur unitaire v
β	scalaire quelconque
γ	angle de rotation autour de l'axe du vecteur unitaire w
x	longueur de translation selon l'axe du vecteur unitaire u
y	longueur de translation selon l'axe du vecteur unitaire v
z	longueur de translation selon l'axe du vecteur unitaire w
u	vecteur unitaire d'un référentiel orthonormé direct
v	vecteur unitaire d'un référentiel orthonormé direct
w	vecteur unitaire d'un référentiel orthonormé direct
(S)	solide de nom S
(S_0)	solide de nom S_0
(S_1)	solide de nom S_1
\times	symbole du produit vectoriel
$+$	symbole d'addition

-	symbole de soustraction
/	symbole de la division scalaire
.	symbole du produit scalaire (comoment)
Σ	symbole d'ensemble de sommation
\in	symbole d'appartenance
=	symbole d'égalité
\doteq	symbole de correspondance
\otimes	symbole de projection vectorielle
!	symbole de l'opérateur factoriel
$\sin ()$	symbole de la fonction mathématique sinus
$\cos ()$	symbole de la fonction mathématique cosinus
Ref	référentiel quelconque
Ro	référentiel Ro
R0	référentiel R0
R1	référentiel R1
d τ	différentiel de temps
n	scalaire quelconque
A	scalaire quelconque
X	scalaire quelconque
B	scalaire quelconque
G	scalaire quelconque
A1	scalaire quelconque
A2	scalaire quelconque
a	composante scalaire d'un vecteur déplacement d
b	composante scalaire d'un vecteur déplacement d
c	composante scalaire d'un vecteur déplacement d
δ	composante scalaire d'un vecteur rotation λ
ζ	composante scalaire d'un vecteur rotation λ

η	composante scalaire d'un vecteur rotation λ
π	constante mathématique (3.1415...)
d	vecteur déplacement
λ	vecteur rotation
F	vecteur force quelconque
m_1'	vecteur déplacement identifiant le point m_1'
m_1	vecteur déplacement identifiant le point m_1
m_2'	vecteur déplacement identifiant le point m_2'
m_2	vecteur déplacement identifiant le point m_2
$(m_1 m_2)$	vecteur déplacement liant le point m_1 au point m_2
(ab)	vecteur déplacement liant le point a au point b
(mn)	vecteur déplacement liant le point m au point n
A	point A
B	point B
m	point m (quelconque)
n	point n (quelconque)
$D ()$	opérateur déplacement vectoriel
(T)	transformation de l'opérateur exponentiel vectoriel
(R)	résultante du torseur initial
(M)	moment de la résultante du torseur initial
(R')	résultante du torseur final
(M')	moment de la résultante du torseur final
$\{T\}$	torseur quelconque
$\{T1\}$	torseur quelconque T1
$\{T2\}$	torseur quelconque T2
$\{Ji\}_{ref/M}$	torseur de jeu d'indice i défini au point M dans le référentiel <i>ref</i>
j_i	vecteur résultant du jeu
Ψ_i	vecteur moment de la résultante du jeu
$\{E_y^{(x)}\}_{ref/M}$	torseur d'erreurs dimensionnelles et d'erreurs géométriques associé

	à la demi-boucle "x". "y" est l'indice de la composante sur laquelle l'erreur est appliquée. Ce torseur est défini au point M dans le référentiel ref
$\{J_y^{(x)}\}_{ref/M}$	torseur des jeux associé à la demi-boucle "x" pour chacune des liaisons d'indice "y". Ce torseur est défini au point M dans le référentiel ref
$\{C\}_{ref/M}$	torseur de la condition fonctionnelle à satisfaire sur la liaison de coupure. On notera que dans ce cas les liaisons sont dites faibles (liaisons unilatérales par contact) puisque rien ne doit s'opposer au déplacement dans la direction de R ou M
$\{T_y^{(x)}\}_{ref/M}$	torseur d'inter-effort d'une liaison d'indice "y", possédant un jeu et contenue dans la demi-boucle "x". Ce torseur est défini au point M et dans le référentiel ref
$(Dim_y^{(x)})$	torseur des dimensions nominales des composantes "y" associées à chacune des demi-boucles fonctionnelles "x"
s	nombre de composantes dans la demi-boucle 1
t	nombre de composantes dans la demi-boucle 2
n	nombre de liaisons dans la demi-boucle 1
m	nombre de liaisons dans la demi-boucle 2
ref	référentiel d'accumulation d'erreurs
M	point de coupure où nous réalisons l'accumulation des erreurs
θ	angle d'entrée du générateur
ϕ	angle de sortie du générateur
γ	angle intermédiaire du générateur
$\Delta\theta$	erreur sur l'angle d'entrée du générateur
$\Delta\phi$	erreur sur l'angle de sortie du générateur
$\Delta\gamma$	angle intermédiaire du générateur
R_1	longueur de la membrure 1 du générateur
R_2	longueur de la membrure 2 du générateur

$\Delta\gamma$	angle intermédiaire du générateur
R_1	longueur de la membrure 1 du générateur
R_2	longueur de la membrure 2 du générateur
R_3	longueur de la membrure 3 du générateur
R_4	longueur de la membrure 4 du générateur
ΔR_1	tolérance dimensionnelle de la longueur de la membrure 1 du générateur
ΔR_2	tolérance dimensionnelle de la longueur de la membrure 2 du générateur
ΔR_3	tolérance dimensionnelle de la longueur de la membrure 3 du générateur
ΔR_4	tolérance dimensionnelle de la longueur de la membrure 4 du générateur
Transport	opérateur de transport
CR	opérateur de changement de référentiel
ΔV_I	vecteur total des résultantes d'erreurs (demi-boucle I)
ΔV_{II}	vecteur total des résultantes d'erreurs (demi-boucle II)
v_x	vecteur d'erreur local ($x = [1...7]$)

LISTE DES ANNEXES

	page
ANNEXE I	
Dessins du variateur mécanique de vitesse	106
ANNEXE II	
Programme du calculateur symbolique	111
ANNEXE III	
Torseurs employés lors de l'analyse du variateur mécanique de vitesse	134

INTRODUCTION

L'accumulation et la propagation des erreurs dans les mécanismes ont d'importantes conséquences sur la qualité de fonctionnement des produits. En effet, les erreurs mécaniques affectent grandement la précision des machines et compliquent les étapes d'assemblage et de fabrication. Il est donc normal que les chercheurs et les industriels tentent d'améliorer la compréhension de ces phénomènes. Il semble d'ailleurs que l'établissement de techniques permettant de prévoir et de contrôler les accumulations de tolérances au cours des étapes de la conception est devenu un véritable objectif stratégique pour les entreprises manufacturières et les fabricants de systèmes de CAO. Nous pouvons aisément donner quelques exemples des avantages d'un contrôle rigoureux des erreurs mécaniques. Un concepteur peut désirer ajuster les tolérances dimensionnelles et géométriques des composantes d'un mécanisme, de manière à optimiser son fonctionnement tout en respectant les conditions fonctionnelles. Un concepteur peut également chercher à répartir les tolérances en tenant compte du coût de fabrication de chacune des composantes. Un concepteur peut chercher à améliorer la précision d'un assemblage en réduisant les dispersions des composantes les plus critiques pour la fonctionnalité du système mécanique.

Les recherches bibliographiques réalisées n'ont pas permis d'identifier une méthode d'analyse convenant aux besoins de notre étude. En fait, seule la méthode différentielle fréquemment employée dans le domaine de la robotique se rapproche de nos attentes. Toutefois, plusieurs auteurs ont traité, de façon partielle, du tolérancement tel que présenté dans la bibliographie.

Les travaux de recherche réalisés ont conduit à la création d'une nouvelle méthode d'analyse des tolérances basée sur les torseurs. Sa validation a été accomplie à l'aide du générateur d'équations à quatre membrures. L'intérêt porté sur ce modèle résulte dans le nombre important de publications dont ce mécanisme a fait l'objet dans

le cadre de l'approche par calcul différentiel. Ainsi, une comparaison entre les deux méthodes a été conduite avec une précision acceptable.

L'analyse d'un modèle simple tel que celui du générateur d'équations à quatre membrures ne permet pas de démontrer l'efficacité pratique de la méthode vectorielle d'analyse. En effet, ce modèle ne contient que des tolérances dimensionnelles dans la direction des membrures. Pour cette raison, un modèle mécanique d'ingénierie est étudié afin d'introduire les aspects de tolérancement géométrique et de jeux. Ces concepts sont fondamentaux dans tous les assemblages de composantes mécaniques.

Un outil informatique permettant de réaliser efficacement les calculs vectoriels associés à notre méthodologie est développé. Cet outil permet de valider rapidement l'approche mathématique proposée et d'analyser le cas du variateur mécanique de vitesse.

Le premier chapitre discute de la problématique associée à l'étude de l'accumulation des erreurs dans les systèmes mécaniques. Le chapitre deux présente une revue bibliographique. Nous donnons les grandes lignes des travaux réalisés dans le domaine de l'analyse des erreurs effectués au cours des deux dernières décennies. Le chapitre trois présente les fondements de la méthodologie vectorielle d'analyse. Les concepts mathématiques proposés sont à la base de la méthode de modélisation et d'accumulation des erreurs. Le chapitre quatre présente le modèle mathématique d'accumulation des dispersions et des jeux mécaniques. Nous décrivons comment le torseur peut définir et cumuler les tolérances dimensionnelles, les tolérances géométriques et les jeux d'un système mécanique. Le cinquième chapitre donne la démarche et les résultats obtenus lors de la validation de la méthode d'analyse des erreurs mécaniques. Ce chapitre présente également l'ensemble des développements mathématiques permettant de résoudre symboliquement un générateur d'équations à quatre membrures. Le chapitre six illustre la procédure et les résultats de l'analyse d'un

cas pratique: le variateur mécanique de vitesse. Ce modèle est partiellement analysé sur un cycle fonctionnel de façon à vérifier la contribution des erreurs dimensionnelles, des erreurs géométriques et des jeux au niveau du "backlash" dans les dents des engrenages coniques. Le chapitre sept explique quelques généralités relatives au fonctionnement du logiciel qui permet de manipuler symboliquement les torseurs et les vecteurs. Cet outil nous permet d'accélérer l'analyse tout en démontrant la possibilité d'en informatiser certaines opérations. Finalement, dans la conclusion nous discutons de l'intérêt du nouvel outil mathématique d'analyse des erreurs mécaniques et des jeux en présentant les développements futurs envisagés.

1.0 PROBLÉMATIQUE

Plusieurs travaux de recherche ont été réalisés dans le domaine de l'analyse des dispersions mécaniques depuis les années soixante [GAR62,VEI86,DAI94]. Malheureusement, les outils mathématiques employés depuis n'ont pas véritablement évolués. Les techniques d'analyse d'erreurs sont encore fondées sur le calcul différentiel pour répartir les poids relatifs des composantes des systèmes mécaniques [FEN89,CLE93,FU88,VEI86]. Le calcul différentiel est certainement un outil puissant pour mesurer la contribution à l'erreur totale des variables critiques d'expressions analytiques quelconques. En effet, une telle approche est avantageuse lorsque nous connaissons la fonction mathématique liant ce que nous connaissons à ce que nous cherchons. Il est toutefois difficile d'établir une telle fonction lorsque le mécanisme atteint un certain niveau de complexité. Il faut mettre en évidence les variables à analyser avant d'appliquer les techniques de différentiation, ce qui n'est pas toujours une tâche réalisable.

L'informatique permet de faciliter le travail des ingénieurs dans un grand nombre de secteurs d'activités. Il est donc intéressant d'envisager l'automatisation de l'analyse des tolérances et des jeux. Compte tenu des difficultés engendrées par le développement des fonctions mathématiques nécessaires à la résolution par calcul différentiel, nous ne pouvons imaginer l'emploi de l'analyse différentielle des erreurs dans l'élaboration d'algorithmes informatiques traitant les dispersions mécaniques. En fait, il semble impossible de systématiser la procédure d'analyse différentielle de façon à envisager son utilisation par les ordinateurs.

Une autre limitation importante de l'analyse des dispersions mécaniques par le calcul différentiel est sans nul doute l'impossibilité de réaliser une étude simultanée des tolérances dimensionnelles, des tolérances géométriques et des jeux. Aucun auteur semble avoir réussi à fondre ces trois éléments importants dans une seule analyse basée

sur le calcul différentiel.

Afin de lever les limitations de l'approche différentielle, une nouvelle approche fondée sur le concept de torseur de petits déplacements est proposée. Les torseurs permettent de représenter vectoriellement les erreurs mécaniques dans un espace à trois dimensions. De plus, ils permettent d'éliminer entièrement les calculs différentiels en introduisant une pondération vectorielle des composantes de l'erreur mécanique totale.

Pour comprendre la méthode qui est proposée et pour démontrer la complexité de l'analyse, nous allons définir certains concepts fondamentaux et obligatoires à la compréhension des travaux de ce mémoire.

Tout mécanisme possède une ou plusieurs boucles fermées constituées de liaisons et de composantes telles qu'illustrées ci-dessous (Figure 1.1). Nous pouvons concevoir une situation nominale et une situation réelle. La situation nominale existe lorsque toutes les composantes sont à leurs dimensions nominales excluant de cette façon, tout défaut géométrique. Par opposition, la situation réelle doit inclure les erreurs sur les composantes qui représentent évidemment, des déviations à la situation nominale. Ces erreurs regroupent les tolérances dimensionnelles et les tolérances géométriques sur chacune des composantes mécaniques.

Un mécanisme est constitué d'un ensemble de composants en relation les uns avec les autres. Son assemblage et son fonctionnement en situation réelle implique inévitablement, l'introduction de jeux et la prise en compte d'efforts et de déformations mécaniques pour analyser les incertitudes introduites par les erreurs. Nous devons tenir compte, de plus, d'un minimum de critères pour assurer le fonctionnement du mécanisme. Ces critères sont représentés par les conditions fonctionnelles qui se définissent comme l'ensemble des conditions et des contraintes, qui assurent une réponse adéquate du mécanisme à une ou plusieurs fonctions. Les jeux se retrouvent concentrés

dans les liaisons entre les composants. Les efforts qui agissent sur le mécanisme affectent inévitablement les composants et définissent de quelle façon les jeux sont repris par ce dernier. Enfin, les déformations représentent un élément important qui doit être considéré dans l'analyse de tout mécanisme. Il existe deux types de déformation. Premièrement, les déformations surfaciques dans les liaisons, qui sont engendrées par le contact localisé entre les surfaces. Deuxièmement, les déformations volumiques dans les composants, qui sont générées par les efforts que celles-ci doivent subir compte tenu des forces internes ou externes au mécanisme.

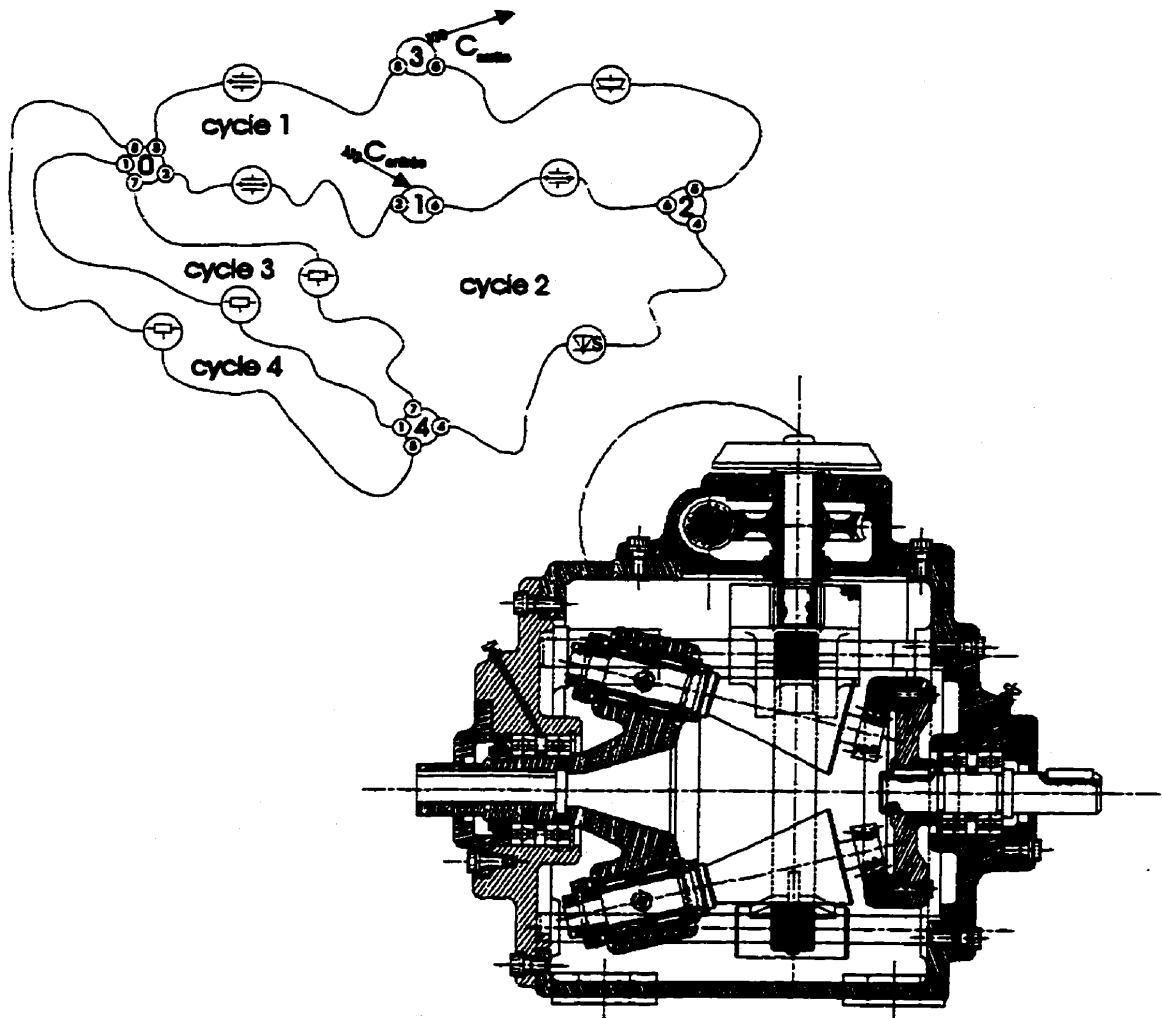


Figure 1.1 Boucles d'un mécanisme

Considérant l'ensemble des facteurs agissant sur les mécanismes, il s'agit maintenant d'établir une équation liant l'ensemble de ces phénomènes et définissant leurs influences relatives. Pour établir cette équation nous avons d'abord défini certaines hypothèses:

1. nous considérons que les mécanismes étudiés sont constitués d'un ensemble de pièces et de liaisons mécaniques formant une ou plusieurs boucles fermées;
2. nous considérons les déformations volumiques dues à des efforts comme absentes ou négligeables;
3. nous considérons les déformations surfaciques dues à des efforts comme absentes ou négligeables;
4. nous considérons les composantes mécaniques comme des corps indéformables (liées aux hypothèses 2 et 3);
5. nous considérons les composantes mécaniques comme étant en situation réelle et possédant des défauts géométriques (tolérances dimensionnelles et tolérances géométriques);
6. nous proposons que les liaisons possèdent des jeux;
7. nous définissons une condition fonctionnelle comme un espace contrôlé entre deux composantes;
8. nous considérons que la forme d'une surface est parfaite.

En résumé, nous cherchons à mettre au point une méthodologie d'analyse vectorielle des défauts géométriques et des jeux de mécanismes complexes employant des outils mathématiques compatibles les uns aux autres. Cette démarche permet de préparer la voie à des travaux menant à une informatisation efficiente de l'analyse de la propagation des erreurs mécaniques et des jeux.

2.0 REVUE BIBLIOGRAPHIQUE

Les travaux réalisés dans notre champ de recherche ont surtout porté sur l'étude de la cinématique, lors de perturbations engendrées par des erreurs mécaniques. Dès les années 50, Denavit et Hartenberg établissaient une notation pour l'analyse cinématique des liaisons en employant quatre paramètres géométriques [DEN75]. Cette notation a été employée par ces mêmes auteurs lors d'analyses d'erreurs de liaisons mécaniques telles que les joints Hooke [AUS65]. Des recherches similaires ont été réalisées par plusieurs chercheurs [DHA78,VEI86,DAI94]. Les matrices homogènes de transformations géométriques (matrices 4×4) sont utilisées conjointement avec la notation Denavit-Hartenberg pour l'analyse différentielle des erreurs [GAR62,VEI86,DAI94]. Évidemment, les travaux consultés n'ont pas porté exclusivement sur l'analyse des erreurs. Par exemple, Denavit et Hartenberg traitent de l'effet des erreurs sur la cinématique des mécanismes. Due aux limitations naturelles de la méthode d'analyse, seules les erreurs dimensionnelles et les jeux ont fait l'objet d'étude.

Garrett et Hall quant à eux, se sont intéressés à l'empilement probabilistique des erreurs dans les mécanismes [GAR62,GAR69]. Ces chercheurs ont également étudié, tout comme Denavit et Hartenberg l'avaient fait, les problèmes cinématiques causés par les erreurs dimensionnelles. Dhande et Chakraborty proposent une technique d'analyse probabilistique par linéarisation permettant d'établir le poids relatif à l'erreur totale de chacun des éléments d'un générateur d'équations à quatre membrures. Ils ont cherché à optimiser les membrures des mécanismes de manière à respecter une erreur mécanique maximale. Ces mêmes auteurs ont étudié le comportement des erreurs aux joints mécaniques [DHA73,DHA78]. Veitschegger et Wu ont validé l'analyse par linéarisation des équations différentielles d'erreur en vérifiant la contribution réelle des termes de degrés supérieurs [VEI86]. Tout comme leurs prédécesseurs, ils ont employé les méthodes développées par Denavit et Hartenberg ainsi que la notation associée [VEI86].

Un élément commun unit les travaux de tous ces chercheurs: l'utilisation du calcul différentiel pour l'évaluation des erreurs mécaniques.

Plus récemment, Lee et Yi [LEE95,LEE94] ont étudié l'*assemblabilité* des systèmes mécaniques. Ils ont compris l'importance du contrôle des erreurs et des jeux lors de la conception et de l'assemblage de produits tridimensionnels. En effet, ils illustrent les limitations inhérentes des méthodes traditionnelles d'empilements unidimensionnels de cotes ("Stack-up"). Ces chercheurs proposent donc une nouvelle méthode d'analyse des défauts géométriques et des jeux. Ils abordent le tolérancement géométrique de positionnement en supposant un désalignement des référentiels d'analyse par rapport à une situation nominale. Ils proposent également de modéliser les autres catégories de tolérances géométriques par une représentation matricielle de type: tolérance dimensionnelle. Ils discutent brièvement de l'applicabilité de leur nouvelle méthode d'analyse à des chaînes de tolérances et de jeux à boucles multiples. Whitney et Gilbert [WHI93] proposent quant à eux de modéliser les erreurs d'une façon statistique grâce au modèle de l'ellipsoïde probabilistique. L'un des apports importants de leurs travaux est sans nul doute, la prise de conscience de l'aspect essentiel des jeux dans les mécanismes. Ils comprennent la relation fondamentale qui existent entre les tolérances et les jeux dans les assemblages.

Whitney et Gilbert [WHI93,GIL92] traitent de la représentation et de l'analyse des erreurs géométriques grâce à un ellipsoïde d'approximation probabilistique. Ils proposent un modèle d'accumulation probabilistique des tolérances dimensionnelles et de certaines tolérances géométriques de manière à vérifier l'*assemblabilité* des mécanismes tridimensionnels. Ce modèle d'accumulation est basé sur un algorithme "Closed-Form" recherchant les paramètres définissant la position statistique et l'orientation statistique des composantes d'un assemblage. Les dispersions sont générées grâce à des algorithmes de type Monte Carlo. Tout comme les travaux présentés dans ce mémoire, Whitney et Gilbert utilisent un outil mathématique définissant les erreurs

mécaniques comme de petits déplacements possédant 6 degrés de liberté, soient les matrices homogènes de transformation géométrique. De plus, leur modèle d'accumulation demande la mise en place de référentiels pour chacune des erreurs tel que nous le proposons. Whitney et Gilbert ne tiennent toutefois pas compte des jeux et des conditions fonctionnelles qui sont évidemment essentiels à tout mécanisme.

Les travaux de Whitney, Gilbert, Lee et Yi constituent des avancements majeurs dans l'étude des erreurs des systèmes mécaniques tridimensionnels. Ils ont apporté plusieurs nouveaux concepts mathématiques à l'analyse statistique de l'*assemblabilité* des mécanismes. Malheureusement, la réussite des assemblages n'assure d'aucune façon le bon fonctionnement des produits mécaniques. En effet, il existe d'autres paramètres qui doivent être considérés tels que les conditions fonctionnelles et l'orientation des forces internes dans les mécanismes.

Le tolérancement vectoriel a été abordé par quelques chercheurs. L'équipe de Gaunet [GAU93] propose une formulation vectorielle pour décrire le tolérancement dimensionnel et le tolérancement géométrique. Ils définissent le concept de torseurs de tolérancement d'une manière se rapprochant des modèles que nous proposons. Dans leur modèle, chacune des tolérances de dimensions nominales se définit par deux torseurs de petits déplacements et un système de contraintes. Malheureusement, le modèle d'analyse de Gaunet est limité à la vérification de l'*assemblabilité* des mécanisme. Il ne permet pas de vérifier l'effet des défauts géométriques et des jeux lors du fonctionnement des mécanismes.

Bjørke a traité de l'informatisation du contrôle du tolérancement [BJO89]. Dans son ouvrage, il effectue un tour d'horizon d'un grand nombre de techniques, de méthodes et d'outils associés au tolérancement mécanique. Bjørke propose divers modèles statistiques ou modèles analytiques de représentation des tolérances. Il présente également divers algorithmes de résolution des chaînes de cotes. Il propose

des modèles d'accumulation d'erreurs pour les mécanismes à multi-cycles fonctionnels. Malheureusement, Bjørke ne propose pas de méthode intégrée associant toutes les catégories de tolérances et de jeux.

L'étude des erreurs mécaniques en utilisant les dérivées partielles, exprimées sous forme de matrices de transformations jacobienes, sont certainement l'une des constances retrouvées dans plusieurs des travaux des chercheurs étudiés. Fenton, Cleghorn et Fu n'ont pas dévié de cette tendance. En effet, ils emploient les différentielles pour introduire les erreurs dimensionnelles associées à chacune des membrures des générateurs d'équations étudiées [CLE93,FU88]. Ils ont également abordé les systèmes mécaniques à boucles multiples et l'analyse par indice de sensibilité [FEN89,CLE93,FU88]. Ce mémoire se propose d'employer les résultats et les modèles développés par ces auteurs pour valider le nouvel outil mathématique d'analyse qui y est présenté.

3.0 FONDEMENTS MATHÉMATIQUES DE LA MÉTHODOLOGIE D'ANALYSE

Ce chapitre présente les fondements mathématiques de la méthodologie définie pour l'analyse vectorielle des erreurs mécaniques et des jeux.

Nous nous devons de définir certains concepts préliminaires à la compréhension des développements mathématiques présentés dans ce chapitre. Les jeux définis dans les liaisons entre les composantes et les défauts géométriques définis sur chacune de ces composantes possèdent des caractères essentiellement locaux. En effet, les jeux et les tolérances ont une portée locale sur le mécanisme en affectant par exemple, certaines dimensions et certains espacements. Par contre, le fonctionnement du mécanisme est déterminé par l'interrelation de l'ensemble des composantes incluant, évidemment, les défauts géométriques et les jeux qui y sont associés. Pour cette raison, nous disons que le mécanisme possède un caractère essentiellement global. Pour réaliser le processus d'accumulation des défauts géométriques et des jeux, nous devons définir un pont unissant ces deux caractères opposés. De ce fait, les modèles qui sont développés dans ce chapitre ont pour but de répondre aux deux questions suivantes: Comment modéliser les jeux dans les liaisons et les défauts géométriques au sein des pièces sachant que ceux-ci possèdent un caractère essentiellement local? Comment déterminer l'incidence de ces défauts en un point donné du mécanisme et dans une direction donnée sachant que celui-ci a un caractère essentiellement global? Notons que les modèles mathématiques utilisés dans ce chapitre sont tous définis à partir de la théorie des déplacements des corps solides.

Nous présentons à la section 3.1 les notions de transformations isométriques essentielles à la compréhension de la méthodologie d'analyse. Nous concluons à la section 3.2, en définissant les modèles mathématiques utilisés pour décrire les tolérances et les jeux.

3.1 Fondements mathématiques

Nous résumons à la sous-section 3.1.1, la théorie des grands déplacements de laquelle dérive la théorie des torseurs de petits déplacements présentée à la sous-section 3.1.2. Ces deux théories permettent d'obtenir une intégration complète de la méthodologie d'analyse vectorielle et de réconcilier les caractères locaux et globaux.

3.1.1 Déplacement d'un point et champ de déplacement

La procédure mise au point pour calculer la contribution de chacune des erreurs d'un système mécanique quelconque est basée sur la théorie des déplacements vectoriels. Nous nous devons de bien définir ce que nous entendons par déplacement. En effet, toutes les notions mathématiques, constituant le coeur de la procédure d'analyse proposée, sont fondées sur ce concept.

Un déplacement se définit comme une transformation isométrique dont la matrice orthogonale est directe [HER75]. Nous savons de plus, que l'ensemble des déplacements dans un espace à trois dimensions constitue un groupe continu fini d'ordre six [DUP91,HER75]. Nous pouvons définir tout déplacement d'un solide (S) dans un référentiel orthonormé R_0 par l'expression suivante:

$$\mathbf{m}_{\text{new}} = \mathbf{t} + [\mathbf{A}] \cdot \mathbf{m}_{\text{old}}$$

où,

- \mathbf{m}_{old} : vecteur associé au point $\mathbf{m}_{\text{old}} \in (S)$, exprimé dans R_0 ;
- \mathbf{m}_{new} : vecteur associé au point $\mathbf{m}_{\text{new}} \in (S)$, homologue du point \mathbf{m}_{old} , après transformation, exprimé dans R_0 ;
- \mathbf{t} : vecteur de translation du solide (S) exprimé dans R_0 ;
- $[\mathbf{A}]$: matrice régulière orthogonale directe exprimant une rotation du solide (S) autour d'un point ou d'un axe.

Cette équation traduit le fait que tout mouvement d'un corps solide (S) dans un repère (Ro) peut être obtenu par vissage, c'est-à-dire par une combinaison d'une translation et d'une rotation telle qu'illustrée à la figure 3.1.

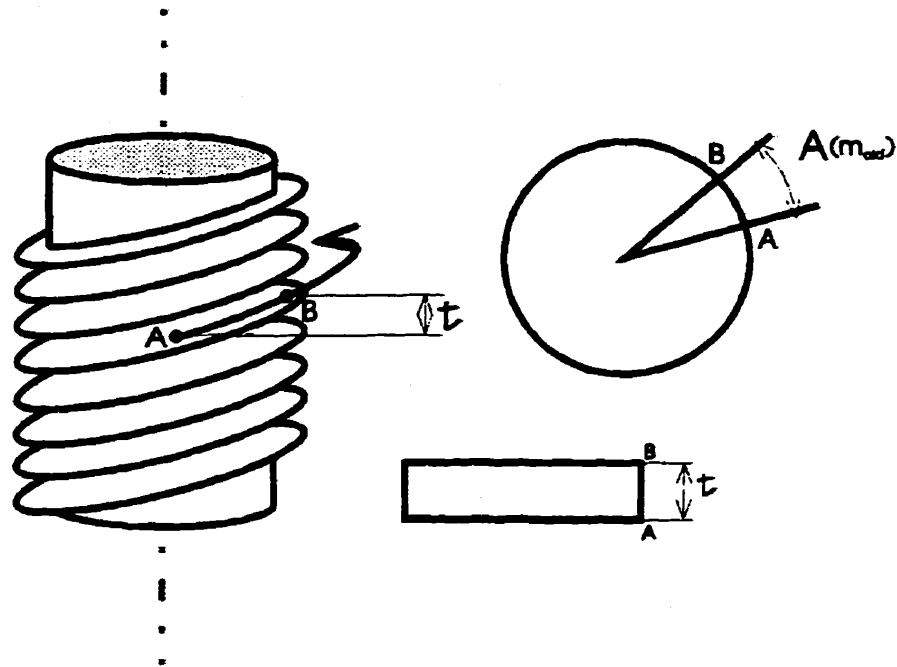


Figure 3.1 Modèle du mouvement infinitésimal d'un corps

Jacques-Marie Hervé [HER75] introduit la notation exponentielle vectorielle dérivée de l'algèbre de Lie des déplacements pour caractériser une rotation. Ainsi, le déplacement du point $m_{old} \in (S)$ qui l'amène en m_{new} homologue de m_{old} , est:

$$m_{new} = t + \exp(\alpha u \times) \cdot m_{old}$$

où,

- $\exp()$: opérateur exponentiel vectoriel;
- m_{old} : vecteur associé au point $m_{old} \in (S)$, exprimé dans Ro ;
- m_{new} : vecteur associé au point $m_{new} \in (S)$, homologue du point m_{old} , après transformation, exprimé dans Ro ;
- α : angle rotation autour de l'axe du vecteur unitaire u ;

\mathbf{t} : vecteur de translation du solide (S) exprimé dans R_0 ;
 \times : symbole du produit vectoriel.

La forme matricielle se définit de la manière suivante:

$$\begin{pmatrix} m_{new} \\ 1 \end{pmatrix} = \begin{pmatrix} \exp(\alpha \mathbf{u} \times) & \mathbf{t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} m_{old} \\ 1 \end{pmatrix}$$

Sous cette forme, il est possible d'appliquer un déplacement à tout vecteur \mathbf{m} ou à toute entité vectorielle qu'il s'agisse de points, de droites ou de plans [DUP91]. Nous notons enfin qu'un changement de référentiel est interprété comme un déplacement du fait de leur équivalence [DUP91].

Compte tenu que nous voulons définir les opérations mathématiques d'une manière plus explicite, nous avons redéfini la notation de certains opérateurs mathématiques. Notons également que ces nouvelles notations sont employées par l'outil informatique présenté dans ce mémoire. Définissons d'abord une rotation d'angle α autour de l'axe du vecteur unitaire \mathbf{u} de la façon suivante:

$$R(\alpha, \mathbf{u}) = \exp(\alpha \mathbf{u} \times)$$

et, une translation de longueur β selon l'axe du vecteur unitaire \mathbf{u} :

$$T(\beta, \mathbf{u}) = \mathbf{t} = \beta \mathbf{u}$$

Les propriétés mathématiques de l'exponentielle vectorielle les plus utilisées lors de la compilation des dispersions mécaniques sont présentées ci-dessous.

Linéarité:

$$1. \exp(\mathbf{A} \mathbf{u} \times)(\mathbf{B} \mathbf{v} + \mathbf{G} \mathbf{w}) = \mathbf{B} \cdot \exp(\mathbf{A} \mathbf{u} \times) \cdot \mathbf{v} + \mathbf{G} \cdot \exp(\mathbf{A} \mathbf{u} \times) \cdot \mathbf{w}$$

$$2. \exp(A \mathbf{u} \times) \cdot \mathbf{u} = \mathbf{u}$$

$$3. \exp(A1 \mathbf{u} \times) \cdot \exp(A2 \mathbf{u} \times) = \exp((A1 + A2) \mathbf{u} \times).$$

$$4. \exp(-\pi/2 \mathbf{u} \times) = \mathbf{u} \times$$

Développement en série de l'opérateur exponentiel:

$$\exp(A \mathbf{u} \times) = 1 + (1/1!)A\mathbf{u} \times + (1/2!)A^2\mathbf{u} \times (\mathbf{u} \times) + \dots + (1/n!)A^n(\mathbf{u} \times)^n$$

Expression de l'opérateur "exp":

$$\exp(A\mathbf{u} \times) \cdot \mathbf{v} = \mathbf{v} + \sin(A) \cdot (\mathbf{u} \times \mathbf{v}) + (1 - \cos(A)) \cdot ((\mathbf{u} \times) \mathbf{u} \times \mathbf{v})$$

Cette expression devient dans un repère orthonormé direct $(\mathbf{u}, \mathbf{v}, \mathbf{w})$:

$$\exp(A\mathbf{u} \times) \cdot \mathbf{v} = \sin(A) \cdot (\mathbf{u} \times \mathbf{v}) + \cos(A) \cdot \mathbf{v} = \sin(A) \cdot \mathbf{w} + \cos(A) \cdot \mathbf{v}$$

et pour un repère quelconque, elle devient:

$$\exp(A\mathbf{u} \times) \cdot \mathbf{v} = (\mathbf{u} \cdot \mathbf{v})\mathbf{u} + \sin(A)[\mathbf{u} \times (\mathbf{v} - (\mathbf{u} \cdot \mathbf{v})\mathbf{u})] + \cos(A)[\mathbf{v} - (\mathbf{u} \cdot \mathbf{v})\mathbf{u}]$$

D'autres propriétés intéressantes sont présentées dans les publications de Jacques-Marie Hervé [HER75].

Remarquons que l'exponentielle vectorielle ne représente aucunement une approximation. Une dérivation par rapport au temps de l'expression exponentielle d'un déplacement quelconque conduit directement à l'équation cinématique correspondante (voir section 3.1.2) [DUP91].

3.1.2 Les torseurs de petits déplacements

Le second concept mathématique d'importance employé dans ce travail est la théorie des torseurs. Soit le solide S_0 soumis à une translation de vecteur \vec{t} et à une rotation infinitésimale d'amplitude β autour de l'axe de vecteur unitaire \vec{u} centré sur m_1 ($R(\beta, \vec{u})$) (voir Figure 3.2).

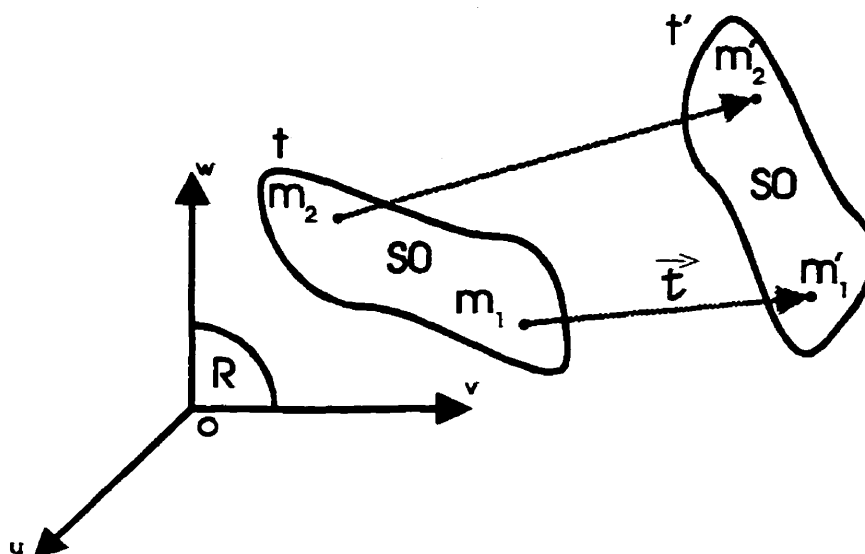


Figure 3.2 Illustration d'un corps se déplaçant durant une période $d\tau$

Le déplacement de (S_0) amène le point $m_1 \in (S_0)$ au point m_1' :

$$\vec{m}_1' = \vec{m}_1 + \vec{t}$$

où,

\vec{m}_1' : vecteur (\vec{Om}_1') identifiant le point m_1' ;

\vec{m}_1 : vecteur (\vec{Om}_1) identifiant le point m_1 ;

\vec{t} : vecteur déplacement définissant la translation du point m_1 au point m_1' .

Ce même déplacement amène le point $m_2 \in (S_0)$ au point m_2' tel que:

$$m_2' = m_1' + \exp(\beta u \times)(m_1, m_2).$$

où,

m_2' : vecteur (om_2') identifiant le point m_2' ;

(m_1, m_2) : vecteur constant liant le point m_1 au point m_2 .

Pour des rotations infinitésimales, nous pourrions développer l'exponentielle vectorielle $(\exp(\dots))$ à l'ordre 1. Ainsi,

$$m_2' = m_1' + (m_1, m_2) + \beta u \times (m_1, m_2),$$

or,

$$m_2 = m_1 + (m_1, m_2),$$

$$m_2' = m_1 + (m_1, m_2) + t + \beta u \times (m_1, m_2),$$

$$m_2' = m_2 + t + \beta u \times (m_1, m_2),$$

ainsi, les déplacements aux points m_1 et m_2 s'expriment de la façon suivante:

$$\text{point } m_1: (m_1, m_1') = m_1' - m_1 = t = D(m_1),$$

$$\text{point } m_2: (m_2, m_2') = m_2' - m_2 = t + \beta u \times (m_1, m_2) = D(m_2).$$

D'où,

$$D(m_2) = D(m_1) + \beta u \times (m_1, m_2).$$

Nous pouvons donc caractériser le champ de déplacement en tout point m du solide $(S_0)/R_0$ à l'aide de l'opérateur $D(m)$, $m \in (S_0)$. Cet opérateur est tel que si au point m_1 $D(m_1)$ est connu et si βu représente le vecteur de rotation infinitésimale alors au point m_2 $D(m_2) = D(m_1) + \beta u \times (m_1, m_2)$ (théorème des moments), on appellera

Torseur des petits déplacements au point m_2 le couple $\{D(m_2), \beta u\}$. Au point m_1 , le torseur s'exprime par $\{D(m_1) = t, \beta u\}$, le terme βu représente la résultante du torseur et est invariant par rapport à tout point de l'espace. $D(m_1)$ est le moment du torseur.

Un torseur de petits déplacements représente donc la résultante et le moment, en un point donné, d'un champ vectoriel traduisant le déplacement d'un corps (S_0) et respectant le théorème des moments. Chacun de ces vecteurs peut s'exprimer dans un espace à trois dimensions orthonormé direct dont les vecteurs unitaires sont illustrés par les symboles u , v et w (par convention). Nous pouvons donc représenter aisément les six degrés de liberté exprimant les possibilités de petits déplacements d'un solide S_1 auquel est attaché le référentiel R_1 par rapport à un solide S_0 auquel est attaché le référentiel R_0 . Les trois translations s'expriment par le vecteur moment du torseur calculé au centre de rotation et les trois rotations s'expriment par le vecteur résultant.

Selon les hypothèses sous jacentes au concept du torseur de petits déplacements, tout mouvement d'amplitude infinitésimal d'un corps solide indéformable peut se représenter par un "vissage" infinitésimal. Compte tenu de cette approximation, il est clair que nous devons nous soumettre à quelques limitations. La méthode proposée ne peut s'appliquer en effet qu'à des rotations infinitésimales dont l'angle n'excède pas 5 degrés [DUP91]. Au-delà de cette limite, l'erreur sur l'approximation devient trop élevée.

Notre intention est de mettre au point une procédure d'analyse pour les systèmes mécaniques dont les composantes sont soumises à de faibles variations dimensionnelles autour d'une position nominale ou moyenne. La méthode vectorielle d'analyse des erreurs mécaniques et des jeux perd de son efficacité pour les produits où les dispersions dimensionnelles et géométriques sont importantes. En fait, l'emploi de cette procédure d'analyse n'a d'intérêt que pour les produits dont les tolérances et les jeux sont critiques pour la fonctionnalité [DUP91]. Nous verrons au chapitre 5.0 "Validation de

la méthodologie" (section 5.2) que les erreurs générées par l'approche différentielle et par la méthode de l'exponentielle vectorielle, conduisent à des déviations similaires par rapport à l'erreur mécanique calculée analytiquement.

Afin de simplifier les développements mathématiques qui suivent, nous allons établir une convention dans la notation des torseurs. Nous définissons le torseur $\{T\}$ exprimé dans le référentiel ref et appliqué au point 0 de la façon suivante:

$$\{T\}_{ref/0} = \{ \overline{M}, \overline{R} \}_{ref/0}$$

et matriciellement,

$$\{T\}_{ref/0} = \begin{bmatrix} x & \alpha \\ y & \beta \\ z & \gamma \end{bmatrix}_{ref/0}$$

Posons un torseur $\{T\}$ composé d'un champ constant (résultante du torseur) $R(T)$ et d'un champ de moments $M(m,T)$ tel que $M(b,T) = M(a,T) + R(T) \times (ab)$. Nous dirons que $R(T)$ et $M(m,T)$ sont les éléments de réduction au point m du torseur $\{T\}$ défini dans le référentiel ref [AGA86].

Les **propriétés** mathématiques les plus importantes des torseurs sont les suivantes [AGA86,SPE93]:

- un torseur est complètement défini par ses éléments de réduction en un point quelconque de l'espace;
- la résultante est un invariant du torseur c'est-à-dire, que sa direction, son sens et sa grandeur ne varieront pas quelque soit le point choisi dans l'espace;

- l'automoment $\mathbf{R}(T) \cdot \mathbf{M}(m, T)$ est un invariant scalaire du torseur; Pour tous points m, n : $\mathbf{R}(T) \cdot \mathbf{M}(m, T) = \mathbf{R}(T) \cdot \mathbf{M}(n, T)$
- le champ de moment d'un torseur est équiprojectif. Pour tous points m, n : $\mathbf{M}(m, T) \cdot (\mathbf{mn}) = \mathbf{M}(n, T) \cdot (\mathbf{mn})$

Les principales **opérations** sur les torseurs sont [AGA86]:

Égalité de deux torseurs.

$$\{T1\}_{ref/m} = \{T2\}_{ref/m} \text{ où}$$

$$\mathbf{R}(T1) = \mathbf{R}(T2)$$

$$\mathbf{M}(m, T1) = \mathbf{M}(m, T2)$$

Somme de deux torseurs.

$$\{T\}_{ref/m} = \{T1\}_{ref/m} + \{T2\}_{ref/m} \text{ où}$$

$$\mathbf{R}(T) = \mathbf{R}(T1) + \mathbf{R}(T2)$$

$$\mathbf{M}(m, T) = \mathbf{M}(m, T1) + \mathbf{M}(m, T2)$$

Produit d'un torseur par un scalaire.

$$\{T\}_{ref/m} = X \cdot \{T1\}_{ref/m} \text{ où}$$

$$\mathbf{R}(T) = X \cdot \mathbf{R}(T1)$$

$$\mathbf{M}(m, T) = X \cdot \mathbf{M}(m, T1)$$

Rappelons que cette dernière opération est:

- associative pour la multiplication des scalaires;
- distributive par rapport à l'addition des scalaires;
- distributive par rapport à l'addition des torseurs.

TRANSPORT.

Le macro-opérateur "TRANSPORT" constitue le calcul de l'effet d'un torseur en un point b lorsque ce même torseur est défini au point a. Il n'est qu'une reformulation du théorème des moments sous forme d'un opérateur. Voici la représentation symbolique du transport d'un torseur (Figure 3.3).

$$\begin{array}{c} \text{Transport} \\ \{ T \}_{ref/A} \xrightarrow{\quad \bar{d} \quad} \{ T \}_{ref/B} \end{array}$$

où, $\bar{d} = au + bv + cw = (\mathbf{AB})$ exprimé dans ref, peut être obtenu par combinaison de déplacement en translations et en rotations.

$$\begin{bmatrix} x_B & \alpha_B \\ y_B & \beta_B \\ z_B & \gamma_B \end{bmatrix}_{ref/B} = \begin{bmatrix} x_A & \alpha_A & a \\ y_A & \beta_A & b \\ z_A & \gamma_A & c \end{bmatrix} \times \begin{bmatrix} \alpha_A \\ \beta_A \\ \gamma_A \end{bmatrix}_{ref/B}$$

$$\begin{bmatrix} x_B & \alpha_B \\ y_B & \beta_B \\ z_B & \gamma_B \end{bmatrix}_{ref/B} = \begin{bmatrix} x_A + \beta_A c - \gamma_A b & \alpha_A \\ y_A + \gamma_A a - \alpha_A c & \beta_A \\ z_A + \alpha_A b - \beta_A a & \gamma_A \end{bmatrix}_{ref/B}$$

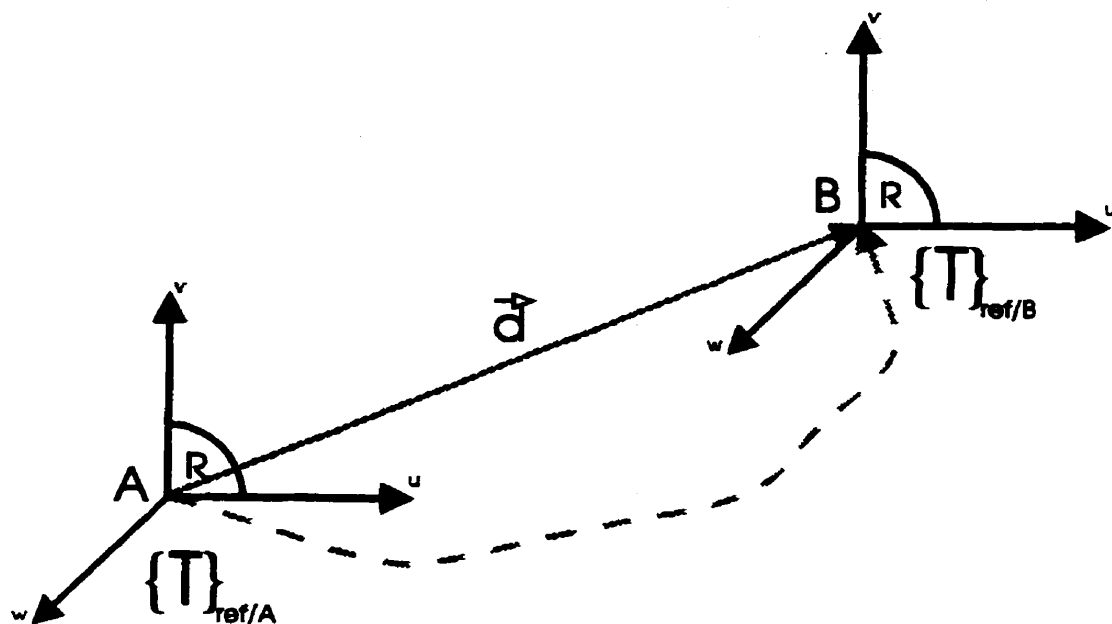


Figure 3.3 Transport du torseur $\{T\}$ du point A au point B

Changement de Référentiel.

Le "Changement de Référentiel" permet comme son nom l'indique, d'exprimer adéquatement le torseur dans un référentiel compatible avec l'expression des erreurs et des jeux. Notons que l'action du changement de référentiel s'effectue par l'action de transformations de type rotation (Figure 3.4).

$$\{T\}_{ref AO} \xrightarrow[\bar{\lambda}]{CR^*} \{T\}_{ref BO}$$

*Changement de Référentiel

Le vecteur λ contient les transformations angulaires exprimées de la façon suivante:

$$\lambda = (\delta u + \zeta v + \eta w).$$

$$(\mathbf{R}') = (\mathbf{T})(\mathbf{R})$$

$$(\mathbf{M}') = (\mathbf{T})(\mathbf{M})$$

où, (\mathbf{T}) : transformation de l'opérateur exponentiel vectoriel;

(\mathbf{R}) : résultante du torseur initial;

(\mathbf{M}) : moment de la résultante du torseur initial;

(\mathbf{R}') : résultante du torseur final;

(\mathbf{M}') : moment de la résultante du torseur final.

ou encore,

$$(\mathbf{R}') = (\exp(\delta \mathbf{u} \times) \cdot \exp(\zeta \mathbf{v} \times) \cdot \exp(\eta \mathbf{w} \times)) \cdot (\mathbf{R})$$

$$(\mathbf{M}') = (\exp(\delta \mathbf{u} \times) \cdot \exp(\zeta \mathbf{v} \times) \cdot \exp(\eta \mathbf{w} \times)) \cdot (\mathbf{M})$$

finalement,

$$(\mathbf{R}') = (\exp(\delta \mathbf{u} \times) \cdot \exp(\zeta \mathbf{v} \times) \cdot \exp(\eta \mathbf{w} \times)) \cdot (X\mathbf{u} + Y\mathbf{v} + Z\mathbf{w})$$

$$(\mathbf{M}') = (\exp(\delta \mathbf{u} \times) \cdot \exp(\zeta \mathbf{v} \times) \cdot \exp(\eta \mathbf{w} \times)) \cdot (\alpha \mathbf{u} + \beta \mathbf{v} + \gamma \mathbf{w})$$

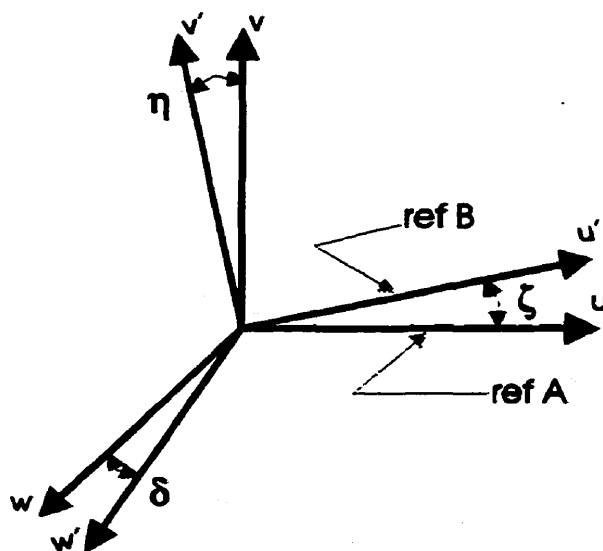


Figure 3.4 Changement de Référentiel du torseur $\{T\}$ du réf. A au réf. B

Nous venons de présenter l'ensemble des propriétés et des opérations qui sont utilisées dans l'élaboration de la procédure d'analyse des erreurs et des jeux de systèmes mécaniques. Nous décrivons dans la section 3.2, comment les tolérances et les jeux doivent être modélisés en utilisant les outils mathématiques proposés.

3.2 Tolérances et Jeux

Toutes les composantes d'un assemblage contiennent des erreurs dimensionnelles et des erreurs géométriques. Pour réaliser l'assemblage de ces composantes, il est habituellement nécessaire d'introduire des jeux. Toutefois, une trop grande abondance de jeux diminue la qualité du fonctionnement de l'assemblage. Les tolérances et les jeux sont intimement liés.

Tolérances

Les tolérances dimensionnelles et les tolérances géométriques représentent les défauts géométriques admissibles sur les composantes mécaniques dans une situation réelle. Ces défauts sont, comme nous l'avons déjà mentionné, des déviations par rapport à la situation nominale de ces mêmes composantes mécaniques. De ce fait, nous pouvons imaginer une tolérance comme une déviation physique du référentiel d'un objet par rapport à sa position nominale ou sa position moyenne. Cette déviation de nature infinitésimale, peut s'exprimer suivant les six degrés de liberté d'un espace à trois dimensions. En effet, la position d'un référentiel dans l'espace peut toujours être définie par trois rotations et trois translations. Cette définition de la tolérance d'un objet montre l'intérêt que nous portons à employer les torseurs de petits déplacements. Cet outil mathématique est utilisé pour analyser de petits déplacements tels que ceux introduits par les tolérances géométriques et par les tolérances dimensionnelles. La figure 3.5 donne une représentation de l'utilisation d'un torseur pour définir une tolérance.

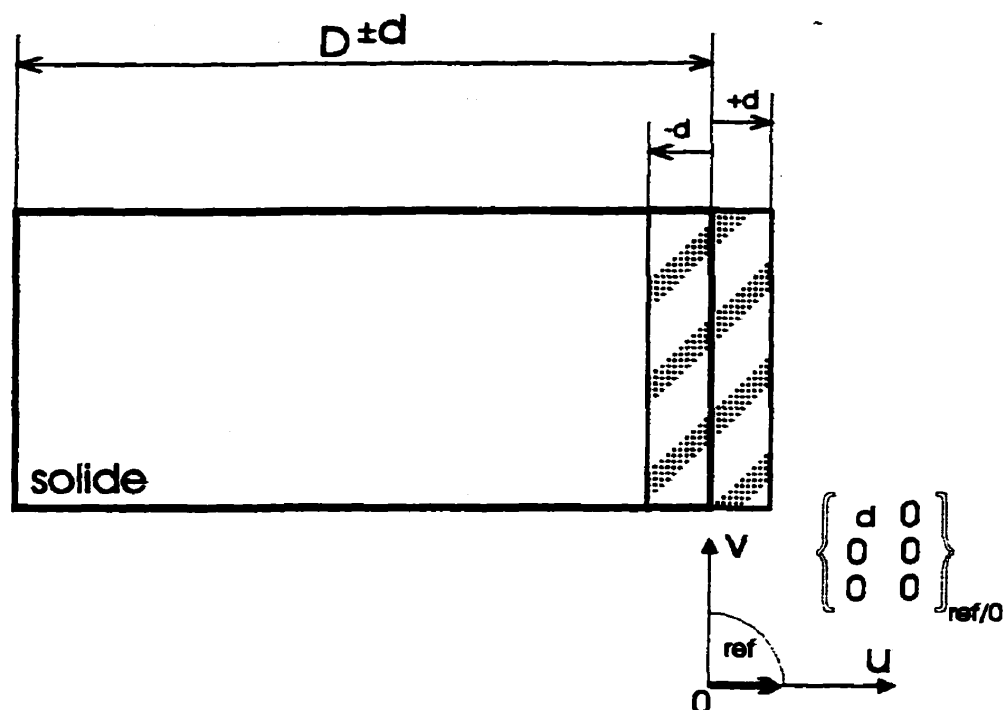


Figure 3.5 Représentation d'une tolérance dimensionnelle par un torseur

Nous savons maintenant que les torseurs de petits déplacements permettent de définir les erreurs sur une composante. Il est habituellement nécessaire d'établir un système de contraintes pour en compléter leur définition. En effet, les torseurs sont des expressions vectorielles générales. Ces contraintes permettent donc de délimiter leur action dans l'espace. Elles permettent également d'exprimer mathématiquement les relations entre chacun des degrés de liberté. Ces relations établissent les limites physiques des erreurs exprimées sur les composantes. Quelques auteurs ont défini des modèles mathématiques pour chacun des groupes de défauts géométriques et chacun des groupes de jeux [CLE94]. Nous nous sommes donc inspirés de leurs travaux pour établir nos systèmes de contraintes.

Nous devons présenter quelques exemples pour illustrer la modélisation des tolérances grâce aux torseurs de petits déplacements. Définissons trois cas standards de

tolérancement soient: une tolérance dimensionnelle, une tolérance d'orientation et une tolérance de position telles qu'illustrées à la figure 3.6.

La tolérance dimensionnelle se définit d'abord par un torseur de petits déplacements positionné sur un point appartenant à l'un des deux plans de référence associé à la dimension nominale (ou moyenne). Le référentiel du torseur est ensuite défini dans un repère où l'axe principal est orienté normalement au plan de référence. Finalement, un vecteur d'erreur dimensionnelle est défini grâce au torseur tel que présenté à la figure 3.6. Comme nous l'avons dit précédemment, certaines contraintes doivent être développées de manière à limiter la plage des erreurs. Ainsi, nous définissons, pour cette tolérance, l'inéquation suivante:

$$\left\{ x_1 \leq X \right.$$

Elle exprime la limite physique que peut adopter la tolérance dimensionnelle.

La tolérance d'orientation illustrée à la figure 3.6 correspond à une tolérance de parallélisme. Elle comporte une composante en translation et deux composantes en rotation. En effet, une telle tolérance est formée de deux plans limites entre lesquels le plan réel peut se trouver. Ce plan réel peut donc être soumis à une combinaison de rotations autour des axes définis par les vecteurs unitaires v et w et de translations sur l'axe défini par le vecteur unitaire u par rapport à sa situation nominale elle même définie par x_1 . Notons qu'il existe un couplage entre ces degrés de libertés. En effet, une translation ne peut être maximale lorsqu'il existe un défaut angulaire. Le couplage de ces degrés de liberté est exprimé par le système d'inéquations présenté ci-dessous. Nous constatons également dans l'exemple, qu'une erreur de parallélisme affecte l'erreur dimensionnelle. Les deux tolérances sont couplées car elles affectent l'erreur associée à la même dimension nominale (ou moyenne). Ce couplage est défini dans le système d'inéquation ci-dessous.

$$\begin{cases}
 x_1 + \frac{0.001}{2} \leq x & \text{(Couplage entre la tolérance dimensionnelle et la tolérance de parallélisme)} \\
 2x_2 + \beta_2 h + \gamma_2 H \leq 0.001 & \text{(Équation de couplage générale)} \\
 \beta_2 \leq \frac{0.001}{h} & \text{(Si nous négligeons la translation et la rotation sur w)} \\
 \gamma_2 \leq \frac{0.001}{H} & \text{(Si nous négligeons la translation et la rotation sur v)} \\
 x_2 \leq \frac{0.001}{2} & \text{(Si nous négligeons les rotations)}
 \end{cases}$$

Le plus souvent, $x_2 = 0$

La tolérance de position de la figure 3.6 permet d'établir l'erreur sur la position d'un trou cylindrique. Elle comporte deux composantes en translation et deux composantes en rotation. L'axe du trou doit être positionné à l'intérieur d'une région cylindrique dont l'axe de symétrie de la géométrie est orientée selon le vecteur unitaire w . La tolérance de position nécessite également la mise en place d'un système d'inéquations. En effet, il existe un couplage entre les degrés de liberté du torseur comme nous l'avons explicité précédemment pour la tolérance d'orientation. De plus, ces inéquations permettent d'introduire l'effet de l'environnement sur la tolérance tel que la profondeur h du trou de la figure 3.6. Voici ce système d'inéquations:

$$\begin{cases}
 (2x_3 + \beta_3 h)^2 + (2y_3 + \alpha_3 h)^2 \leq 0.005^2 & \text{(Équation générale de couplage)} \\
 \alpha_3 \leq \frac{0.005}{h} & \text{(Si nous négligeons les translations et la rotation sur v)} \\
 \beta_3 \leq \frac{0.005}{h} & \text{(Si nous négligeons les translations et la rotation sur u)} \\
 x_3^2 + y_3^2 \leq 0.005^2/4 & \text{(Si nous négligeons les rotations)}
 \end{cases}$$

La décision relative à la négligence des mouvements en rotation ou des mouvements en translation est déterminée par la nature du problème. Il faut considérer l'importance des moments générés par les TRANSPORTS des erreurs. Par exemple, nous pouvons négliger les erreurs angulaires lorsque les erreurs longitudinales sont plus

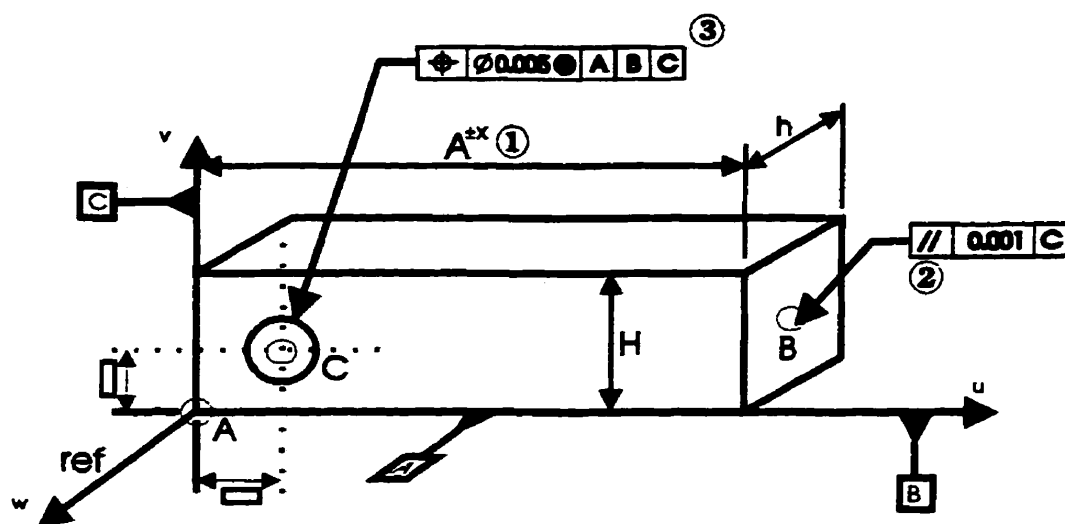
importantes et vice-versa.

La nature des contacts entre les pièces affecte l'action des TRANSPORTS sur les tolérances géométriques. Ainsi, lorsque les contacts plans dominent les contacts cylindriques de certaines liaisons, l'effet des bras de levier peut être négligé car l'erreur angulaire est compensée.

$$\{T_{(1)}\}_{refA} = \begin{bmatrix} x_1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_{refA}$$

$$\{T_{(2)}\}_{refB} = \begin{bmatrix} x_2 & 0 \\ 0 & \beta_2 \\ 0 & \gamma_2 \end{bmatrix}_{refB}$$

$$\{T_{(3)}\}_{refC} = \begin{bmatrix} x_3 & \alpha_3 \\ y_3 & \beta_3 \\ 0 & 0 \end{bmatrix}_{refC}$$



(Les coefficients doivent être interprétés en + ou en -; La référence de déplacement correspond aux dimensions nominales ou moyennes et on exprime un écart par rapport à la position de référence.)

Figure 3.6 Tolérances dimensionnelles et tolérances géométriques modélisées par les torseurs de petits déplacements

Jeux

Les jeux se présentent comme des micro-degrés de liberté entre deux pièces d'une liaison mécanique ou comme les petits déplacements permis à une pièce 1, à partir de sa position nominale, par rapport à une pièce 2 au sein d'une liaison mécanique. Les composantes de ces micro-degrés de liberté peuvent être obtenues par le torseur dual du torseur cinématique de chaque liaison par emboitement. En effet, ces jeux ne peuvent s'exprimer que dans les directions complémentaires des directions cinématiques compatibles avec la liaison. Voici la représentation mathématique adoptée pour définir les torseurs de jeu:

$$\{J_i\}_{ref/M} = \{j_i, \Psi_i\}_{ref/M}$$

où,

- | | | |
|-------------------|---|---|
| $\{J_i\}_{ref/M}$ | : | torseur de jeu d'indice i défini au point M dans le référentiel ref ; |
| j_i | : | vecteur résultant du jeu; |
| Ψ_i | : | vecteur moment de la résultante du jeu. |

Nous définissons les jeux mécaniques comme les dimensions nominales ou moyennes de solides virtuels à corps parfaitement déformables par opposition aux dimensions nominales (ou moyennes) des solides réels à corps non déformable. En d'autres mots, les jeux ne peuvent résister à l'effort. Tout comme les solides réels, les jeux possèdent une dimension nominale (ou moyenne) et des tolérances. La figure 3.7 présente un exemple de solides virtuels soumis à un effort F . Ainsi, dépendamment du sens de l'effort, un solide virtuel particulier est comprimé. La situation "avant" illustre le cas où le solide virtuel 2 est écrasé. La situation "après" illustre le cas où le solide virtuel 1 est écrasé.

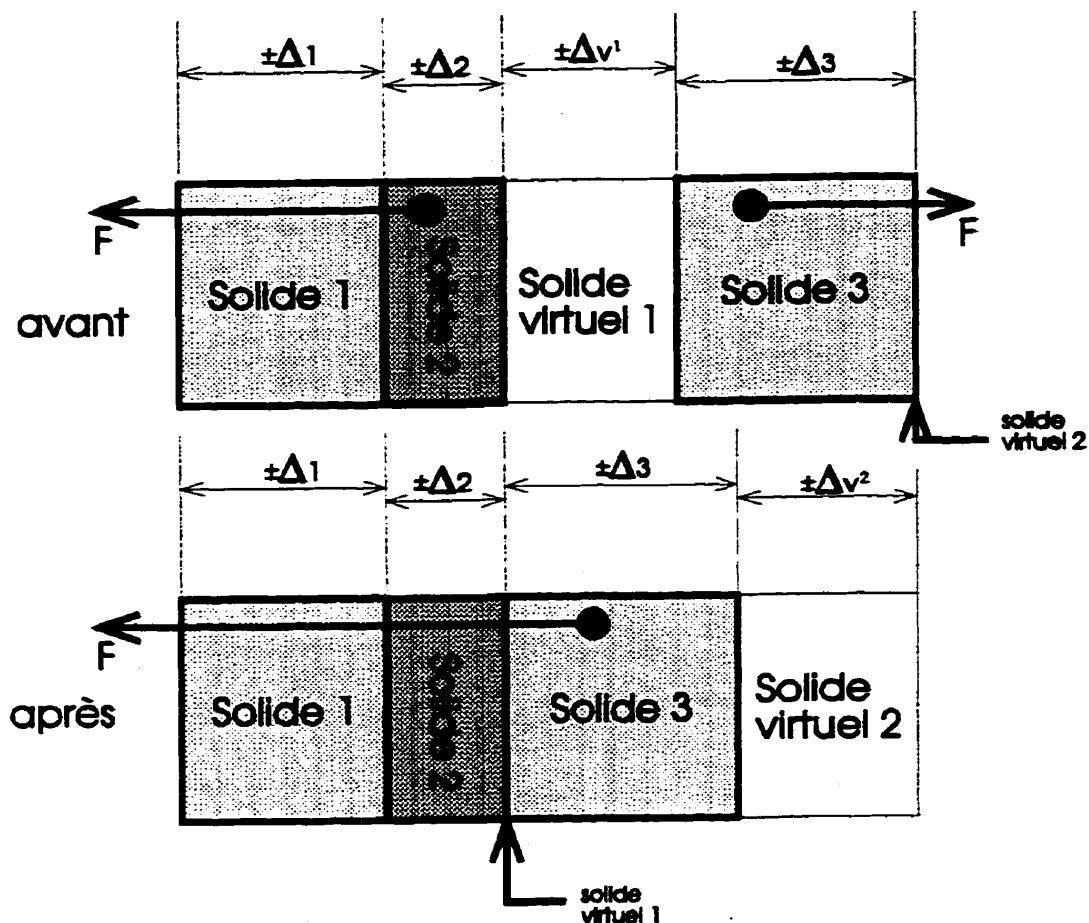


Figure 3.7 Solides virtuels soumis à un effort F

Il est donc possible d'employer le même outil mathématique pour définir les "tolérances" associés aux jeux, que celui employé pour définir les tolérances associées aux surfaces des solides réels soient les torseurs de petits déplacements. La figure 3.8 illustre un exemple de modélisation du jeu dans une liaison pivot par torseur de petits déplacements.

Contraintes:

$$\begin{cases} \theta_{jmn} \leq \frac{jcu}{h} \\ \gamma_{jmn} \leq \frac{jcu}{h} \\ y_{jmn}^2 + z_{jmn}^2 \leq jcu^2/4 \\ (2y_{jmn} + \theta_{jmn}h)^2 + (2z_{jmn} + \gamma_{jmn}h)^2 \leq jcu^2 \end{cases}$$

$$\{T_{jmn}\}_{reg0} = \begin{bmatrix} 0 & 0 \\ y_{jmn} & \theta_{jmn} \\ z_{jmn} & \gamma_{jmn} \end{bmatrix}_{reg0}$$

note: Pour chacune des composantes du torseur dual du torseur cinématique, un micro-degré de liberté existe.

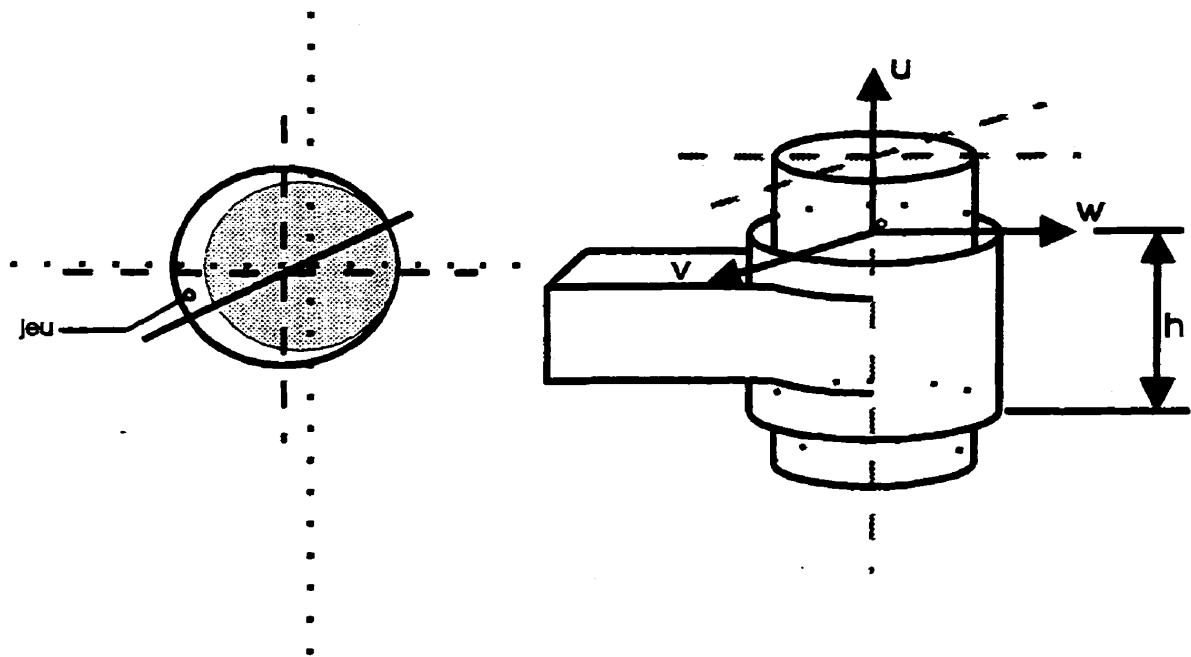








Figure 3.8 Modélisation d'un jeu par torseur de petits déplacements

Tout comme les torseurs employés pour définir les défauts géométriques, les torseurs de jeux nécessitent l'introduction de contraintes limitant la plage de valeurs admissibles. La figure 3.8 présente un pivot avec son système d'inéquations. Nous constatons que les contraintes sont de même nature que celles définies pour la tolérance de position de la figure 3.6.

Remarquons que les serrages peuvent également être modélisés par les torseurs de petits déplacements. Ceux-ci sont définis de la même façon que les tolérances géométriques. En effet, nous pouvons considérer un serrage comme un jeu ayant été fixé instantanément dans une position donnée à l'intérieur d'une enveloppe d'erreurs admissibles.

Le tableau suivant (Tableau 3.1) donne pour les cas de liaisons cinématiques standards les torseurs de petits déplacements modélisant les jeux.

Tableau 3.1: Liaisons cinématiques standards (AFNOR NSE 04-015 / ISO 3952)

Liaisons		Torseurs de jeux
Noms	Symbolisation normalisée (sur les 2 vues)	
glissière d'axe de vecteur unitaire u		$\{J\}_{re00} = \{j, \Psi\}_{re00}$ $j = yv + zw$ $\Psi = \alpha u + \beta v + \gamma w$
pivot d'axe de vecteur unitaire v		$\{J\}_{re00} = \{j, \Psi\}_{re00}$ $j = xu + yv + zw$ $\Psi = \alpha u + \gamma w$
pivot glissant d'axe de vecteur unitaire v		$\{J\}_{re00} = \{j, \Psi\}_{re00}$ $j = xu + zw$ $\Psi = \alpha u + \gamma w$
rotule		$\{J\}_{re00} = \{j, \Psi\}_{re00}$ $j = xu + yv + zw$ $\Psi = 0$
linéaire annulaire d'axe de vecteur unitaire v		$\{J\}_{re00} = \{j, \Psi\}_{re00}$ $j = zw + xu$
glissière hélicoïdale d'axe de vecteur unitaire u		$\{J\}_{re00} = \{j, \Psi\}_{re00}$ $j = yv + zw + xu$ $\Psi = \beta v + \gamma w$

Nous présentons à la section 3.1 les prérequis essentiels à la compréhension des travaux résumés dans ce mémoire. Nous présentons à la section 3.2, comment les

modèles mathématiques sont utilisés pour décrire les tolérances et les jeux mécaniques.

Conclusion

Nous voyons rapidement les raisons qui nous ont mené à utiliser les torseurs de petits déplacements pour l'analyse des dispersions et des jeux mécaniques. En effet, cet outil permet l'uniformisation des manipulations mathématiques d'analyse. Nous pouvons cumuler et transformer les tolérances dimensionnelles, les tolérances géométriques et les jeux grâce aux mêmes outils de calculs vectoriels. Les transformations exponentielles permettent d'exprimer les changements de référentiel et les TRANSPORTS. Les torseurs expriment les tolérances et les jeux dans un format vectoriel compatible avec ces transformations exponentielles. Contrairement à la méthode différentielle, qui ne permet pas une intégration facile de toutes les composantes d'erreurs, la méthode qui sera présentée au chapitre 4.0 donne la possibilité d'accumuler conjointement les erreurs dimensionnelles et les erreurs géométriques. Grâce à cette accumulation, nous évaluons l'effet de chacune des dispersions mécaniques sur l'erreur totale ou sur les conditions fonctionnelles choisies. Nous aurons alors fait le pont entre le caractère local des défauts géométriques et des jeux et le caractère global du fonctionnement du mécanisme.

La direction et le sens des vecteurs résultants des torseurs peuvent être employés pour contenir l'orientation et la valeur min/max des tolérances et des jeux. L'obtention d'un "maximum" global ou d'un "minimum" global d'une accumulation de tolérances est par conséquent, une combinaison appropriée des vecteurs résultants locaux d'erreurs agissant dans un sens propre à chacun d'eux. L'évaluation du poids relatif de chacune des composantes d'erreurs s'obtient par la comparaison des vecteurs correspondant les uns avec les autres comme nous le présentons au chapitre suivant. Cette comparaison est réalisée sur la direction de la condition fonctionnelle ou sur tout autre direction de référence. Le chapitre 4.0 décrit le modèle mathématique vectoriel employé au cours de l'analyse des tolérances et des jeux.

4.0 MODÈLE D'ACCUMULATION DES ERREURS ET DES JEUX DANS UN MÉCANISME

Ce chapitre présente le modèle de calculs vectoriels proposé pour effectuer l'analyse des erreurs mécaniques. Ce modèle permet d'évaluer et de contrôler chacune des erreurs et chacun des jeux tout en respectant certaines conditions d'assemblage ou certaines conditions fonctionnelles.

Que ce soit pour satisfaire l'*assemblabilité* des composantes d'un système mécanique ou pour satisfaire une condition fonctionnelle entre deux composantes, il faut dans tous les cas identifier la boucle cinématique dans laquelle les composantes considérées sont intégrées. Cette recherche suppose l'usage de méthodes d'analyse des cycles au sein d'un graphe. Deux types de graphes sont employés pour réaliser ces analyses: les diagrammes de liaisons et les diagrammes explosés de liaisons.

Les diagrammes de liaisons sont composés de pièces ou des sous-ensembles mécaniques fondamentaux unis grâce à de multiples liaisons. Ces diagrammes permettent d'identifier efficacement les cycles fonctionnels nécessaires au mécanisme étudié. Ils permettent également de mieux visualiser le fonctionnement global des systèmes mécaniques et facilitent l'identification des composantes fondamentales. Ces diagrammes de liaisons sont définis dans plusieurs ouvrages [BER70] et ne sont donc pas décrits en détail dans ce mémoire.

Les diagrammes explosés de liaisons sont formés par l'ensemble des pièces, reliées les unes aux autres par des liaisons élémentaires de contact unissant les surfaces fonctionnelles associées. Ces diagrammes illustrent donc l'ensemble des composantes fonctionnelles d'un mécanisme. Ils sont employés dans le but d'identifier avec précision les relations existantes entre les différentes composantes d'un mécanisme. Ils permettent également de faciliter la recherche des tolérances et des jeux à l'intérieur de chaque

cycle fonctionnel comme nous le présentons au chapitre 6.0. Ces diagrammes sont conçus pour organiser l'information de façon à permettre l'automatisation de l'analyse des cycles fonctionnels.

Les cycles fonctionnels étant détectés, deux cas vont alors se présenter:

- 1- recherche de l'assemblabilité: dans ce cas, le cycle à l'étude doit être ouvert sur une de ses liaisons au point dit, de coupure. La coupure se fait le plus souvent sur la liaison la moins contrainte du cycle. La recherche de l'assemblabilité est donc une vérification de l'existence d'une condition de compatibilité géométrique de fermeture de cycle suffisante pour la réussite du montage des composantes de la boucle du mécanisme étudié.
- 2- satisfaction d'une condition fonctionnelle: le choix de la liaison où se fera la coupure est automatique puisque défini par la condition fonctionnelle. La satisfaction d'une telle condition peut être employée pour valider le choix des tolérances dans un mécanisme quelconque.

Une coupure étant définie, il faut procéder au choix du solide origine sur lequel sera défini le référentiel global d'analyse. Cette pièce est choisie le plus souvent sur le bâti de la machine étudiée ou bien sur un des éléments de la liaison cinématique la plus contrainte (nombre de degrés de liaison le plus élevé). Le choix de ce solide d'origine se justifie par le fait qu'il représente pour le premier cas, la structure sur laquelle repose toutes les autres composantes et dans le second cas, l'élément de référence nécessaire au fonctionnement du mécanisme. Le cycle est alors décomposé en deux sous-chaînes cinématiques partant toutes deux de la pièce origine et se terminant sur la coupure. Nous nommons ces sous-chaînes: demi-boucles car ils représentent un des deux parcours pour réaliser le bouclage lors d'une analyse.

Nous présentons à la section 4.1 les modèles mathématiques utilisés pour étudier les mécanismes dans les deux cas d'analyse envisagés soit la recherche de l'assemblabilité et la satisfaction d'une condition fonctionnelle. Nous discutons brièvement à la section 4.2 de la procédure d'analyse à suivre lorsqu'un mécanisme à boucles multiples est étudié. Nous présentons à la section 4.3, la méthode employée pour définir les référentiels d'analyse. Nous résumons à la section 4.4, la procédure systématique à suivre lors d'une analyse.

4.1 Conditions fonctionnelles et *assemblabilité*

Cette section décrit les modèles mathématiques permettant de prendre en compte les effets des conditions fonctionnelles sur le résultat de l'assemblage. Ces mêmes modèles mathématiques sont employés pour réaliser l'étude des erreurs et des jeux mécaniques à l'intérieur des boucles fonctionnelles, lorsque l'objectif est de vérifier l'*assemblabilité*.

Nous devons définir un nouvel opérateur mathématique pour présenter l'inéquation principale:

-Opérateur de projection vectoriel (\otimes):

$$\{J_i\}_{\text{refM}} \otimes \{T_i\}_{\text{refM}} = \{\bar{J}_i \cdot \bar{R}_i; \bar{\Psi}_i \cdot \bar{M}_i\}$$

Cet opérateur définit la projection vectorielle d'un torseur quelconque $\{J_i\}_{\text{refM}}$ sur un torseur $\{T_i\}_{\text{refM}}$. Il ne correspond en fait, qu'à un simple produit scalaire des vecteurs résultants et des vecteurs moments de torseurs quelconques. Il est employé pour aligner les jeux de chacune des liaisons avec les vecteurs d'inter-efforts associés. En effet, les jeux ne peuvent agir que selon cette direction.

Ayant défini cet opérateur, il est maintenant possible de formuler l'inéquation

d'analyse des accumulations des jeux et des défauts géométriques en fonction des conditions fonctionnelles et des contraintes fonctionnelles:

$$\begin{aligned}
 & \left(\sum_{q=1..s} \{E^{(1)}\}_{refM} - \sum_{r=1..t} \{E^{(2)}\}_{refM} \right) \otimes \{C\}_{refM} \\
 & \leq \\
 & \{C\}_{refM} \\
 & - \mp \\
 & j=1..m \\
 & \left(\sum_{i=1..n} \{U^{(1)}\}_{refM} \otimes \{T^{(1)}\}_{refM} \pm \sum_{j=1..m} \{U^{(2)}\}_{refM} \otimes \{T^{(2)}\}_{refM} \right) \otimes \{C\}_{refM}
 \end{aligned}$$

- + : Si nous cherchons à vérifier l'assemblabilité et que nous supposons les pièces à l'état maximum de matière pour les éléments dimensionnels tolérancés (avec J_{min} et B_{min}).
- : Si nous cherchons à vérifier une condition fonctionnelle et que nous supposons les pièces à l'état minimum de matière pour les éléments dimensionnels tolérancés (avec J_{max} et B_{max}).

note : les bornes introduites par cet état pour une cotation \ominus , n'ont pas été pris en compte.

où,

$\{E^{(x)}_y\}_{refM}$: torseur d'erreurs dimensionnelles et d'erreurs géométriques associé à la demi-boucle "x". "y" est l'indice de la composante sur laquelle l'erreur est appliquée. Ce torseur est défini au point M dans le référentiel ref;

$\{J_y^{(x)}\}_{ref/M}$:	torseur des jeux associé à la demi-boucle "x" pour chacune des liaisons d'indice "y". Ce torseur est défini au point M dans le référentiel ref;
$\{C\}_{ref/M}$:	torseur de la condition fonctionnelle à satisfaire sur la liaison de coupure.
$\{T_y^{(x)}\}_{ref/M}$:	torseur d'inter-effort d'une liaison d'indice "y", possédant un jeu et contenue dans la demi-boucle "x". Ce torseur est défini au point M et dans le référentiel ref;
s	:	nombre de composantes dans la demi-boucle 1;
t	:	nombre de composantes dans la demi-boucle 2;
n	:	nombre de liaisons dans la demi-boucle 1;
m	:	nombre de liaisons dans la demi-boucle 2.

Remarquons que cette inéquation n'est pas suffisante pour définir complètement les relations entre les jeux, les défauts géométriques et les conditions fonctionnelles. Nous devons introduire les systèmes de contraintes qui sont préalablement définis pour chacune des tolérances et chacun des jeux. De plus, la vérification de l'*assemblabilité* ou la vérification du fonctionnement d'un mécanisme, nécessite l'interprétation de l'inéquation sous une forme convenable.

Notons également que les dimensions nominales (moyennes) du cycle fonctionnel doivent respecter sa condition de fermeture géométrique. Cette condition s'exprime par l'équation suivante :

$$\sum_{q=1..s} (Dim^{(1)}_q) - \sum_{r=1..t} (Dim^{(2)}_r) = 0$$

où,

- $(\text{Dim}^{(n)}_y)$: vecteur déplacement des dimensions nominales des composantes "y" associées à chacun des demi-boucles fonctionnelles "x";
- s : nombre de composantes dans la demi-boucle 1;
- t : nombre de composantes dans la demi-boucle 2.

Dans le cas où les jeux considérés dans le mécanisme sont jugés négligeables, l'inéquation d'analyse vectorielle des erreurs peut se réécrire de la façon suivante:

$$\left(\sum_{q=1..s} \{E^{(1)}_q\}_{refM} - \sum_{r=1..t} \{E^{(2)}_r\}_{refM} \right) \otimes \{C\}_{refM} \leq \{C\}_{refM}$$

Enfin, si la liaison de coupure est une liaison forte (liaison par emboîtement), la condition fonctionnelle s'exprime alors par le torseur jeu dans la liaison. Si les jeux sont supposés négligeables dans ces conditions, l'équation s'écrira de la manière suivante:

$$\left(\sum_{q=1..s} \{E^{(1)}_q\}_{refM} - \sum_{r=1..t} \{E^{(2)}_r\}_{refM} \right) = \{0\}$$

Dans ce contexte, seule l'égalité est acceptable car autrement l'assemblage ne pourrait être réalisé.

Cette condition peut être retrouvée dans le cas du test d'*assemblabilité* avec les mêmes hypothèses. Ainsi, pour un mécanisme à simple boucle, elle peut être utilisée pour étudier les variations d'une variable de position attachée à un mouvement de sortie en fonction des variations dimensionnelles des composantes du mécanisme et en fonction de la variable de position attachée à un mouvement d'entrée. C'est avec cette

condition, que nous développons l'exemple du générateur d'équations à quatre membrures, présenté au chapitre 5.0.

Nous avons présenté dans cette section, les modèles mathématiques employés pour réaliser une analyse des erreurs et des jeux en respectant les conditions fonctionnelles et les conditions d'assemblage pour une boucle fonctionnelle unique et ce, de deux manières différentes:

1. soit en respectant simultanément les conditions fonctionnelles et les conditions d'assemblage;
2. soit en respectant les conditions fonctionnelles ou les conditions d'assemblage.

Remarquons que la vérification des conditions fonctionnelles et des conditions d'assemblages sont réalisées simultanément. En effet, ces deux contraintes sont complémentaires l'une de l'autre.

Nous discutons brièvement à la sous-section suivante de la procédure à suivre dans le cas où plusieurs boucles fonctionnelles sont prises en compte.

4.2 Boucles fonctionnelles multiples

Les mécanismes sont habituellement définis par plus d'une boucle fonctionnelle. En effet, il existe peu de modèle comme celui du générateur d'équations à quatre membrures, présenté au chapitre 5.0, qui possède une seule boucle fonctionnelle.

La première étape de l'analyse d'un mécanisme possédant plus d'une boucle, débute par la reconnaissance de ses cycles fonctionnels. Cette étape est franchie lorsque le diagramme explosé des liaisons est complété. Comme nous l'avons déjà exposé,

celui-ci illustre l'ensemble des liaisons et relations existantes entre chacune des composantes fonctionnelles du mécanisme. Par la suite, l'analyse est réalisée, sur chacune des boucles, en employant la méthode vectorielle proposée. Toutefois, lorsqu'il y a couplage entre certaines branches des cycles fonctionnelles quelques précautions doivent être prises. Des contraintes sont introduites de façon à rendre compatible les torseurs de tolérancement et les torseurs de jeux associés à ces branches communes. Ces contraintes sont résolues avec les systèmes d'inéquations générées lors du parcours de tous les cycles fonctionnels. Signalons que nous n'avons pas poussé plus avant notre étude sur cet aspect du problème.

4.3 Référentiel proposé

Chacun des torseurs modélisant un jeu ou une tolérance doit être positionné et orienté de manière à faciliter la mise en équation initiale et finale. Les torseurs sont donc organisés localement sur chacune des composantes incluses dans l'analyse. Le choix de la position et de l'orientation des référentiels est effectué de façon naturelle. Ainsi, dans le cas du générateur d'équations à quatre membrures, ces référentiels ont été positionnés à la première extrémité de chacune des membrures et orientés de manière à ce que la direction du vecteur unitaire u soit toujours parallèle à la direction de ces dernières. La figure 5.2 du chapitre 5.0 présente un exemple de cette organisation des référentiels. Cette méthode de positionnement des référentiels semble suffisante compte tenu des résultats observés aux chapitres 5.0 et 6.0.

4.4 Résumé de la procédure d'analyse

Voici le processus employé pour l'analyse de la propagation des erreurs mécaniques et des jeux. Évidemment, cette procédure n'est valable que pour l'analyse d'une seule boucle fonctionnelle. Les étapes de cette procédure seront légèrement différentes dans le cas d'une analyse d'un mécanisme à boucles multiples.

1. **Création du diagramme des liaisons afin de visualiser le fonctionnement du mécanisme, d'identifier les boucles fonctionnelles et de cerner les composantes essentielles.**
2. **Définition de la condition fonctionnelle et (ou) de la condition d'assemblage.**
3. **Choix de la liaison de coupure en respectant la définition de l'étape 2.**
4. **Création du diagramme explosé des liaisons sur chacune des boucles. Ce diagramme permet d'organiser les référentiels d'analyse et d'identifier les tolérances dimensionnelles, les tolérances géométriques et les jeux qui sont considérés. Ce diagramme permet également d'observer l'écoulement du flux fonctionnel.**
5. **Identification des tolérances et des jeux (voir étape 4).**
6. **Identification des référentiels locaux d'analyse (voir étape 4).**
7. **Choix d'un référentiel global d'analyse.**
8. **Définition des torseurs de tolérancement et de jeux à chacun des référentiels locaux.**
9. **Définition des inéquations associées aux contraintes de modélisation des tolérances géométriques et des jeux.**
10. **Définition du torseur de la condition fonctionnelle ou de la condition d'assemblage.**
11. **Recherche du parcours pour l'écoulement du flux fonctionnel.**
12. **Accumulation des torseurs selon l'équation présentée à la section 4.1.**
13. **Résolution.**

Nous présentons dans ce chapitre, le modèle mathématique développé pour atteindre les objectifs visés par ce travail de maîtrise. Nous discutons également des méthodes employées pour définir les référentiels et imaginons de quelle manière les mécanismes à boucles multiples sont résolus. Nous présentons à la dernière section, les étapes de la procédure d'analyse. Nous présentons au chapitre suivant, les résultats des

travaux réalisés, en vue de la validation du modèle mathématique d'analyse vectorielle des erreurs et des jeux mécaniques.

5.0 VALIDATION DE LA MÉTHODOLOGIE

Nous allons présenter dans ce chapitre un exemple de développement mathématique illustrant l'utilisation de la procédure d'analyse vectorielle des erreurs des systèmes mécaniques. Nous avons choisi d'effectuer notre démonstration à l'aide d'un modèle très bien connu: *le générateur d'équations à quatre membrures*. En effet, un grand nombre d'auteurs ont réalisé des travaux sur ce modèle, ce qui facilite l'évaluation de la méthode d'analyse proposée.

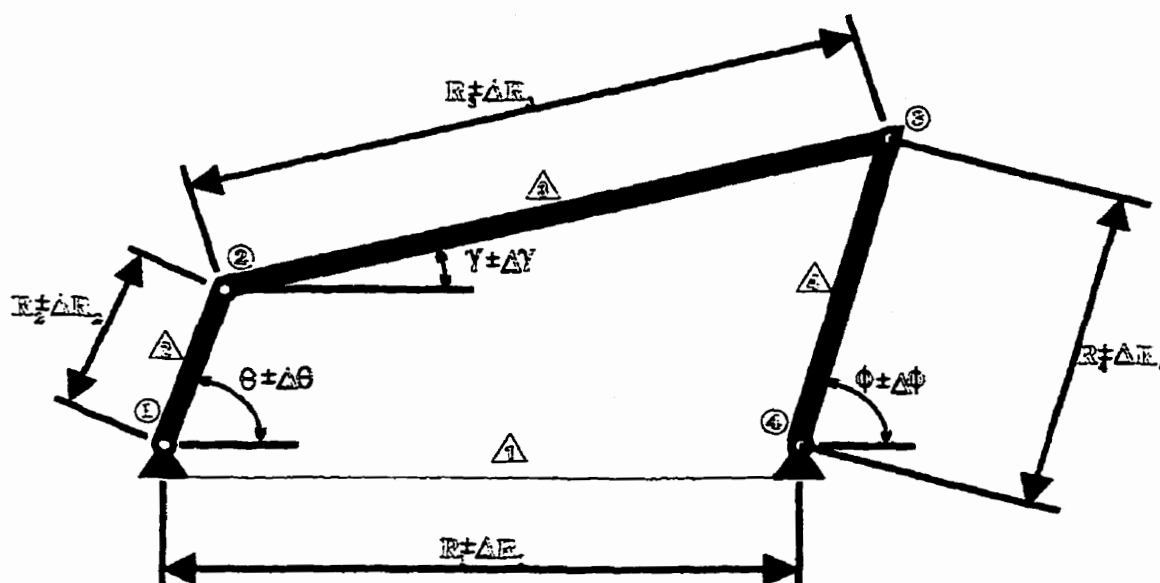


Figure 5.1 Générateur d'équations à quatre membrures

Pour valider la méthodologie présentée au chapitre 4.0, la section 5.1 débute par une définition du problème. Ainsi, le générateur d'équations à quatre membrures est analysé de façon à établir les inéquations mathématiques nécessaires à la résolution du problème. Nous présentons à la section 5.2, les résultats expérimentaux obtenus. Ces résultats permettent de comparer la méthode d'analyse différentielle à la nouvelle méthode vectorielle qui est proposée dans ce mémoire.

5.1 Définition du problème

Fenton, Cleghorn et Fu [FEN89,CLE93,FU88] ont réalisé quelques travaux à l'aide du générateur d'équations à quatre membrures. Nous nous servons de leurs résultats pour valider notre procédure de calcul. Pour respecter les paramètres d'analyse employés par cette équipe [CLE93,FU88], nous nous limitons à des erreurs dimensionnelles sur les membrures 1,2,3 et 4. Les développements mathématiques qui sont présentés dans ce chapitre permettent de calculer l'erreur sur l'angle ϕ (sortie) lorsqu'un angle θ (entrée) est donné. Nous plaçons la coupure au noeud 3 (Figure 5.1) du générateur et choisissons de placer le solide d'origine au noeud 4 lié au sol. Le référentiel global est lié au solide d'origine. La figure ci-dessous montre de quelle façon les référentiels locaux sont positionnés dans l'espace et comment nous réalisons le bouclage entre la pièce origine et la coupure.

À noter que chacun des torseurs d'erreurs possède un référentiel propre de manière à exprimer les erreurs selon l'axe du vecteur unitaire approprié.

$$\{T_{(2)}\}_{ref4OI} \begin{bmatrix} \Delta R_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_{ref4OI}$$

$$\{T_{(4)}\}_{ref4OZ} \begin{bmatrix} \Delta R_3 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_{ref4OZ}$$

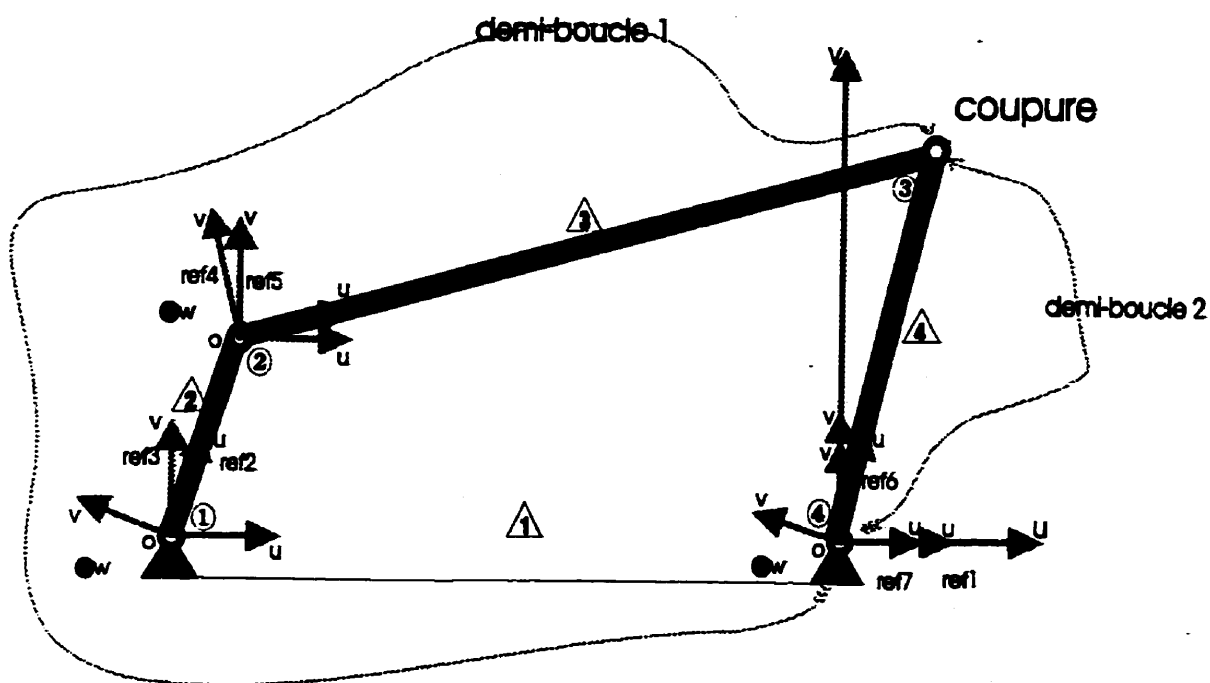
$$\{T_{(5)}\}_{ref4OZ} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \Delta \gamma \end{bmatrix}_{ref4OZ}$$

$$\{T_{(3)}\}_{ref4OI} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \Delta \theta \end{bmatrix}_{ref4OI}$$

$$\{T_{(7)}\}_{ref7OI} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \Delta \phi \end{bmatrix}_{ref7OI}$$

$$\{T_{(8)}\}_{ref7OI} \begin{bmatrix} \Delta R_4 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_{ref7OI}$$

$$\{T_{(1)}\}_{ref1OI} \begin{bmatrix} -\Delta R_1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_{ref1OI}$$



note : ici, aucune contrainte n'est à exprimer dans le cas de ces torseurs

Figure 5.2 Générateur d'équations, boucles et torseurs d'erreurs

Comme nous l'avons mentionné précédemment, nous devons effectuer l'accumulation des erreurs en parcourant chacune des deux demi-boucles présentées à la figure 5.2. Remarquons que chacun des vecteurs d'erreurs est orienté dans le sens du parcours de chacune des demi-boucles soit: de l'origine vers la coupure. Les calculs d'accumulation (transports et changements de référentiel) servent à exprimer adéquatement les torseurs dans un même référentiel et à calculer l'action des erreurs mécaniques au point de coupure. Les expressions symboliques d'accumulations s'illustrent comme suit:

(1)

$$\begin{array}{c} \text{Transport} \\ \{ T_{(1)} \}_{\text{ref101}} \xrightarrow{\overline{d_1}} \{ T_{(1)} \}_{\text{ref/coupure}} \\ \\ \text{CR} \\ \{ T_{(1)} \}_{\text{ref/coupure}} \xrightarrow{\overline{\lambda_1}} \{ T_{(1)} \}_{\text{ref/coupure}} \end{array}$$

Transformation réalisée pour le transport ($\overline{d_1}$) : $T(u, -R_1)$, $R(w, \theta)$, $T(u, R_2)$, $R(w, \gamma - \theta)$, $T(u, R_3)$.

Aucune transformation pour le changement de référentiel ($\overline{\lambda_1}$) :

(2)

$$\begin{array}{c} \text{Transport} \\ \{ T_{(2)} \}_{\text{ref201}} \xrightarrow{\overline{d_2}} \{ T_{(2)} \}_{\text{ref/coupure}} \\ \\ \text{CR} \\ \{ T_{(2)} \}_{\text{ref/coupure}} \xrightarrow{\overline{\lambda_2}} \{ T_{(2)} \}_{\text{ref/coupure}} \end{array}$$

Transformation réalisée pour le transport (d_2) : $T(u, R_2)$, $R(w, \gamma - \theta)$, $T(u, R_3)$.

Transformation réalisée pour le changement de référentiel (λ_2) :

$T(u, -R_1)$, $R(w, \theta)$.

(3)

$$\{T_{(3)}\}_{ref\#01} \xrightarrow[\overline{d_3}]{Transport} \{T_{(3)}\}_{ref\#coupsure}$$

$$\{T_{(3)}\}_{ref\#coupsure} \xrightarrow[\overline{\lambda_3}]{CR} \{T_{(3)}\}_{ref\#coupsure}$$

Transformation réalisée pour le transport (d_3) : $R(w, \theta)$, $T(u, R_2)$, $R(w, \gamma - \theta)$, $T(u, R_3)$.

Aucune transformation réalisée pour le changement de référentiel (λ_3) :

(4)

$$\{T_{(4)}\}_{ref\#02} \xrightarrow[\overline{d_4}]{Transport} \{T_{(4)}\}_{ref\#coupsure}$$

$$\{T_{(4)}\}_{ref\#coupsure} \xrightarrow[\overline{\lambda_4}]{CR} \{T_{(4)}\}_{ref\#coupsure}$$

Transformation réalisée pour le transport (d_4) : $T(u, R_3)$.

Transformation réalisée pour le changement de référentiel (λ_4) :

$T(u, -R_1)$, $R(w, \theta)$, $T(u, R_2)$, $R(w, -\theta + \gamma)$.

(5)

$$\{T_{(5)}\}_{\text{réf}O_2} \xrightarrow[\overline{d_5}]{\text{Transport}} \{T_{(5)}\}_{\text{réf}O_{\text{compure}}}$$

$$\{T_{(5)}\}_{\text{réf}O_{\text{compure}}} \xrightarrow[\overline{\lambda_5}]{CR} \{T_{(5)}\}_{\text{réf}O_{\text{compure}}}$$

Transformation réalisée pour le transport ($\overline{d_5}$) : $R(w, \gamma)$, $T(u, R_3)$.

Aucune transformation réalisée pour le changement de référentiel ($\overline{\lambda_5}$) :

(6)

$$\{T_{(6)}\}_{\text{réf}O_4} \xrightarrow[\overline{d_6}]{\text{Transport}} \{T_{(6)}\}_{\text{réf}O_{\text{compure}}}$$

$$\{T_{(6)}\}_{\text{réf}O_{\text{compure}}} \xrightarrow[\overline{\lambda_6}]{CR} \{T_{(6)}\}_{\text{réf}O_{\text{compure}}}$$

Transformation réalisée pour le transport ($\overline{d_6}$) : $T(u, R_4)$.

Transformation réalisée pour le changement de référentiel ($\overline{\lambda_6}$) : $R(w, \phi)$.

(7)

$$\begin{array}{c} \text{Transport} \\ \{T_{(7)}\}_{ref/04} \xrightarrow{\overline{d_7}} \{T_{(7)}\}_{ref/compteur} \\ \\ \text{CR} \\ \{T_{(7)}\}_{ref/compteur} \xrightarrow{\overline{\lambda_7}} \{T_{(7)}\}_{ref/compteur} \end{array}$$

Transformation réalisée pour le transport ($\overline{d_7}$) : $R(w, \phi)$, $T(u, R_4)$.

Aucune transformation pour le changement de référentiel ($\overline{\lambda_7}$) :

Les vecteurs locaux d'erreurs sont obtenus après développement des expressions symboliques présentées ci-dessus.

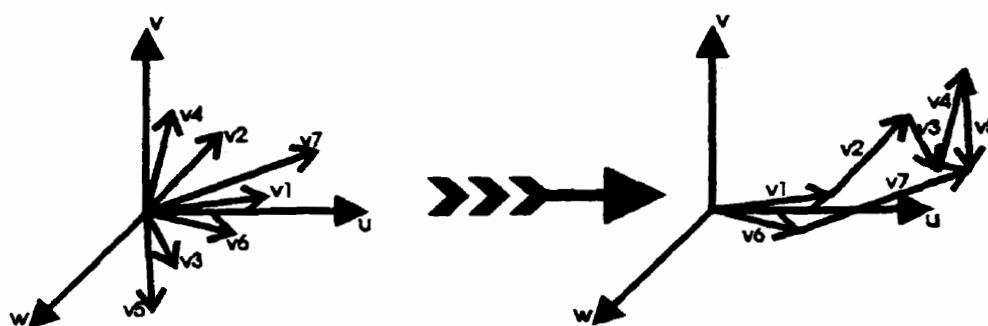


Figure 5.3 Schéma des vecteurs composants les vecteurs d'erreurs totaux.

vecteur local d'erreurs 1 (v_1) - Résultante du torseur $\{T_{(1)}\}_{ref/compteur}$:

$$\{\Delta R_1\}u + \{0\}v + \{0\}w$$

vecteur local d'erreurs 2 (v2) - Résultante du torseur $\{T_{(2)}\}_{ref/coupure}$:

$$\{(\Delta R_2) \cdot \cos(\theta)\}u + \{(\Delta R_2) \cdot \sin(\theta)\}v + \{0\}w$$

vecteur local d'erreurs 3 (v3) - Résultante du torseur $\{T_{(3)}\}_{ref/coupure}$:

$$\{-\Delta R_2 \cdot (R_3 \cdot \sin(\gamma) + R_2 \cdot \sin(\theta))\}u + \{\Delta R_2 \cdot (R_3 \cdot \cos(\gamma) + R_2 \cdot \cos(\theta))\}v + \{0\}w$$

vecteur local d'erreurs 4 (v4) - Résultante du torseur $\{T_{(4)}\}_{ref/coupure}$:

$$\{-\Delta \theta \cdot (R_3 \cdot \sin(\gamma) + R_2 \cdot \sin(\theta))\}u + \{\Delta \theta \cdot (R_3 \cdot \cos(\gamma) + R_2 \cdot \cos(\theta))\}v + \{0\}w$$

vecteur local d'erreurs 5 (v5) - Résultante du torseur $\{T_{(5)}\}_{ref/coupure}$:

$$\{-\Delta \gamma \cdot R_3 \cdot \sin(\gamma)\}u + \{\Delta \gamma \cdot R_3 \cdot \cos(\gamma)\}v + \{0\}w$$

vecteur local d'erreurs 6 (v6) - Résultante du torseur $\{T_{(6)}\}_{ref/coupure}$:

$$\{\Delta R_4 \cdot \cos(\phi)\}u + \{\Delta R_4 \cdot \sin(\phi)\}v + \{0\}w$$

vecteur local d'erreurs 7 (v7) - Résultante du torseur $\{T_{(7)}\}_{ref/coupure}$:

$$\{-\Delta \phi \cdot R_4 \cdot \sin(\phi)\}u + \{\Delta \phi \cdot R_4 \cdot \cos(\phi)\}v + \{0\}w$$

Les vecteurs locaux d'erreurs composent les deux vecteurs totaux d'erreurs. Les vecteurs locaux 1 à 5 forment le premier vecteur total d'erreurs (demi-boucle 1) et les vecteurs locaux 6 et 7 forment le second vecteur total d'erreurs (demi-boucle 2). L'addition vectorielle des vecteurs locaux d'erreurs peut s'écrire:

$$\Delta V_I = v1 + v2 + v3 + v4 + v5$$

$$\Delta V_{II} = v6 + v7$$

En développant l'ensemble de ces équations, nous obtenons les deux vecteurs résultants des torseurs d'erreurs totaux associés à la demi-boucle 1 et à la demi-boucle 2:

ΔV_I (demi-boucle 1) - Résultante du torseur $\{E_{res}^{(1)}\}$:

$$\{ \Delta r1 + \Delta r2 \cos(\theta) + \Delta r3 \cos(\gamma) - \Delta \gamma r3 \sin(\gamma) \} u$$

+

$$\{ \Delta r2 \sin(\theta) + \Delta r3 \sin(\gamma) + \Delta \gamma r3 \cos(\gamma) \} v$$

+

$$\{0\} w$$

ΔV_{II} (demi-boucle 2) - Résultante du torseur $\{E_{res}^{(2)}\}$:

$$\{ \Delta r4 \cos(\phi) - \Delta \phi r4 \sin(\phi) \} u$$

+

$$\{ \Delta r4 \sin(\phi) + \Delta \phi r4 \cos(\phi) \} v$$

+

$$\{0\} w$$

Lorsque la liaison de coupure est une liaison forte (liaison par emboitement), la condition fonctionnelle s'exprime par le torseur nul ($\{0\}$). Si les jeux sont supposés négligeables, l'équation s'écrira de la manière suivante (section 4.1):

$$\left(\sum_{q=1..s} \{E^{(1)}\}_{q,refM} - \sum_{r=1..t} \{E^{(2)}\}_{r,refM} \right) = \{0\}$$

où,

ref : référentiel d'accumulation des erreurs;

M : point de coupure où nous réalisons l'accumulation des erreurs;

s : nombre de composantes associées à la demi-boucle 1;

t : nombre de composantes associées à la demi-boucle 2.

Dans ce contexte, seule l'égalité est acceptable car autrement l'assemblage ne pourrait être réalisé. Les deux vecteurs ΔV_I et ΔV_{II} doivent donc être égaux en

grandeur, en sens et en direction de manière à respecter la condition de fermeture cinématique vue précédemment. Nous réécrivons cette équation de la façon suivante:

$$\Delta V_I - \Delta V_{II} = 0$$

Remarquons que l'équation précédente définit uniquement la fermeture cinématique des vecteurs résultants des torseurs d'erreurs. Il existe évidemment une équation similaire pour assurer la fermeture cinématique des vecteurs moments des torseurs d'erreurs. Nous ne l'avons pas défini considérant qu'elle n'apporte aucune donnée supplémentaire pour le présent problème.

Tout comme l'équation d'accumulation d'erreurs, il existe une équation définissant la fermeture géométrique du cycle fonctionnel des dimensions nominales (moyennes). En effet, il est nécessaire que l'accumulation des dimensions nominales de chacune des demi-boucles convergent vers un point de coupure unique. La formulation générale de cette équation est définie au chapitre 4.0 de la façon suivante:

$$\sum_{q=1..s} (Dim^{(1)}_q) - \sum_{r=1..t} (Dim^{(2)}_r) = 0$$

où,

- | | | |
|-----------------|---|--|
| $(Dim^{(x)}_y)$ | : | vecteur déplacement des dimensions nominales des composantes "y" associées à chacun des demi-boucles fonctionnelles "x"; |
| s | : | nombre de composantes dans la demi-boucle 1; |
| t | : | nombre de composantes dans la demi-boucle 2. |

Nous reformulons cette équation pour satisfaire la notation choisie:

$$\mathbf{V}_I - \mathbf{V}_{II} = \mathbf{0}$$

avec, $\mathbf{V}_I = T(\mathbf{u}, -R_1), R(\mathbf{w}, \theta), T(\mathbf{u}, R_2), R(\mathbf{w}, \gamma - \theta), T(\mathbf{u}, R_3).$

et, $\mathbf{V}_{II} = R(\mathbf{w}, \phi), T(\mathbf{u}, R_4).$

Pour obtenir l'erreur sur l'angle ϕ , nous n'avons qu'à résoudre le problème d'égalité des deux vecteurs d'erreurs totaux pour un angle θ donné. Nous arrivons à formuler le système d'équations suivant:

projection/u:

$$(\Delta R_1 + \Delta R_2 \cos(\theta) + \Delta R_3 \cos(\gamma) - \Delta \gamma R_3 \sin(\gamma)) -$$

$$(\Delta R_4 \cos(\phi) - \Delta \phi R_4 \sin(\phi)) = 0$$

projection/v:

$$(\Delta R_2 \sin(\theta) + \Delta R_3 \sin(\gamma) + \Delta \gamma R_3 \cos(\gamma)) -$$

$$(\Delta R_4 \sin(\phi) + \Delta \phi R_4 \cos(\phi)) = 0$$

Les inconnues du système d'équations sont $\Delta \phi$ et $\Delta \gamma$.

Nous présentons à la section 5.1, la définition du problème. Nous présentons à la section 5.2, les résultats de l'expérimentation obtenus à partir des équations de la définition du problème. Une discussion suit la présentation de ces résultats.

5.2 Expérimentation

Dans cette section, nous allons vérifier numériquement que les résultats obtenus, grâce à la nouvelle méthode d'analyse qui est proposée dans ce mémoire, permet d'obtenir des résultats équivalents à ceux obtenus par la méthode d'analyse différentielle. Ces résultats sont comparés à ceux de la solution mathématique exacte.

La solution exacte est obtenue d'une équation liant l'angle d'entrée à l'angle de

sortie telle que présentée ci-dessous. Cette équation est extraite des travaux de recherche de Dhande et Chakraborty [DHA73].

$$\phi = 2 \tan^{-1} \left[\frac{A \pm D}{B + C} \right]$$

$$A = \sin \theta$$

$$B = \cos \theta - \frac{R_1}{R_2}$$

$$C = \frac{R_1^2 + R_2^2 - R_3^2 + R_4^2}{2R_2R_4} - \frac{R_1}{R_4} \cos \theta$$

$$D = +\sqrt{(A^2 + B^2 - C^2)}$$

où,

- θ : représente l'angle d'entrée;
- ϕ : représente l'angle de sortie;
- R_1 : représente la longueur de la membrure 1;
- R_2 : représente la longueur de la membrure 2;
- R_3 : représente la longueur de la membrure 3;
- R_4 : représente la longueur de la membrure 4.

Remarquons que les mêmes résultats sont obtenus par l'expression de la condition de fermeture géométrique sur les nominaux telle que définie précédemment. C'est d'ailleurs avec une telle relation que nous avons choisi les dimensions nominales (moyennes) des membrures du générateur d'équations à quatre membrures.

Les termes associés à la méthode différentielle sont déduits par différentiation à partir de la solution exacte et en s'inspirant des modèles d'équations différentielles développés par les auteurs Fu, Cleghorn et Fenton [FU88]. Les équations différentielles ont été générées et résolues numériquement grâce au logiciel de calcul symbolique

MAPLE V version 2. Par la suite, ces résultats numériques ont été importés dans le chiffrier électronique EXCEL pour fin de comparaison avec les résultats de la solution exacte et les résultats de la méthode d'analyse vectorielle. Ces derniers résultats sont obtenus grâce aux outils informatiques développés au cours des présents travaux de recherche. Le chapitre 7.0 en présente les éléments principaux.

Nous avons calculé numériquement les erreurs sur l'angle ϕ pour une rotation de 360 degrés de l'angle θ . Nous avons réalisé nos essais en choisissant les paramètres suivants: $R_1 = 10.0$, $R_2 = 4.0$, $R_3 = 12.0$, $R_4 = 10.3923$, $\Delta R_1 = \pm 0.05$, $\Delta R_2 = \pm 0.05$, $\Delta R_3 = \pm 0.05$ et finalement $\Delta R_4 = \pm 0.05$. En fait, ces valeurs correspondent aux dimensions choisies pour des angles $\theta = 0$ et $\phi = 90$ degrés, par les auteurs [FU88] des travaux de recherche sur lesquels nous comparons nos résultats. La figure 5.1 donne la signification de ces paramètres. Remarquons que les longueurs nominales des membrures (R_1 , R_2 , R_3 et R_4) sont choisies de manière à satisfaire la condition vectorielle de fermeture géométrique. Il existe 81 (3^4) combinaisons possibles d'erreurs pour un générateur d'équations à quatre membrures, c'est-à-dire, 81 façons d'organiser le choix des bornes minimum et maximum des tolérances des quatres membrures du générateur. La figure 5.4 présente une seule des combinaisons possibles soit le cas où $\Delta R_1 = 0.05$, $\Delta R_2 = 0.05$, $\Delta R_3 = 0.05$ et $\Delta R_4 = 0.05$. Les essais réalisés montrent que l'ensemble des 81 combinaisons conduisent à des résultats similaires à ceux obtenus grâce à l'approche différentielle.

Comparaison de la méthode d'analyse différentielle, de la méthode d'analyse vectorielle par rapport à la solution exacte

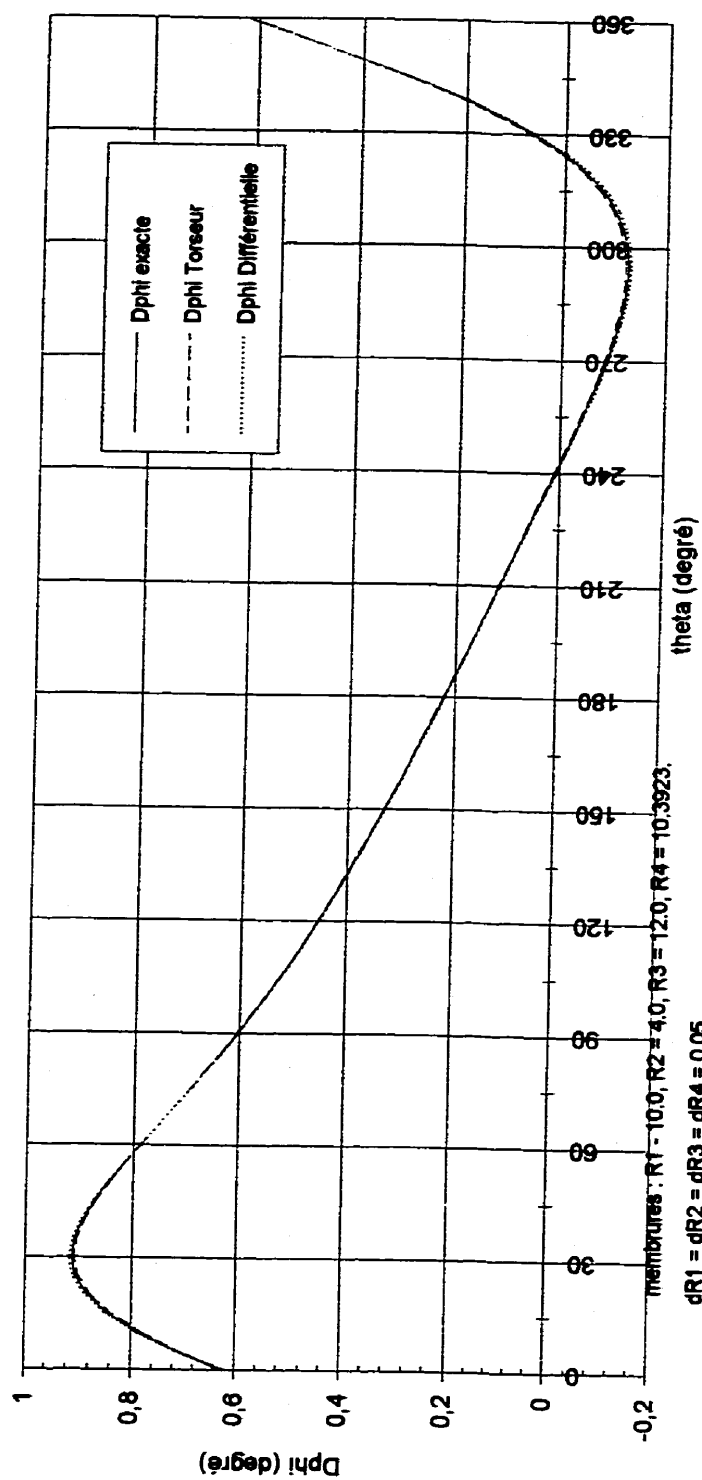


Figure 5.4 Essai pour ΔR_i à ± 0.05 .

La figure 5.4 illustre trois courbes permettant de comparer les résultats obtenus par l'approche différentielle, ceux obtenus utilisant les torseurs et la solution exacte. Nous constatons que la méthode basée sur les torseurs donne des résultats similaires à celle employant les différentielles. Les trois courbes sont confondues. En effet, nous constatons que la différence entre les valeurs générées par la méthode différentielle et les valeurs générées par la méthode utilisant les torseurs est inférieure à 0.1%. Nous sommes satisfaits de ces résultats compte tenu que les algorithmes de résolution employés ne sont pas optimisés. De plus, nous constatons que l'évolution des courbes de la méthode différentielle et de la méthode par torseur est comparable. En effet, la différence entre la courbe exacte et chacune des courbes approximatives est très similaire en valeur absolue en chacun des points analysés et ce, à 0.1 % près.

Comme nous l'avons mentionné plus tôt, l'approche par torseur possède certaines limites. En effet, il est important que l'ordre de grandeur des rotations infinitésimales soient inférieures à 5 degrés. Cette limitation est générée par l'approximation $\sin(\alpha) \approx \alpha$ qui conduit inévitablement à d'importantes déviations lorsque l'angle devient plus grand que 5 degrés [DUP91]. Nous avons effectué un essai pour illustrer ce comportement. L'essai a été réalisé en choisissant les paramètres suivants: $R_1=1.0$, $R_2=0.4$, $R_3=1.2$, $R_4=1.03923$, $\Delta R_1 = +0.05$, $\Delta R_2 = +0.05$, $\Delta R_3 = +0.05$, $\Delta R_4 = +0.05$ pour des angles $\theta = 0$ et $\phi = 90$ degrés. La figure 5.5 illustre les résultats obtenus.

Nous constatons des déviations importantes engendrées en certains points des courbes d'approximation des erreurs. Les déviations entre les méthodes d'approximation (méthode différentielle et méthode par torseur) et la solution exacte sont importantes dans la région où la pente est nulle ($df(x)/dx = 0$). Nous obtenons jusqu'à 15% d'erreur sur la solution exacte en ces points. Toutefois, nous voyons que la courbe générée par la méthode différentielle et la courbe générée par la méthode employant les torseurs donnent des résultats similaires malgré la déviation importante vis-à-vis la solution exacte.

Comparaison de la méthode d'analyse différentielle, de la méthode d'analyse vectorielle par rapport à la solution exacte

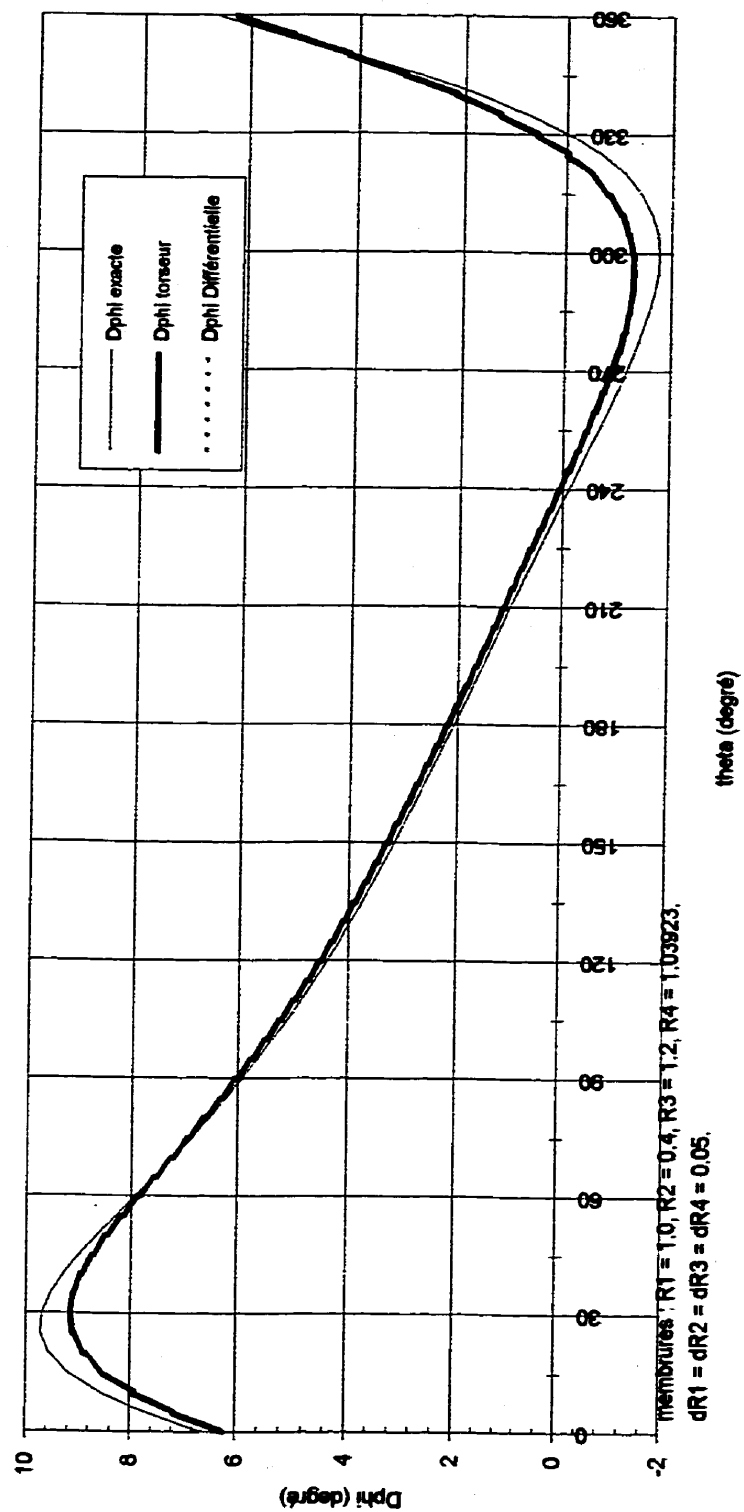


Figure 5.5 Erreurs dépassant les cinq degrés

Conclusion

Les résultats expérimentaux supportent la validité du modèle d'analyse proposé dans ce mémoire. En effet, il a été possible d'obtenir des résultats très similaires à ceux obtenus par l'approche différentielle. Nous constatons d'importantes déviations dans certaines régions des courbes obtenues lorsque les conditions choisies sont en dehors des limites d'utilisation prescrites. Toutefois, ces déviations ne sont pas particulières à l'approche d'analyse vectorielle car elles sont également observées pour l'approche différentielle.

Dans ce chapitre, nous présentons les travaux réalisés de façon à valider la méthodologie d'analyse des erreurs proposées dans ce mémoire. Ces résultats démontrent la validité de nos modèles. Le chapitre 6.0 illustre la résolution d'un cas pratique de manière à démontrer l'utilité de la méthode d'analyse vectorielle des erreurs mécaniques et des jeux sur des modèles d'ingénierie.

6.0 ANALYSE D'UN CAS PRATIQUE

Ce chapitre présente les travaux réalisés au cours de l'analyse du variateur mécanique de vitesse. Ce mécanisme est illustré à la figure 6.1 et son schéma cinématique, à la figure 6.2. Nous présentons d'abord à la section 6.1 l'étude préliminaire qui nous a permis de repérer les boucles fonctionnelles. Compte tenu que l'objectif de la présente étude est de démontrer l'utilité et la validité de la méthode proposée, nous n'analysons qu'une seule des boucles fonctionnelles. En effet, l'analyse de chacun des autres cycles peut être effectuée d'une façon similaire. Nous présentons à la section 6.2 le développement des équations vectorielles nécessaires à la recherche de la solution numérique. Nous évaluons grâce à ces équations, l'effet de chacune des erreurs sur le "backlash" de l'engrenage conique. Nous présentons à la section 6.3 les résultats numériques de l'étude.

6.1 Étude préliminaire du mécanisme

L'étude préliminaire débute par la compréhension du variateur mécanique de vitesse présenté à la figure 6.1 à partir de son schéma cinématique illustré à la figure 6.2. Rappelons que le schéma cinématique permet de définir la fonction principale du mécanisme et permet de mettre en évidence les composantes fonctionnelles qui lui sont essentielles.

Le variateur mécanique permet de moduler la vitesse de rotation de l'arbre de sortie par rapport à la vitesse de rotation de l'arbre d'entrée. Le contrôle du rapport des vitesses est réalisé grâce à un système de roulement sans glissement de trois rouleaux coniques, sur un anneau pouvant se déplacer longitudinalement selon l'axe des arbres de transmission. Les rouleaux coniques sont entraînés par le planétaire d'entrée et leur rotation est transmise à la couronne, tel qu'illustré sur le schéma cinématique. La position longitudinale de l'anneau module le rapport des vitesses. En effet, cette

position détermine le rayon des sections des rouleaux coniques sur lesquelles le roulement s'effectue. De cette façon, les vitesses sont transformées grâce aux relations suivantes :

$$\omega_r \times r = \omega_e \times R$$

$$\text{et, } \omega_s = g(\omega_r)$$

$$\text{et, } \omega_s = h(\omega_e, d)$$

où,

ω_r	:	vitesse angulaire des rouleaux coniques;
ω_e	:	vitesse angulaire de l'arbre d'entrée;
ω_s	:	vitesse angulaire de l'arbre de sortie;
\times	:	opérateur de multiplication scalaire;
R	:	rayon du planétaire;
r	:	rayon des rouleaux de la section en roulement sur l'anneau;
d	:	position longitudinale de l'anneau.

Remarquons, qu'un tel système permet d'arrêter ou d'inverser le sens de rotation de l'arbre de sortie par rapport à l'arbre d'entrée. En effet, il existe un point mort où la vitesse d'entraînement de l'arbre d'entrée est contrebalancé par la vitesse relative des rouleaux par rapport à ce même arbre. Dans cette situation, nous observons une vitesse angulaire nulle pour l'arbre de sortie. Nous pouvons également obtenir une inversion du sens de rotation de l'arbre de sortie.

Les engrenages coniques sont essentiels pour le fonctionnement du mécanisme. Ils sont également difficiles à construire et à assembler. Nous avons donc choisi le "backlash" de ces engrenages comme condition fonctionnelle à contrôler.

Le diagramme des liaisons proposé à la figure 6.3, compte quatre cycles

fonctionnels et cinq solides formant le mécanisme. Ces solides sont les composantes, précédemment identifiées grâce au schéma cinématique. Ils sont essentiels à la fonction du variateur mécanique de vitesse. Remarquons que deux des trois rouleaux coniques ont été négligés dans le diagramme des liaisons. En effet, l'analyse d'un seul de ces trois rouleaux permet de déduire les relations mathématiques agissant sur le "backlash". Dans le diagramme des liaisons proposé, il existe deux flux fonctionnels affectant la condition choisie. Seul le flux du cycle 1 est analysé car l'objectif de ce chapitre est de démontrer la méthodologie. L'analyse des autres cycles s'effectue d'une manière similaire.

Pour déterminer les tolérances et les jeux agissant sur la condition fonctionnelle, il est nécessaire d'établir le diagramme explosé des liaisons tel qu'illustré à la figure 6.4. Nous déterminons grâce à ce diagramme les tolérances et les jeux associées au cycle 1 et agissant sur le "backlash". Nous présentons à l'annexe III l'ensemble des torseurs modélisant ces jeux et ces tolérances.

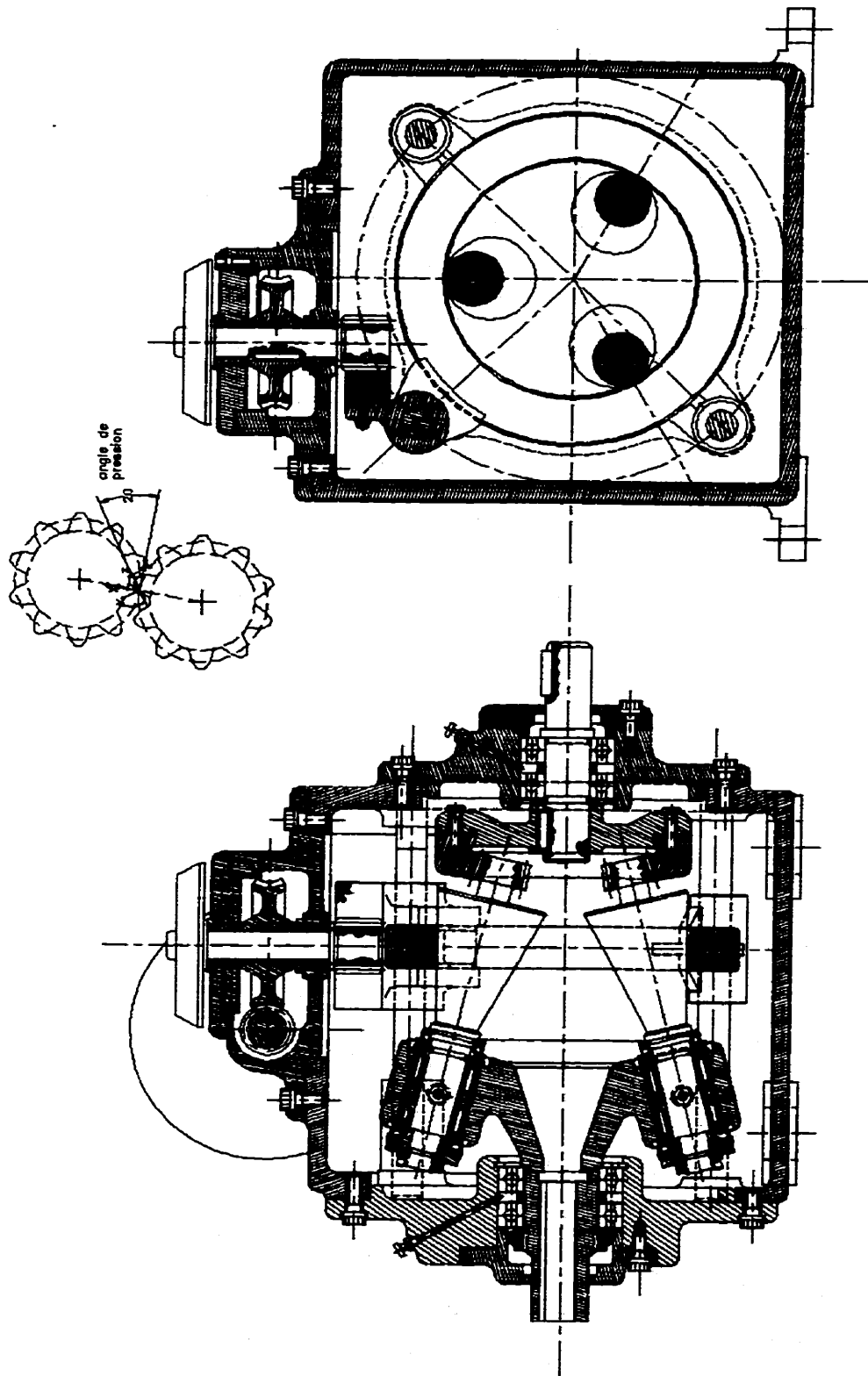


Figure 6.1 Aperçu du variateur mécanique de vitesse

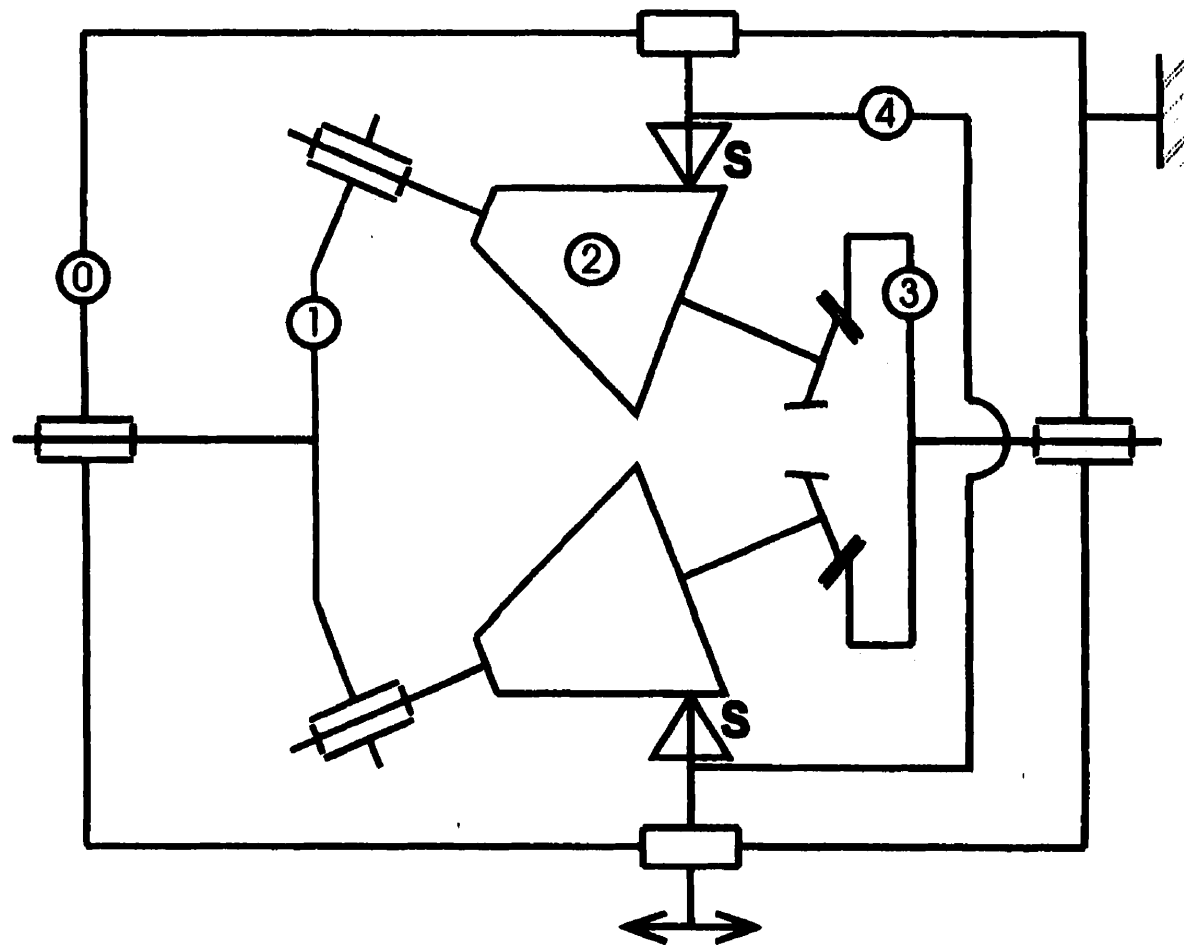


Figure 6.2 Schéma cinématique du variateur mécanique de vitesse

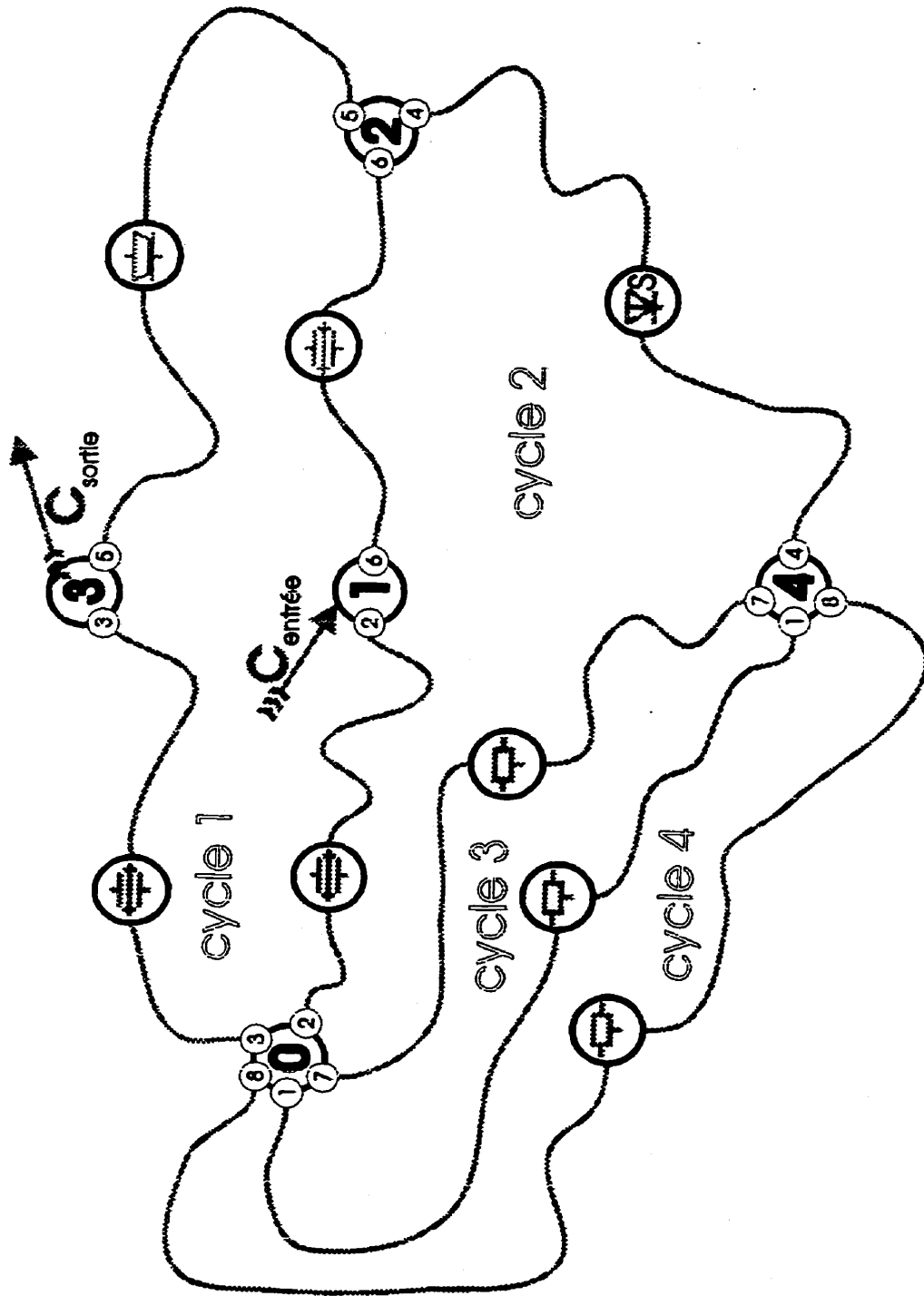


Figure 6.3 Diagramme des liaisons du variateur mécanique de vitesse

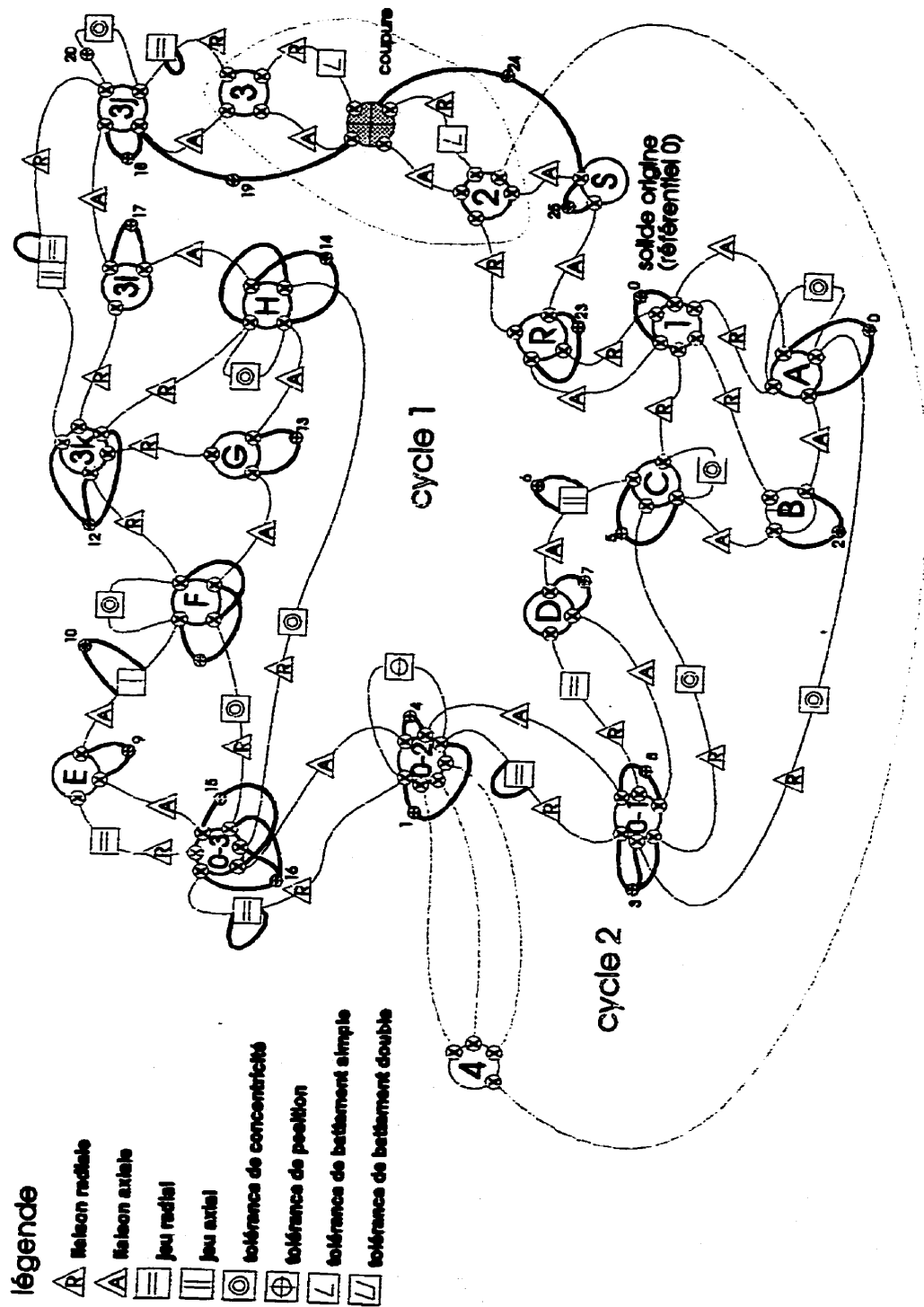


Figure 6.4 Diagramme explosé des liaisons du variateur mécanique de vitesse

6.2 Développements symboliques

Nous présentons dans cette section, les développements mathématiques d'accumulation des torseurs de tolérances géométriques, de tolérances dimensionnelles et de jeux. Ces développements permettront d'analyser les demi-boucles fonctionnelles du cycle 1 relatives au "backlash" de l'engrenage conique.

6.2.1 Torseurs d'erreurs et de jeux

Nous avons établi la liste des torseurs d'erreurs et des torseurs de jeux nécessaires à une analyse adéquate du variateur mécanique de vitesse. L'annexe III contient la description détaillée des torseurs utilisés au cours de l'analyse du "backlash" de l'engrenage conique. Ces torseurs sont définis et positionnés sur les plans fournis à l'annexe I. Remarquons que chacun des vecteurs d'erreurs est orienté dans le sens du parcours de sa demi-boucle soit: de l'origine vers la coupure. Nous présentons dans cette sous-section, les résultats vectoriels des manipulations mathématiques effectuées sur les torseurs. Précisons que certaines tolérances géométriques et certains jeux sont négligés au cours de l'étude. En effet, nous cherchons à simplifier l'analyse compte tenu de l'objectif qui est d'illustrer la faisabilité pratique du calcul vectoriel sur les tolérances dimensionnelles et géométriques ainsi que sur les jeux. Nous considérons que certains jeux du mécanisme ont un effet négligeable sur le résultat final.

6.2.1.1 Torseurs d'erreurs et de jeux pour la demi-boucle 1 du cycle 1

Voici la liste des vecteurs d'erreurs et des vecteurs de jeux associés à la demi-boucle 1. Chacun de ces vecteurs est identifié conformément au référentiel global défini par convention pour le variateur mécanique de vitesse tel qu'illustré sur les plans fournis à l'annexe I.

Vecteurs d'erreurs dimensionnelles:

$$\begin{aligned}
 \{T3\} & : V3 = (-X_3)u + (0)v + (0)w \\
 \{T4\} & : V4 = (-X_4)u + (0)v + (0)w \\
 \{T5\} & : V5 = (-X_5)u + (0)v + (0)w \\
 \{T6\} & : V6 = (-X_6)u + (0)v + (0)w \\
 \{T8\} & : V8 = (X_8)u + (0)v + (0)w \\
 \{T9\} & : V9 = (X_9)u + (0)v + (0)w \\
 \{T11\} & : V11 = (X_{11})u + (0)v + (0)w \\
 \{T13\} & : V13 = (-X_{13})u + (0)v + (0)w \\
 \{T15\} & : V15 = (-X_{15})u + (0)v + (0)w \\
 \{T16\} & : V16 = (-X_{16})u + (0)v + (0)w \\
 \{T17\} & : V17 = (-X_{17})u + (0)v + (0)w \\
 \{T18\} & : V18 = (-X_{18})u + (0)v + (0)w \\
 \{T19\} & : V19 = (-X_{19})u + (0)v + (0)w \\
 \{T21\} & : V21 = (X_{21})u + (0)v + (0)w
 \end{aligned}$$

Vecteurs d'erreurs géométriques:

$$\begin{aligned}
 \{Tgéo1\} & : Vg1 = (-\gamma_{g1}dy_{g1})u + (Y_{g1} + \gamma_{g1}dx_{g1})v + (Z_{g1} - \beta_{g1}dx_{g1})w \\
 \{Tgéo3\} & : Vg3 = (-\gamma_{g3}dy_{g3})u + (Y_{g3} + \gamma_{g3}dx_{g3})v + (Z_{g3} - \beta_{g3}dx_{g3})w \\
 \{Tgéo12\} & : Vg12 = (-\gamma_{g12}dy_{g12})u + (Y_{g12} + \gamma_{g12}dx_{g12})v + (Z_{g12} - \beta_{g12}dx_{g12})w \\
 \{Tgéo20\} & : Vg20 = (-\gamma_{g20}dy_{g20})u + (Y_{g20} + \gamma_{g20}dx_{g20})v + (Z_{g20} - \beta_{g20}dx_{g20})w \\
 \{Tgéo16\} & : Vg16 = (-\gamma_{g16}dy_{g16})u + (Y_{g16} + \gamma_{g16}dx_{g16})v + (Z_{g16} - \beta_{g16}dx_{g16})w \\
 \{Tgéo21\} & : Vg21 = (-\gamma_{g21}dy_{g21})u + (Y_{g21} + \gamma_{g21}dx_{g21})v + (Z_{g21} - \beta_{g21}dx_{g21})w
 \end{aligned}$$

Les composantes "dx" et "dy" représentent les contributions des dimensions nominales (ou moyennes) sur l'accumulation des erreurs grâce à l'application des opérateurs TRANSPORT et Changement de Référentiel (CR). Elles se définissent comme suit:

$$\begin{aligned}
 dx_{g1} & = (-e2/2 + L9) + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - L19 + L21 - L100 \\
 dy_{g1} & = r3 \\
 dx_{g3} & = (-e1/2 + L9) + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - L19 + L21 - L100
 \end{aligned}$$

$$\begin{aligned}
dy_{g3} &= r3 \\
dx_{g20} &= e4/2 + L21 - L100 \\
dy_{g20} &= r3 \\
dx_{g16} &= e3/2 + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - L19 + L21 - L100 \\
dy_{g16} &= r3 \\
dx_{g12} &= (e12/2) - L19 + L21 - L100 \\
dy_{g12} &= r3 \\
dx_{g21} &= (e21/2) + L20 + L21 - L100
\end{aligned}$$

Les symboles " α ", " β " et " γ " sont les composantes des vecteurs "résultants" des torseurs.

Vecteurs de jeux:

$$\begin{aligned}
\{\text{Tjeu1}\} &: \mathbf{Vj1} = (-X_{\text{jeu1}})\mathbf{u} + (0)\mathbf{v} + (0)\mathbf{w} \\
\{\text{Tjeu2}\} &: \mathbf{Vj2} = (-X_{\text{jeu2}})\mathbf{u} + (0)\mathbf{v} + (0)\mathbf{w}
\end{aligned}$$

6.2.1.2 Torseurs d'erreurs et de jeux pour la demi-boucle 2 du cycle 1

Voici la liste des vecteurs d'erreurs et des vecteurs de jeux associés à la demi-boucle 2. Chacun de ces vecteurs est identifié conformément au référentiel global défini par convention pour le variateur mécanique de vitesse tel qu'illustré sur les plans de l'annexe I. Le symbole λ est introduit afin de simplifier l'écriture de ces équations. Il se définit de la façon suivante:

$$\lambda = (\text{ang2} - 90) \text{ degrés}$$

Vecteurs d'erreurs dimensionnelles:

$$\begin{aligned}
\{\text{T51}\} &: \mathbf{V51} = (X_{51}\cos(\lambda))\mathbf{u} + (-X_{51}\sin(\lambda))\mathbf{v} + (0)\mathbf{w} \\
\{\text{T54}\} &: \mathbf{V54} = (X_{54}\cos(\lambda))\mathbf{u} + (-X_{54}\sin(\lambda))\mathbf{v} + (0)\mathbf{w} \\
\{\text{T55}\} &: \mathbf{V55} = (X_{55}\cos(\lambda))\mathbf{u} + (-X_{55}\sin(\lambda))\mathbf{v} + (0)\mathbf{w}
\end{aligned}$$

Vecteurs d'erreurs géométriques:

$$\begin{aligned}
 \{T_{géo26}\} : V_{g26} &= (-\gamma_{g26} dy_{g26} \cos(\lambda) - (Y_{g26} + \gamma_{g26} dx_{g26}) \sin(\lambda))u + \\
 &\quad (-\gamma_{g26} dy_{g26} \sin(\lambda) + (Y_{g26} - \gamma_{g26} dx_{g26}) \cos(\lambda))v + \\
 &\quad (Z_{g26} - \beta_{g26} dx_{g26})w \\
 \{T_{géo27}\} : V_{g27} &= (-\gamma_{g27} dy_{g27} \cos(\lambda) - (Y_{g27} + \gamma_{g27} dx_{g27}) \sin(\lambda))u + \\
 &\quad (-\gamma_{g27} dy_{g27} \sin(\lambda) + (Y_{g27} - \gamma_{g27} dx_{g27}) \cos(\lambda))v + \\
 &\quad (Z_{g27} - \beta_{g27} dx_{g27})w
 \end{aligned}$$

La contribution des dimensions nominales (ou moyennes) sur l'accumulation des erreurs pour la demi-boucle 2, est introduit par les symboles "dx" et "dy" ci-dessous:

$$\begin{aligned}
 dx_{g26} &= e8/2 + L51 + L54 + L55 \\
 dy_{g26} &= r2 \\
 dx_{g27} &= 0 \\
 dy_{g27} &= r2
 \end{aligned}$$

6.2.1.3 Backlash de l'engrenage conique

Le "backlash" de l'engrenage conique du variateur mécanique de vitesse correspond à la contrainte fonctionnelle qui doit être respectée pour assurer le bon fonctionnement du mécanisme. Ce vecteur est défini dans le référentiel global du variateur mécanique de vitesse tel que défini sur les plans de l'annexe I. Le vecteur **B** représentant le "backlash" est illustré sur le schéma de la figure 6.5.

Par hypothèse, seules les composantes des moments des torseurs résultants suivants les vecteurs unitaires **u** et **v** sont analysés. Elles correspondent aux contraintes technologiques les plus sévères.

$$B = (Back1)u + (Back2)v + (Back3)w$$

On notera que l'analyse des résultantes peut être utile lorsque nous voulons contrôler la répartition de la pression sur la denture.

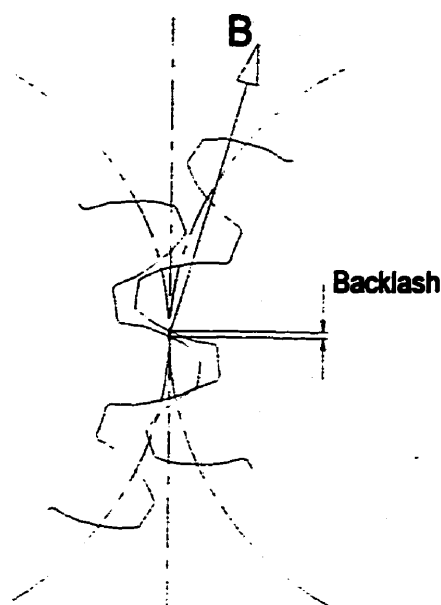


Figure 6.5 Représentation du vecteur **B** sur les dents de l'engrenage

Les vecteurs "moments" énumérés ci-dessus sont les résultats des opérations de TRANSPORT et de Changement de Référentiel (CR) effectuées sur les torseurs correspondants. Ces opérations mathématiques demandent la connaissance des dimensions nominales (ou moyennes) du mécanisme. Nous donnons ces dimensions dans les tableaux 6.3 et 6.4. Ces dimensions respectent la condition de fermeture géométrique du cycle 1. Nous présentons aux annexes I et III l'ensemble de ces dimensions ainsi qu'un aperçu de leur utilisation au cours des TRANSPORTS et des Changements de Référentiel (CR).

6.2.2 Accumulation vectorielle des erreurs

Nous savons qu'il est possible d'évaluer l'influence sur le "backlash" **B** de l'ensemble des dispersions mécaniques du cycle fonctionnel 1. Cette évaluation est réalisée grâce à la relation mathématique du chapitre 4.0:

$$\begin{aligned}
 & \left(\sum_{q=1..s} \{E^{(1)}\}_{regM} - \sum_{r=1..t} \{E^{(2)}\}_{regM} \right) \otimes \{C\}_{regM} \\
 & \leq \\
 & \{C\}_{regM} \\
 & \mp \\
 & \left(\sum_{i=1..n} \{U^{(1)}\}_{regM} \otimes \{T^{(1)}\}_{regM} - \sum_{j=1..m} \{U^{(2)}\}_{regM} \otimes \{T^{(2)}\}_{regM} \right) \otimes \{C\}_{regM}
 \end{aligned}$$

Cette inégalité représente la relation entre les torseurs d'erreurs mécaniques, les torseurs de jeux et le torseur de la condition (ou contrainte) fonctionnelle. Elle peut être interprétée de manière à mettre en évidence les paramètres particuliers au problème du variateur mécanique de vitesse dans le cas où nous cherchons à vérifier une condition fonctionnelle. Cette inégalité a donc été transformée pour représenter les vecteurs obtenus précédemment.

$$\begin{aligned}
 & ((\Delta E_1 - \Delta E_2) + ((\Delta J_1 \otimes T_1) - (\Delta J_2 \otimes T_2))) \otimes B \\
 & \leq \\
 & B
 \end{aligned}$$

où,

- | | | |
|--------------|---|--|
| ΔE_1 | : | vecteur représentant la somme des vecteurs d'erreurs générés sur la demi-boucle 1; |
| ΔE_2 | : | vecteur représentant la somme des vecteurs d'erreurs générés sur la demi-boucle 2; |
| ΔJ_1 | : | vecteur représentant l'ensemble des jeux agissant sur la demi-boucle 1; |

- ΔJ_2 : vecteur représentant l'ensemble des jeux agissant sur la demi-boucle 2;
 T_1 : vecteur d'inter-effort de chacune des liaisons où existe un jeu;
 T_2 : vecteur d'inter-effort de chacune des liaisons où existe un jeu;
 B : vecteur dispersion du "backlash" des dents de l'engrenage conique.

L'accumulation des erreurs mécaniques locales est obtenue par l'addition des vecteurs de chacune des demi-boucles. Les deux vecteurs résultants sont projetés sur le vecteur de la contrainte fonctionnelle. Finalement, une différence des vecteurs d'erreurs projetés sur B est effectuée. Le résultat correspond à l'erreur mécanique introduite par les tolérances dimensionnelles et les tolérances géométriques associées au cycle 1. La projection des vecteurs d'erreurs résultants sur la contrainte fonctionnelle est illustrée à la figure 6.6.

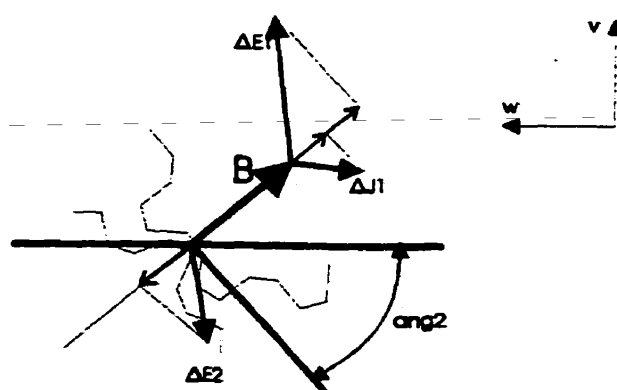


Figure 6.6 Projection des erreurs et des jeux sur la contrainte fonctionnelle B

Comme nous l'avons déjà mentionné, il est possible d'évaluer l'ensemble des combinaisons d'erreurs pour un mécanisme quelconque. Toutefois, l'intérêt pratique est porté sur l'obtention du minimum global et du maximum global des tolérances. En effet, nous cherchons à connaître l'enveloppe limite des tolérances du mécanisme. Les vecteurs ΔE_1 et ΔE_2 sont évalués de manière à connaître cet enveloppe limite.

$$\Delta E_1 = \text{Maximum/Minimum}\{V3 + V4 + V5 + V6 + V8 + V9 + V11 + V13 + V15 + V16 + V17 + V18 + V19 + V21 + Vg1 + Vg3 + Vg12 + Vg20 + Vg16 + Vg21\}$$

$$\Delta E_2 = \text{Maximum/Minimum}\{V51 + V54 + V55 + Vg26 + Vg27\}$$

L'évaluation de la contribution des jeux résultants à la coupure s'effectue différemment. Contrairement aux défauts géométriques, la prise en compte des jeux à l'intérieur des cycles fonctionnels demande une bonne connaissance des forces agissant dans le mécanisme. Ces forces affectent la reprise des jeux et leur organisation interne. De ce fait, les jeux ne peuvent agir que selon la direction des vecteurs d'inter-efforts de chacune des liaisons. Voici l'équation d'accumulation des vecteurs de jeux:

$$\Delta J = (\Delta J_1 \otimes T_1) \otimes B - (\Delta J_2 \otimes T_2) \otimes B$$

où, ΔJ_1 : maximum/minimum $\{Vj1 + Vj2\}$;
 ΔJ_2 : $\{0\}$;
 T_1 : vecteur d'inter-effort associé au jeu 1;
 T_2 : vecteur d'inter-effort associé au jeu 2;
 B : vecteur du "backlash".

Les jeux associés sont alignés avec les inter-efforts T_1 et T_2 . Ainsi, nous

pouvons réécrire l'équation précédente de la façon suivante:

$$\Delta J = \Delta J_1 \otimes B - \Delta J_2 \otimes B$$

À partir de ces équations, nous reformulons l'expression mathématique de propagation des erreurs et des jeux:

$$\begin{aligned} & \{\text{Maximum/Minimum}\{V3 + V4 + V5 + V6 + V8 + V9 + V11 + V13 + V15 + V16 \\ & + V17 + V18 + V19 + V21 + Vg1 + Vg3 + Vg12 + Vg20 + Vg16 + Vg21\}\} \otimes B \\ & - \\ & \{\text{Maximum/Minimum}\{V51 + V54 + V55 + Vg26 + Vg27\}\} \otimes B \\ & + \\ & \{\text{Maximum/Minimum}\{Vj1 + Vj2\}\} \otimes B \\ & \leq \\ & B \end{aligned}$$

Évidemment, l'introduction des tolérances géométriques et des tolérances dimensionnelles dans l'équation de la propagation des erreurs nécessite la mise en place d'un système de contraintes. Les inéquations ci-dessous représentent le système de contraintes qui doit être respecté au cours de la recherche d'une solution.

Tolérance géométrique (12)

$$4(Y_{g12})^2 + 4(Z_{g12})^2 + (\beta_{g12}e12)^2 + (\gamma_{g12}e12)^2 \leq t_6^2$$

Tolérance géométrique (20)

$$4(Y_{g20})^2 + 4(Z_{g20})^2 + (\beta_{g20}e4)^2 + (\gamma_{g20}e4)^2 \leq t_5^2$$

Tolérance géométrique (21)

$$4(Y_{g21})^2 + 4(Z_{g21})^2 + (\beta_{g21}e21)^2 + (\gamma_{g21}e21)^2 \leq t_4^2$$

Tolérance géométrique (26)

$$4(Y_{g26})^2 + 4(Z_{g26})^2 + (\beta_{g26}e8)^2 + (\gamma_{g26}e8)^2 \leq t_1^2$$

Tolérance géométrique (3)

$$4(Y_{g3})^2 + 4(Z_{g3})^2 + (\beta_{g3}e1)^2 + (\gamma_{g3}e1)^2 \leq t_9^2$$

Tolérance géométrique (1)

$$4(Y_{g1})^2 + 4(Z_{g1})^2 + (\beta_{g1}e2)^2 + (\gamma_{g1}e2)^2 \leq t_8^2$$

Tolérance géométrique (16)

$$4(Y_{g16})^2 + 4(Z_{g16})^2 + (\beta_{g16}e3)^2 + (\gamma_{g16}e3)^2 \leq t_7^2$$

Tolérance géométrique (27)

$$4(Y_{g27})^2 + 4(Z_{g27})^2 + (\beta_{g27}e9)^2 + (\gamma_{g27}e9)^2 \leq t_3^2$$

Tolérances dimensionnelles

X_3	\leq	Δx_3
X_4	\leq	Δx_4
X_5	\leq	Δx_5
X_6	\leq	Δx_6
X_8	\leq	Δx_8
X_9	\leq	Δx_9
X_{11}	\leq	Δx_{11}
X_{13}	\leq	Δx_{13}
X_{15}	\leq	Δx_{15}
X_{16}	\leq	Δx_{16}
X_{17}	\leq	Δx_{17}
X_{18}	\leq	Δx_{18}
X_{19}	\leq	Δx_{19}
X_{21}	\leq	Δx_{21}
X_{51}	\leq	Δx_{51}
X_{54}	\leq	Δx_{54}
X_{55}	\leq	Δx_{55}

Autres équations:

$$\begin{aligned}
\lambda &= (\text{ang2} - 90) \\
dx_{g26} &= e8/2 + L51 + L54 + L55 \\
dy_{g26} &= r2 \\
dx_{g27} &= 0 \\
dy_{g27} &= r2 \\
dx_{g1} &= -e2/2 + L9 + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - L19 + L21 - L100 \\
dy_{g1} &= r3 \\
dx_{g3} &= -e1/2 + L9 + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - L19 + L21 - L100 \\
dy_{g3} &= r3 \\
dx_{g20} &= e4/2 + L21 - L100 \\
dy_{g20} &= r3 \\
dx_{g16} &= e3/2 + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - L19 + L21 - L100 \\
dy_{g16} &= r3 \\
dx_{g12} &= e12/2 - L19 + L21 - L100 \\
dy_{g12} &= r3 \\
dx_{g21} &= e21/2 + L20 + L21 - L100 \\
dy_{g21} &= r3
\end{aligned}$$

Toutes les contraintes associées au cycle 1 ont été définies. Nous allons maintenant réécrire le système d'équations et d'inéquations de manière à résumer le problème d'optimisation. Chacun des vecteurs est représenté par ses variables.

Inéquation principale:

$$\begin{aligned}
& \left[-\{(-X_3)u\} - \{(-X_4)u\} - \{(-X_5)u\} - \{(-X_6)u\} + \{(X_8)u\} + \{(X_9)u\} + \{(X_{11})u\} - \{(-X_{13})u\} \right. \\
& - \{(-X_{15})u\} - \{(-X_{16})u\} - \{(-X_{17})u\} - \{(-X_{18})u\} - \{(-X_{19})u\} + \{(X_{21})u\} + \{(-\gamma_{g1}dy_{g1})u + \\
& (Y_{g1} + \gamma_{g1}dx_{g1})v + (Z_{g1} - \beta_{g1}dx_{g1})w\} + \{(-\gamma_{g3}dy_{g3})u + (Y_{g3} + \gamma_{g3}dx_{g3})v + (Z_{g3} - \\
& \beta_{g3}dx_{g3})w\} + \{(-\gamma_{g12}dy_{g12})u + (Y_{g12} + \gamma_{g12}dx_{g12})v + (Z_{g12} - \beta_{g12}dx_{g12})w\} + \{(-\gamma_{g20}dy_{g20})u \\
& + (Y_{g20} + \gamma_{g20}dx_{g20})v + (Z_{g20} - \beta_{g20}dx_{g20})w\} + \{(-\gamma_{g16}dy_{g16})u + (Y_{g16} + \gamma_{g16}dx_{g16})v + \\
& (Z_{g16} - \beta_{g16}dx_{g16})w\} + \{(-\gamma_{g21}dy_{g21})u + (Y_{g21} + \gamma_{g21}dx_{g21})v + (Z_{g21} - \beta_{g21}dx_{g21})w\} \left. \right] \\
& \cdot \{(\text{Back1})u + (\text{Back2})v + (\text{Back3})w\} / (\{(\text{Back1})u + (\text{Back2})v + (\text{Back3})w\} \cdot \{(\text{Back1})u \\
& + (\text{Back2})v + (\text{Back3})w\}) \{(\text{Back1})u + (\text{Back2})v + (\text{Back3})w\}
\end{aligned}$$

-

$$\left[\{(X_{s1}\cos(\lambda))u + (-X_{s1}\sin(\lambda))v\} + \{(X_{s4}\cos(\lambda))u + (-X_{s4}\sin(\lambda))v\} + \{(X_{s5}\cos(\lambda))u + \right.$$

$$\begin{aligned}
& (-X_{55}\sin(\lambda))v\} + \{(-\gamma_{g26}dy_{g26}\cos(\lambda) - (Y_{g26} + \gamma_{g26}dx_{g26})\sin(\lambda))u + \\
& (-\gamma_{g26}dy_{g26}\sin(\lambda) + (Y_{g26} - \gamma_{g26}dx_{g26})\cos(\lambda))v + (Z_{g26} - \beta_{g26}dx_{g26})w\} + \{(-\gamma_{g27}dy_{g27}\cos(\lambda) \\
& - (Y_{g27} + \gamma_{g27}dx_{g27})\sin(\lambda))u + (-\gamma_{g27}dy_{g27}\sin(\lambda) + (Y_{g27} - \gamma_{g27}dx_{g27})\cos(\lambda))v + \\
& (Z_{g27} - \beta_{g27}dx_{g27})w\} \Big] \cdot \{(Back1)u + (Back2)v + (Back3)w\} / (\{(Back1)u + (Back2)v + \\
& (Back3)w\} \cdot \{(Back1)u + (Back2)v + (Back3)w\}) \\
& \{(Back1)u + (Back2)v + (Back3)w\} \\
& + \\
& \{(-X_{jeu1} - X_{jeu2})u\} \cdot \{(Back1)u + (Back2)v + (Back3)w\} / (\{(Back1)u + (Back2)v + \\
& (Back3)w\} \cdot \{(Back1)u + (Back2)v + (Back3)w\}) \\
& \{(Back1)u + (Back2)v + (Back3)w\} \\
& \leq \\
& (Back1)u + (Back2)v + (Back3)w
\end{aligned}$$

Contraintes associées:

$$\begin{aligned}
4(Y_{g12})^2 + 4(Z_{g12})^2 + (\beta_{g12}e12)^2 + (\gamma_{g12}e12)^2 & \leq t_6^2 \\
4(Y_{g20})^2 + 4(Z_{g20})^2 + (\beta_{g20}e4)^2 + (\gamma_{g20}e4)^2 & \leq t_5^2 \\
4(Y_{g21})^2 + 4(Z_{g21})^2 + (\beta_{g21}e21)^2 + (\gamma_{g21}e21)^2 & \leq t_4^2 \\
4(Y_{g26})^2 + 4(Z_{g26})^2 + (\beta_{g26}e8)^2 + (\gamma_{g26}e8)^2 & \leq t_1^2 \\
4(Y_{g3})^2 + 4(Z_{g3})^2 + (\beta_{g3}e1)^2 + (\gamma_{g3}e1)^2 & \leq t_9^2 \\
4(Y_{g1})^2 + 4(Z_{g1})^2 + (\beta_{g1}e2)^2 + (\gamma_{g1}e2)^2 & \leq t_8^2 \\
4(Y_{g16})^2 + 4(Z_{g16})^2 + (\beta_{g16}e3)^2 + (\gamma_{g16}e3)^2 & \leq t_7^2 \\
4(Y_{g27})^2 + 4(Z_{g27})^2 + (\beta_{g27}e9)^2 + (\gamma_{g27}e9)^2 & \leq t_3^2
\end{aligned}$$

$$\begin{aligned}
X_3 & \leq \Delta x_3 \\
X_4 & \leq \Delta x_4 \\
X_5 & \leq \Delta x_5 \\
X_6 & \leq \Delta x_6 \\
X_8 & \leq \Delta x_8 \\
X_9 & \leq \Delta x_9 \\
X_{11} & \leq \Delta x_{11} \\
X_{13} & \leq \Delta x_{13} \\
X_{15} & \leq \Delta x_{15} \\
X_{16} & \leq \Delta x_{16} \\
X_{17} & \leq \Delta x_{17} \\
X_{18} & \leq \Delta x_{18} \\
X_{19} & \leq \Delta x_{19} \\
X_{21} & \leq \Delta x_{21} \\
X_{51} & \leq \Delta x_{51}
\end{aligned}$$

$$\begin{aligned}
X_{54} &\leq \Delta x_{54} \\
X_{55} &\leq \Delta x_{55} \\
\lambda &= (\text{ang2} - 90) \\
dx_{g26} &= e8/2 + L51 + L54 + L55 \\
dy_{g26} &= r2 \\
dx_{g27} &= 0 \\
dy_{g27} &= r2 \\
dx_{g1} &= -e2/2 + L9 + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - L19 + L21 - L100 \\
dy_{g1} &= r3 \\
dx_{g3} &= -e1/2 + L9 + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - L19 + L21 - L100 \\
dy_{g3} &= r3 \\
dx_{g20} &= e4/2 + L21 - L100 \\
dy_{g20} &= r3 \\
dx_{g16} &= e3/2 + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - L19 + L21 - L100 \\
dy_{g16} &= r3 \\
dx_{g12} &= e12/2 - L19 + L21 - L100 \\
dy_{g12} &= r3 \\
dx_{g21} &= e21/2 + L20 + L21 - L100 \\
dy_{g21} &= r3
\end{aligned}$$

Les variables contenues dans le système d'équations et d'inéquations sont définies aux annexes I et III.

6.3 Résultat de l'étude

Cette section présente les résultats numériques de l'étude du variateur mécanique de vitesse. Nous démontrons qu'il est possible de générer une information utile aux concepteurs en employant la méthode vectorielle d'analyse. Comme exemple, nous évaluons les limites des jeux 1 et 2 et ce, en respectant la contrainte fonctionnelle. Évidemment, différents types d'analyse peuvent être réalisés avec la méthode proposée. Nous pouvons fixer les jeux et la contrainte fonctionnelle afin d'établir un tolérancement convenable. Nous pouvons également balancer les tolérances et les jeux en employant différentes techniques d'optimisation ou de pondération telle que la "logique floue".

Les tableaux suivants présentent l'ensemble des paramètres numériques que nous

proposons afin de résoudre le problème du variateur mécanique de vitesse. Remarquons que les tolérances choisies correspondent à celles d'un mécanisme de précision. En effet, elles représentent des erreurs qui sont à la limite de ce qui peut être réalisées en pratique.

Tableau 6.1 Valeurs numériques des tolérances associées à la demi-boucle 1

Demi-Boucle 1 Tolérances			
variables	valeurs numériques (mm)	variables	valeurs numériques (mm)
t_6	0.01		
t_7	0.01	t_4	0.02
t_5	0.01	t_9	0.01
t_8	0.01	Δx_3	0.01
Δx_4	0.01	Δx_5	0.01
Δx_6	0.01	Δx_8	0.01
Δx_9	0.01	Δx_{11}	0.01
Δx_{13}	0.01	Δx_{15}	0.01
Δx_{16}	0.01	Δx_{17}	0.01
Δx_{18}	0.01	Δx_{19}	0.01
Δx_{21}	0.01		

Tableau 6.2 Valeurs numériques des tolérances associées à la demi-boucle 2

Demi-boucle 2 Tolérances			
variables	valeurs numériques (mm)	variables	valeurs numériques (mm)
t_1	0.02	t_3	0.02
Δx_{51}	0.01	Δx_{54}	0.01
Δx_{55}	0.01		

Tableau 6.3 Valeurs numériques des dimensions nominales associées à la demi-boucle 1

Demi-boucle 1 Nominaux			
variables	valeurs numériques (mm)	variables	valeurs numériques (mm)
e21	11.5	e2	8.00
e4	5.00	e3	6.00
e12	27.00	e1	4.50
L3	12.00	L4	5.95
L5	12.00	L6	10.00
L8	25.00	L9	225.00
L11	30.00	L13	5.00
L15	12.00	L16	5.95
L17	12.00	L18	5.00
L19	20.00	L20	10.00
L21	70.53904668	L100	112.24177614
jeu1	0.05	r3	46.58756432
jeu2	0.05		

Tableau 6.4 Valeurs numériques des dimensions nominales associées à la demi-boucle 2

Demi-boucle 2 Nominaux			
variables	valeurs numériques (mm)	variables	valeurs numériques (mm)
e9	12.00	e8	35.50
L51	5.00	L50	50.00
L53	50.00	L54	8.00
L55	73.50	L52	193.67827472
r2	17.95		
Ang2	76 degrés	Backlash B _{mov}	0.100

Les dimensions nominales des demi-boucles 1 et 2 sont sélectionnées de manière à assurer la fermeture géométrique du cycle 1. Pour cette raison, les dimensions nominales L21, L52, r3 et L100 possèdent des valeurs numériques à plusieurs décimales.

Nous supposons par hypothèse, que le "backlash" minimum possède une valeur de 0.01 mm. De plus, nous supposons que le "backlash" maximum possède une valeur de 0.19 mm. Nous obtenons le vecteur du "backlash" moyen de la façon suivante:

$$B_{\text{moy}} = \frac{B_{\text{max}} + B_{\text{min}}}{2}$$

Nous obtenons alors,

$$B_{\text{moy}} = 0.10^{\pm 0.09} \text{ mm}$$

Finalement,

$$B_{\text{moy}} = (0.09 \sin(14 \text{ degrés}) \text{ mm})u + (0.09 \cos(14 \text{ degrés}) \text{ mm})v$$

$$B_{\text{moy}} = (0.021773 \text{ mm})u + (0.087327 \text{ mm})v$$

Compte tenu des paramètres numériques proposés dans les tableaux précédents, nous allons réécrire les contraintes associées au problème.

Contraintes associées:

$$\begin{aligned} \lambda &= -14^\circ \quad (\text{paramètre géométrique}) \\ dx_{g26} &= 17.75 + 5.00 + 8.00 + 73.50 = \mathbf{104.25 \text{ mm}} \\ dy_{g26} &= \mathbf{17.95 \text{ mm}} \\ dx_{g27} &= \mathbf{0} \\ dy_{g27} &= \mathbf{17.95 \text{ mm}} \\ dx_{g1} &= -4.00 + 225.00 + 30.00 - 5.00 - 0.05 - 12.00 - 5.95 - 12.00 - 5.00 - \\ &\quad 20.00 + 70.53904668 - 112.24177614 = \mathbf{149.29727054 \text{ mm}} \\ dy_{g1} &= \mathbf{46.58756432 \text{ mm}} \\ dx_{g3} &= -2.25 + 225.00 + 30.00 - 5.00 - 0.05 - 12.00 - 5.95 - 12.00 - 5.00 - \\ &\quad 20.00 + 70.53904668 - 112.24177614 = \mathbf{151.04727054 \text{ mm}} \end{aligned}$$

$$\begin{aligned}
dy_{g3} &= 46.58756432 \text{ mm} \\
dx_{g20} &= 2.50 + 70.53904668 - 112.24177614 = -39.20272946 \text{ mm} \\
dy_{g20} &= 46.58756432 \text{ mm} \\
dx_{g16} &= 3.00 + 30.00 - 5.00 - 0.05 - 12.00 - 5.95 - 12.00 - 5.00 - 20.00 + \\
&\quad 70.53904668 - 112.24177614 = -68.70272946 \text{ mm} \\
dy_{g16} &= 46.58756432 \text{ mm} \\
dx_{g12} &= 13.50 - 20.00 + 70.53904668 - 112.24177614 = -48.20272946 \text{ mm} \\
dy_{g12} &= 46.58756432 \text{ mm} \\
dx_{g21} &= 5.75 + 10.00 + 70.53904668 - 112.24177614 = -25.95272946 \text{ mm} \\
dy_{g21} &= 46.58756432 \text{ mm}
\end{aligned}$$

$$\begin{aligned}
4(Y_{g12})^2 + 4(Z_{g12})^2 + ((27.0)\beta_{g12})^2 + ((27.0)\gamma_{g12})^2 &\leq 1.0 \times 10^{-4} \text{ mm}^2 \\
4(Y_{g20})^2 + 4(Z_{g20})^2 + ((5.00)\beta_{g20})^2 + ((5.00)\gamma_{g20})^2 &\leq 1.0 \times 10^{-4} \text{ mm}^2 \\
4(Y_{g21})^2 + 4(Z_{g21})^2 + ((11.50)\beta_{g21})^2 + ((11.50)\gamma_{g21})^2 &\leq 4.0 \times 10^{-4} \text{ mm}^2 \\
4(Y_{g26})^2 + 4(Z_{g26})^2 + ((35.50)\beta_{g26})^2 + ((35.50)\gamma_{g26})^2 &\leq 4.0 \times 10^{-4} \text{ mm}^2 \\
4(Y_{g3})^2 + 4(Z_{g3})^2 + ((4.50)\beta_{g3})^2 + ((4.50)\gamma_{g3})^2 &\leq 1.0 \times 10^{-4} \text{ mm}^2 \\
4(Y_{g1})^2 + 4(Z_{g1})^2 + ((8.00)\beta_{g1})^2 + ((8.00)\gamma_{g1})^2 &\leq 1.0 \times 10^{-4} \text{ mm}^2 \\
4(Y_{g16})^2 + 4(Z_{g16})^2 + ((6.00)\beta_{g16})^2 + ((6.00)\gamma_{g16})^2 &\leq 1.0 \times 10^{-4} \text{ mm}^2 \\
4(Y_{g27})^2 + 4(Z_{g27})^2 + ((12.0)\beta_{g27})^2 + ((12.0)\gamma_{g27})^2 &\leq 4.0 \times 10^{-4} \text{ mm}^2
\end{aligned}$$

$$\begin{aligned}
X_3 &\leq 0.01 \text{ mm} \\
X_4 &\leq 0.01 \text{ mm} \\
X_5 &\leq 0.01 \text{ mm} \\
X_6 &\leq 0.01 \text{ mm} \\
X_8 &\leq 0.01 \text{ mm} \\
X_9 &\leq 0.01 \text{ mm} \\
X_{11} &\leq 0.01 \text{ mm} \\
X_{13} &\leq 0.01 \text{ mm} \\
X_{15} &\leq 0.01 \text{ mm} \\
X_{16} &\leq 0.01 \text{ mm} \\
X_{17} &\leq 0.01 \text{ mm} \\
X_{18} &\leq 0.01 \text{ mm} \\
X_{19} &\leq 0.01 \text{ mm} \\
X_{21} &\leq 0.01 \text{ mm} \\
X_{51} &\leq 0.01 \text{ mm} \\
X_{54} &\leq 0.01 \text{ mm} \\
X_{55} &\leq 0.01 \text{ mm}
\end{aligned}$$

Certaines hypothèses doivent être formulées pour résoudre le système d'équations et d'inéquations. En effet, il existe trop d'inconnues dans ce problème. Nous cherchons l'effet maximal de la distribution des erreurs. Nous supposons donc que les tolérances

dimensionnelles sont connues et possèdent la valeur limite 0.01 mm. Le tolérancement géométrique est choisi de manière à générer une déviation maximale au point de coupure. Deux cas peuvent alors être analysés tels qu'illustrés à la figure 6.7:

1. la déviation est longitudinale et ne possède aucune composante angulaire;

Dans ce cas, la contrainte associée peut se redéfinir ainsi,

$$(Y_{gxx})^2 + (Z_{gxx})^2 \leq (\text{tolérance } xx)^2 \text{ mm}^2$$

2. la déviation est angulaire et ne possède aucune composante longitudinale.

Dans ce cas, la contrainte associée peut se redéfinir ainsi,

$$\beta_{gxx} \leq (\text{tolérance } xx)/(\text{longueur de la zone } xx)$$

$$\gamma_{gxx} \leq (\text{tolérance } xx)/(\text{longueur de la zone } xx)$$

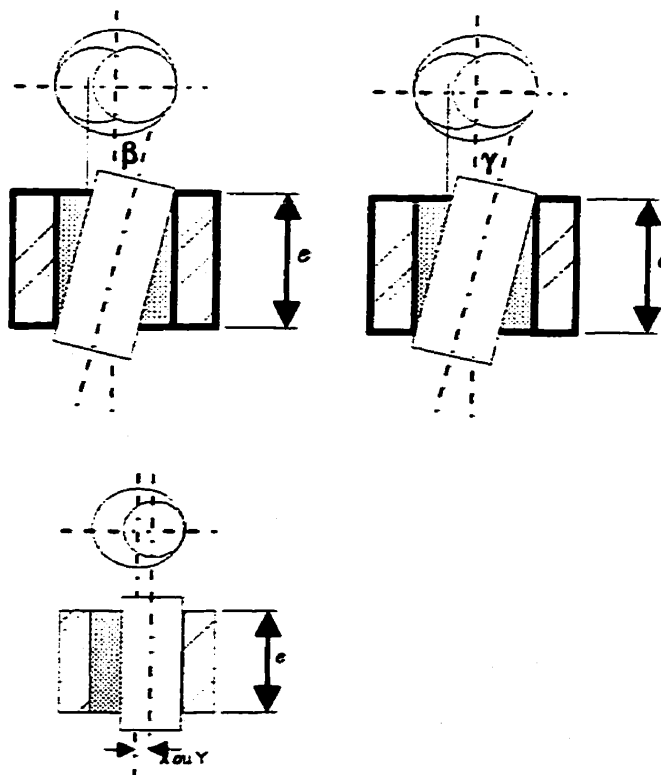


Figure 6.7 Cas extrêmes des déviations engendrées par les tolérances géométriques

Nous avons défini au chapitre 4 certaines règles permettant de simplifier les systèmes d'équations et d'inéquations. Ainsi, nous savons que lorsque le bras de levier est important, les déviations angulaires des tolérances géométriques deviennent prépondérantes vis-à-vis des déviations longitudinales. Il faut également tenir compte de la nature de chacun des contacts unissant les composantes du mécanisme. Par exemple, les déviations angulaires des tolérances géométriques (iv), (iii) et (v) des composantes respectives 11, 18 et 20 n'affectent pas la condition fonctionnelle. En effet, considérant l'importance du contact plan par rapport au contact cylindrique, nous arrivons à la conclusion que l'erreur angulaire n'agit que localement. Pour cette condition, seules les composantes longitudinales de l'erreur sont considérées au cours l'accumulation. Rappelons que l'analyse de la condition fonctionnelle choisie est réalisée sur un plan unique défini par les axes des vecteurs unitaires u et v . Les erreurs agissant selon l'axe du vecteur unitaire w sont négligées. Nous cherchons à vérifier la fonctionnalité de l'engrenage conique dans sa situation extrême c'est-à-dire, lorsque les dimensions fonctionnelles des composantes possèdent une déviation limite. Un tel problème nécessite l'accumulation des erreurs en considérant les pièces au minimum de matière admissible. Ainsi, nous allons choisir le sens de chacun des vecteurs d'erreurs de façon à ce qu'il augmente le jeu entre les dents de l'engrenage conique. Voici le système d'équations à résoudre:

Système

Inéquation vectorielle principale

$$\begin{aligned} & \left[-\{(-X_3)u\} - \{(-X_4)u\} - \{(-X_5)u\} - \{(-X_6)u\} + \{(X_8)u\} + \{(X_9)u\} + \{(X_{11})u\} - \{(-X_{13})u\} \right. \\ & - \{(-X_{15})u\} - \{(-X_{16})u\} - \{(-X_{17})u\} - \{(-X_{18})u\} - \{(-X_{19})u\} + \{(X_{21})u\} + \{(-\gamma_{g1}dy_{g1})u + \\ & (\gamma_{g1}dx_{g1})v + (-\beta_{g1}dx_{g1})w\} + \{(Y_{g3})v + (Z_{g3})w\} + \{(Y_{g20})v + (Z_{g20})w\} + \{(Y_{g16})v + \\ & (Z_{g16})w\} + \{(-\gamma_{g12}dy_{g12})u + (\gamma_{g12}dx_{g12})v + (-\beta_{g12}dx_{g12})w\} + \{(-\gamma_{g21}dy_{g21})u + (\gamma_{g21}dx_{g21})v \\ & \left. + (-\beta_{g21}dx_{g21})w\} \right] \cdot \{(0.021773)u + (0.087327)v\} / \{(0.021773)u + \\ & (0.087327)v\} \cdot \{(0.021773)u + (0.087327)v\} \\ & \{(0.021773)u + (0.087327)v\} \end{aligned}$$

$$\begin{aligned}
& \left[\{ (X_{51} \cos(\lambda))u + (-X_{51} \sin(\lambda))v \} + \{ (X_{54} \cos(\lambda))u + (-X_{54} \sin(\lambda))v \} + \{ (X_{55} \cos(\lambda))u + \right. \\
& \quad \left. (-X_{55} \sin(\lambda))v \} + \{ (-\gamma_{g26} dy_{g26} \cos(\lambda) - (\gamma_{g26} dx_{g26}) \sin(\lambda))u + \right. \\
& \quad \left. (-\gamma_{g26} dy_{g26} \sin(\lambda) + (-\gamma_{g26} dx_{g26}) \cos(\lambda))v + (-\beta_{g26} dx_{g26})w \} + \{ (-\gamma_{g27} dy_{g27} \cos(\lambda) - \right. \\
& \quad \left. (\gamma_{g27} dx_{g27}) \sin(\lambda))u + (-\gamma_{g27} dy_{g27} \sin(\lambda) + (-\gamma_{g27} dx_{g27}) \cos(\lambda))v + \right. \\
& \quad \left. (-\beta_{g27} dx_{g27})w \} \right] \cdot \{ (0.021773)u + (0.087327)v \} / (\{ (0.021773)u + \\
& \quad (0.087327)v \} \cdot \{ (0.021773)u + (0.087327)v \}) \\
& \quad \{ (0.021773)u + (0.087327)v \} \\
& \quad + \\
& \quad \{ (-(-X_{jcu1}) - (-X_{jcu2}))u \} \cdot \{ (0.021773)u + (0.087327)v \} / (\{ (0.021773)u + \\
& \quad (0.087327)v \} \cdot \{ (0.021773)u + (0.087327)v \}) \\
& \quad \{ (0.021773)u + (0.087327)v \} \\
& \quad \leq \\
& \quad (0.021773)u + (0.087327)v
\end{aligned}$$

Contraintes associées

Y_{g3}	\leq	0.005 mm	$((0.0001/4)^{1/2})$
Z_{g3}	$=$	0	
Y_{g16}	\leq	0.005 mm	$((0.0001/4)^{1/2})$
Z_{g16}	$=$	0	
Y_{g20}	\leq	0.005 mm	$((0.0001/4)^{1/2})$
Z_{g20}	$=$	0	
β_{g12}	$=$	0	
γ_{g12}	\leq	3.7037×10^{-4}	(0.01/27.00)
β_{g21}	$=$	0	
γ_{g21}	\leq	1.7391×10^{-3}	(0.02/11.50)
β_{g26}	$=$	0	
γ_{g26}	\leq	5.6338×10^{-4}	(0.02/35.50)
β_{g1}	$=$	0	
γ_{g1}	\leq	1.25000×10^{-3}	(0.01/8.0)
β_{g27}	$=$	0	
γ_{g27}	\leq	1.6667×10^{-3}	(0.02/12.00)

Reformulons le système d'équations et d'inéquations en introduisant chacune des valeurs numériques des bras de levier et en éliminant les termes nuls.

Inéquation vectorielle principale

$$\begin{aligned}
 & \left[\{(0.14)u\} + \{(-\gamma_{g1}46.58756432)u + (\gamma_{g1}149.29727054)v\} - \{(-\gamma_{g12}46.58756432)u + \right. \\
 & \left. (\gamma_{g12}(-48.20272946))v\} + \{(Y_{g3})v\} + \{(Y_{g20})v\} + \{(Y_{g16})v\} - \{(-\gamma_{g21}46.58756432)u + \right. \\
 & \left. (\gamma_{g21}(-25.95272946))v\} \right] \cdot \{(2.6880019734)u + (10.7810199942)v\} \\
 & \quad \{(0.021773)u + (0.087327)v\} \\
 & - \\
 & \left[\{(0.01\cos(-14))u + (-0.01\sin(-14))v\} + \{(0.01\cos(-14))u + (-0.01\sin(-14))v\} + \right. \\
 & \left. \{(0.01\cos(-14))u + (-0.01\sin(-14))v\} - \{(-\gamma_{g26}17.95\cos(-14) - (\gamma_{g26}104.25)\sin(-14))u + \right. \\
 & \left. (-\gamma_{g26}17.95\sin(-14) + (-\gamma_{g26}104.25)\cos(-14))v\} + \{(-\gamma_{g27}17.95\cos(-14) - \right. \\
 & \left. (\gamma_{g27}(0))\sin(-14))u + (-\gamma_{g27}17.95\sin(-14) + (-\gamma_{g27}(0))\cos(-14))v\} \right] \\
 & \quad \cdot \{(2.6880019734)u + (10.7810199942)v\} \\
 & \quad \{(0.021773)u + (0.087327)v\} \\
 & + \\
 & \{(-(-X_{jeu1}) - (-X_{jeu2}))u\} \cdot \{(2.6880019734)u + (10.7810199942)v\} \\
 & \quad \{(0.021773)u + (0.087327)v\} \\
 & \leq \\
 & (0.021773)u + (0.087327)v
 \end{aligned}$$

Contraintes associées

Y_{g3}	\leq	0.005 mm	$((0.0001/4)^{1/2})$
Y_{g16}	\leq	0.005 mm	$((0.0001/4)^{1/2})$
Y_{g20}	\leq	0.005 mm	$((0.0001/4)^{1/2})$
γ_{g12}	\leq	3.7037×10^{-4}	(0.01/27.00)
γ_{g21}	\leq	1.7391×10^{-3}	(0.02/11.50)
γ_{g26}	\leq	5.6338×10^{-4}	(0.02/35.50)
γ_{g1}	\leq	1.25000×10^{-3}	(0.01/8.0)
γ_{g27}	\leq	1.6667×10^{-3}	(0.02/12.00)

Supposons que les tolérances géométriques sont orientées à leurs limites extrêmes c'est-à-dire, remplaçons les inégalités des contraintes associées par des égalités. L'inéquation principale se reformule de la façon suivante:

Inéquation vectorielle principale

$$\begin{aligned}
 & \left[\{(0.14)u\} + \right. \\
 & \quad \{((-1.25 \times 10^{-3})46.58756432)u + ((1.25 \times 10^{-3})149.29727054)v\} - \\
 & \quad \{((-3.7037 \times 10^{-4})46.58756432)u + ((3.7037 \times 10^{-4})(-48.20272946))v\} + \\
 & \quad \{(0.005)v\} + \{(0.005)v\} + \{(0.005)v\} - \\
 & \quad \{((-1.7391 \times 10^{-3})46.58756432)u + ((1.7391 \times 10^{-3})(-25.95272946))v\} \left. \right] . \\
 & \quad \{(2.6880019734)u + (10.7810199942)v\} \\
 & \quad \{(0.021773)u + (0.087327)v\} \\
 & - \\
 & \left[-\{(0.01 \cos(-14))u + (-0.01 \sin(-14))v\} - \right. \\
 & \quad \{(0.01 \cos(-14))u + (-0.01 \sin(-14))v\} - \\
 & \quad \{(0.01 \cos(-14))u + (-0.01 \sin(-14))v\} + \\
 & \quad \{((-5.6338 \times 10^{-4})17.95 \cos(-14) - ((5.6338 \times 10^{-4})104.25) \sin(-14))u + \\
 & \quad \{((-5.6338 \times 10^{-4})17.95 \sin(-14) + ((-5.6338 \times 10^{-4})104.25) \cos(-14))v\} - \\
 & \quad \{((-1.6667 \times 10^{-3})17.95 \cos(-14))u + ((-1.6667 \times 10^{-3})17.95 \sin(-14))v\} \left. \right] \\
 & \quad .\{(2.6880019734)u + (10.7810199942)v\} \\
 & \quad \{(0.021773)u + (0.087327)v\} \\
 & + \\
 & \quad \{((-X_{jeu1}) - (-X_{jeu2}))u\} . \{(2.6880019734)u + (10.7810199942)v\} \\
 & \quad \{(0.021773)u + (0.087327)v\} \\
 & \leq \\
 & \quad (0.021773)u + (0.087327)v
 \end{aligned}$$

L'inéquation devient après simplification:

$$\begin{aligned}
 & \{(0.1800406)u + (0.2646088)v\} \\
 & .\{(2.6880019734)u + (10.7810199942)v\} \\
 & \quad \{(0.021773)u + (0.087327)v\} \\
 & - \\
 & \{(0.004431609)u + (-0.0690366)v\} \\
 & .\{(2.6880019734)u + (10.7810199942)v\} \\
 & \quad \{(0.021773)u + (0.087327)v\} \\
 & + \\
 & \{((-X_{jeu1}) - (-X_{jeu2}))u\} . \{(2.6880019734)u + (10.7810199942)v\}
 \end{aligned}$$

$$\begin{aligned} & \{(0.021773)u + (0.087327)v\} \\ & \leq \\ & (0.021773)u + (0.087327)v \end{aligned}$$

Finalement, nous formulons le problème de la manière suivante:

$$\begin{aligned} & \{(0.07265002)u + (0.29138420)v\} \\ & - \\ & \{(-0.01594595)u + (-0.06395592)v\} \\ & + \\ & \{(-(-X_{\text{jeu1}}) - (-X_{\text{jeu2}}))u\} \cdot \{(2.6880019734)u + (10.7810199942)v\} \\ & \{(0.021773)u + (0.087327)v\} \\ & \leq \\ & (0.021773)u + (0.087327)v \end{aligned}$$

Cette inéquation vectorielle peut se réécrire sous forme de deux inéquations ordinaires qui sont combinaisons linéaires l'une de l'autre:

$$\begin{aligned} \text{pour } u : & 0.07265002 + 0.01594595 + 0.05852587(X_{\text{jeu1}} + X_{\text{jeu2}}) \leq 0.021773 \\ \text{pour } v : & 0.29138420 + 0.06395592 + 0.23473515(X_{\text{jeu1}} + X_{\text{jeu2}}) \leq 0.087327 \end{aligned}$$

et,

$$X_{\text{jeu1}} + X_{\text{jeu2}} \leq -1.14 \text{ mm}$$

Nous avons résolu le problème d'optimisation suggéré dans cette sous-section. Nous allons maintenant analyser le résultat obtenu.

6.3.1 Analyse des résultats

Cette dernière inégalité exprime la relation entre les jeux fonctionnels du variateur mécanique de vitesse et la limite supérieure de la contrainte fonctionnelle (B_{max}). Ainsi, lorsque tous les éléments dimensionnels des composantes du mécanisme sont à leur *état minimum de matière admissible*, soit la pire situation pour le

fonctionnement d'un mécanisme, les jeux étant maximum ainsi que les défauts géométriques.

Nous constatons que l'accumulation des défauts géométriques commandent la suppression d'un jeu supérieur à celui pré-défini par la condition fonctionnelle et ce, par l'ajout de matière. Quatre solutions sont proposées:

1. Nous pouvons réduire le nombre de composantes dans le cycle fonctionnel de manière à diminuer le nombre de défauts géométriques agissant sur le "backlash". Cette solution demande la reconception entière du mécanisme.
2. Nous pouvons donner un jeu suffisant sur les arbres de transmission de manière à assurer l'assemblage. Par la suite, nous devons définir un jeu de calles qui seront introduites dans le mécanisme au moment de son assemblage. Par exemple, grâce au décalage de la couronne, ces calles réduiront le "backlash" de l'engrenage dans sa zone acceptable (0.1 ± 0.09 mm). Ce jeu pourrait se composer de 8 calles de dimensions variées s'échellonnant de la façon suivante: 0.15 mm, 0.30 mm, 0.45 mm, 0.60 mm, 0.75 mm, 0.90 mm, 1.05 mm et 1.14 mm. Remarquons que l'introduction de calles dans le cycle fonctionnel altère sa fermeture géométrique car une nouvelle composante est ajoutée. Cette altération affecte particulièrement les TRANSPORTS des tolérances géométriques par de légères modifications des bras de levier. Elles ne produiront habituellement que de faibles variations des erreurs introduites par les tolérances géométriques. Nous devons donc analyser chacun des problèmes de manière à déterminer la validité des calculs d'accumulation.
3. Nous pouvons surdimensionner une pièce d'appui telle que la pièce 26 (voir annexe I) de manière à créer un jeu suffisant à l'assemblage. Ensuite, elle sera réusinée pour chacun des variateurs assemblés de façon à réduire le jeu et

obtenir un "backlash" acceptable. Encore une fois, cette solution altère la fermeture géométrique du cycle et une procédure similaire à celle proposée à la solution 2 doit être appliquée.

4. Nous savons que les résultats de l'analyse présente la situation la plus désavantageuse pour le fonctionnement du mécanisme. En effet, nous avons calculé la condition limite générée lorsque tous les défauts géométriques tendent à accroître la contrainte fonctionnelle. Il est peut probable que toutes les pièces produites soient à leur état minimum de matière. Nous pouvons donc apparier des pièces différentes pour chercher une combinaison intermédiaire satisfaisant la condition fonctionnelle.

Nous avons résolu le problème du variateur mécanique de vitesse sur son cycle fonctionnel 1. Les résultats obtenus satisfont l'objectif que nous nous étions fixé. En effet, nous avons démontré que la méthode vectorielle d'analyse permet de résoudre des problèmes complexes d'ingénierie.

Le chapitre suivant traite du logiciel que nous avons développé pour manipuler informatiquement les torseurs.

7.0 LOGICIEL DE CALCUL PAR TORSEUR

Ce chapitre présente le logiciel de calcul symbolique réalisé au cours des travaux de recherche qui ont mené à la rédaction de ce mémoire. Ce logiciel a permis de réaliser efficacement la validation de la méthode vectorielle d'analyse des erreurs.

Généralités

Le logiciel de calcul symbolique a été développé en langage C. Il comprend plusieurs bibliothèques de fonctions permettant de résoudre la plupart des problèmes de nature vectorielle tels que ceux engendrés par les torseurs et ce, dans un espace orthonormé direct.

Il est clair que l'objectif des présents travaux de recherche n'est nullement d'arriver à produire un logiciel automatisant l'analyse de l'accumulation des erreurs mécaniques. Ainsi, les fonctions du calculateur symbolique ont été créées dans le seul but de faciliter l'avancement des travaux. Toutefois, la création de ces fonctions C permettant de résoudre les problèmes d'analyse d'accumulation des erreurs mécaniques, illustre la faisabilité d'une démarche d'automatisation. Plusieurs des étapes des travaux de recherche ont nécessité l'emploi du logiciel de calcul symbolique. Ainsi, la validation de la méthodologie a été réalisée grâce à la bibliothèque de fonctions du logiciel. Les fonctions ont été organisées dans un programme de façon à calculer itérativement les résultats numériques des erreurs sur l'angle de sortie ($\Delta\phi$) en fonction de l'angle d'entrée (θ), la longueur nominale des membrures (R_1, R_2, R_3, R_4) et finalement, les tolérances dimensionnelles de ces membrures ($\Delta R_1, \Delta R_2, \Delta R_3, \Delta R_4$).

Vous trouverez à l'annexe II la définition des principales fonctions des bibliothèques du logiciel de calcul symbolique.

CONCLUSION

L'objectif des présents travaux de recherche est de mettre au point une nouvelle méthodologie d'analyse des accumulations des défauts géométriques et des jeux dans les systèmes mécaniques tridimensionnels. Cette nouvelle méthodologie est basée sur des outils mathématiques vectoriels compatibles permettant de fondre les divers types de tolérances et de jeux. Pour se faire, nous employons les torseurs de petits déplacements et les exponentielles vectorielles. Nous cherchons également à développer une méthodologie facilitant la démarche d'informatisation de l'analyse de la propagation et du contrôle des défauts géométriques et des jeux.

Certaines hypothèses simplificatrices ont été définies préalablement au développement de la méthodologie vectorielle d'analyse des erreurs mécaniques:

1. nous considérons que les mécanismes étudiés sont constitués d'un ensemble de pièces et de liaisons mécaniques formant une ou plusieurs boucles fermées;
2. nous considérons les déformations volumiques dues à des efforts comme absentes ou négligeables;
3. nous considérons les déformations surfaciques dues à des efforts comme absentes ou négligeables;
4. nous considérons les composantes mécaniques comme des corps indéformables (liées aux hypothèses 2 et 3);
5. nous considérons les composantes mécaniques comme étant en situation réelle et possédant des défauts géométriques (tolérances dimensionnelles et tolérances géométriques);
6. nous proposons que les liaisons possèdent des jeux;
7. nous définissons une condition fonctionnelle comme un espace contrôlé entre deux composantes;
8. nous considérons que la forme d'une surface est parfaite.

Nous comparons la nouvelle méthodologie d'analyse avec la méthode d'analyse différentielle des erreurs régulièrement employées dans le domaine de la robotique. Nous employons le modèle bidimensionnel du générateur d'équations à quatre membrures pour comparer ces deux méthodes. Les résultats confirment la validité de la nouvelle méthodologie d'analyse proposée. En effet, les résultats générés par la méthode d'analyse vectorielle sont similaires à ceux générés par l'approche différentielle. La variation maximale observée entre les résultats obtenus pour les deux méthodes est de l'ordre de 0.1% pour des tolérances dimensionnelles et angulaires petites par rapport aux dimensions nominales des membrures. Nos résultats montrent également que les deux méthodes d'analyse conduisent à des erreurs importantes par rapport à la solution exacte lorsque que les tolérances sont grandes par rapport aux dimensions nominales. Ces déviations sont particulièrement importantes dans les régions voisines des points d'inflexion des courbes d'erreurs.

Nous démontrons l'utilité et la validité de notre nouvelle méthodologie d'analyse par le traitement d'un cas d'ingénierie réel: le variateur mécanique de vitesse. Grâce à ce modèle tridimensionnel, nous montrons qu'il est possible de contrôler la propagation des défauts géométriques et des jeux à l'intérieur d'un cycle tout en respectant une condition fonctionnelle préalablement définie. Nous avons validé les inéquations développées grâce à notre méthodologie en calculant les jeux fonctionnelles du mécanisme lorsqu'un "backlash" maximum est exigé au niveau des dents des engrenages coniques. Nos résultats confirment l'utilité et la validité de la nouvelle méthodologie d'analyse. En effet, nous obtenons la relation suivante:

$$X_{\text{jeu1}} + X_{\text{jeu2}} \leq -1.14 \text{ mm}$$

qui illustre l'existence de certains problèmes de conception du mécanisme. Nous remarquons une surabondance de tolérances (une trentaine de tolérances) à l'intérieur du cycle fonctionnel analysé. Quatres solutions sont alors proposées:

2. création d'un jeu de 8 calles (0.15 mm, 0.30 mm, 0.45 mm, 0.60 mm, 0.75 mm, 0.90 mm, 1.05 mm et 1.14 mm) qui sont introduit au moment de l'assemblage de chacun des variateurs mécaniques de vitesse de façon à obtenir un "backlash" acceptable;
3. surdimensionnement de la pièce 26 de manière à générer un jeu suffisant au cours de l'assemblage. Réusinage de cette pièce pour obtenir un "backlash" acceptable;
4. recherche de la meilleure combinaison de pièces satisfaisant la condition fonctionnelle.

Grâce aux travaux de recherche entrepris, nous avons développé une nouvelle méthodologie d'analyse apte à cumuler et à contrôler les défauts géométriques ainsi que les jeux à l'intérieur de modèles mécaniques tridimensionnels. Contrairement aux autres méthodes d'analyse des erreurs mécaniques, nous employons des outils mathématiques compatibles les uns aux autres facilitant l'intégration des divers types de tolérances et de jeux. Cette nouvelle procédure d'analyse possède les qualités suffisantes à son utilisation pour l'informatisation du contrôle des jeux et des tolérances mécaniques. En effet, la nouvelle méthodologie d'analyse contourne l'ensemble des problèmes associés à l'analyse différentielle des erreurs. Ainsi, nous n'avons pas d'opérations de différentiation à effectuer numériquement évitant ainsi un grand nombre de difficultés associées aux calculs d'approximation numérique. De plus, nous pouvons employer des équations simples ne nécessitant aucun traitement préliminaire particulier.

La méthodologie d'analyse proposée possède certaines limitations. En effet, nous devons limiter l'utilisation de cette méthode à l'analyse de modèles mécaniques dont les tolérances et les jeux sont faibles vis-à-vis des dimensions nominales correspondantes. Remarquons toutefois, qu'une analyse différentielle possède une limitation similaire. Une autre limitation non négligeable de la méthodologie proposée est sans nul doute, l'importante quantité d'équations et d'inéquations générées au cours d'une analyse. Ces

systèmes d'équations et d'inéquations sont difficiles à manipuler manuellement. La méthodologie d'analyse vectorielle doit donc être employée par l'intermédiaire d'une application informatique inexistante pour le moment. La résolution des systèmes d'équations et d'inéquations au cours d'opérations d'optimisation peut également représenter un obstacle d'importance. En effet, ces systèmes sont composés d'équations et d'inéquations de types non-linéaires qui représentent une difficulté lors de résolutions numériques.

Il est clair que nos travaux de recherches ne constituent qu'une étape préliminaire à des développements plus poussés de l'analyse de la propagation et du contrôle des tolérances et des jeux dans les systèmes mécaniques tridimensionnels. Ainsi, des travaux de recherche et de développements peuvent être entrepris pour compléter les concepts mis en place dans ce mémoire:

- Création d'une procédure d'optimisation des produits mécaniques.
- Introduction des concepts de logique floue pour définir un tolérancement adéquat en fonction de multiples facteurs techniques.
- Résolution d'un problème complet, car nous n'avons présenté que l'analyse partielle d'un cas pratique.
- Création d'un logiciel intégrant les concepts de ce mémoire.
- Développement de la méthodologie pour l'analyse multi-cycles.
- Introduction de l'effet des déformations dans la méthodologie.
- Introduction d'une méthode d'estimation des efforts dans le mécanisme.

BIBLIOGRAPHIE

- [AGA86] AGATI P., BREMONT Y. et DELVILLE G., "Mécanique du solide - Applications Industrielles", Dunod, Paris, 1986, 270p.
- [AUS65] AUSTIN, T.C., DENAVIT J. and HARTENBERG R.S., "Analysis of error in the double hooke joint", ASME Journal of engineering for industry, may 1965, p. 251-257.
- [BER70] BERGE, C., "Graphes et Hypergraphes", Dunod, Paris, 1970, 502p.
- [BJO89] BJØRKE, Ø., "Computer-Aided Tolerancing", second edition, ASME press, New York, 1989, 216 p.
- [CLE94] CLEMENT, A., RIVIÈRE, A. et TEMMERMAN M., "Cotation tridimensionnelle des systèmes mécaniques - Théorie et Pratique", PYC, Paris, 1994, 317 p.
- [CLE93] CLEGHORN, W.L., R.G. FENTON and FU J.F., "Optimum tolerancing of planar mechanisms based on an error sensitivity analysis", ASME journal of mechanical design, vol. 115, june 1993, p.306-313.
- [DAI94] DAIN-LIN, P. and CHEN J.F., "Analysis of errors in precision for closed loop mechanisms", ASME journal of mechanical design, vol. 116, march 1994, p.197-203.
- [DEN75] DENAVIT, J. and HARTENBERG R.S., "A kinematic notation for lower-pair mechanisms based on matrices", Journal of Applied mechanics, vol.22, june 1975, p.215-221.

- [DHA73] DHANDE, S.G. and CHAKRABORTY J., "Analysis and Synthesis of mechanical error in linkages - A stochastic approach", ASME journal of engineering for industry, august 1973, p. 672-676.
- [DHA78] DHANDE, S.G. and CHAKRABORTY J., "Mechanical error analysis of spatial linkages", ASME journal of mechanical design, october 1978, p.732-738.
- [DUP91] DUPINET, É., "Contribution à l'étude d'un système informatique d'aide à la conception de produits mécaniques par la prise en compte des relations fonctionnelles", École Centrale Paris, 1991, thèse de doctorat.
- [FEN89] FENTON, R.G., CLEGHORN W.L. and FU J.F., "Allocation of dimensional tolerances for multiple loop planar mechanisms", ASME journal of mechanisms, transmissions, and automation in design, vol.111, december 1989, p.465-470.
- [FU88] FU, J.F., FENTON R.G. and CLEGHORN W.L., "Synthesis of the dimensional tolerances of a four-bar mechanism", Transactions of the CSME, vol.12 no 1, 1988, p.9-14.
- [GAR62] GARRETT, R.E. and HALL A.S. jr, "Optimal Synthesis of randomly generated linkages", ASME: journal of engineering for industry, august 1968, p.475-480.
- [GAR69] GARRETT, R.E. and HALL A.S. jr, "Effect of tolerance and clearance in linkage design", ASME journal of engineering for industry, february 1969, p.198-202.

- [GAU93] GAUNET, D., "Vectorial Tolerancing Model", CIRP, Proceedings of 3rd CIRP Seminars on Computer Aided Tolerancing, April 27-28 1993, p.25-49.
- [GIL92] GILBERT, O.L., "Representation of geometric variations using matrix transforms for statistical tolerance analysis in assemblies", Massachusetts Institute of Technology, may 1992, master of science thesis.
- [HER75] HERVÉ J.M., "La géométrie du groupe des déplacements appliquée à l'analyse cinématique des mécanismes", Thèse de doctorat d'état es sciences physiques, Université Paris VI, mars 1975.
- [LEE94] LEE, S. and YI C., "Tolerance Analysis for Assembly Planning", Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology, 1994, p.306-311.
- [LEE95] LEE, S. and YI C., "Statistical Measure of Assemblability under the propagation of tolerances and clearances", 1995 IEEE International Conference..., 1995, p.94-99.
- [SPE93] SPENLÉ D. and GOURHANT R., "Guide du calcul en mécanique", Hachette Technique, Paris, 1993, 256 p.
- [VEI86] VEITSCHEGGER, W.K. and WU C.H., "Robot accuracy analysis based on kinematics", IEEE journal of robotics and automation, vol. RA-2 no 3, september 1986, p.171-179.

- [WHI93] WHITNEY, D.E. and GILBERT O.L., "Representation of geometric variations using matrix transforms for statistical tolerance analysis in assemblies", 1993 IEEE transactions on..., 1993, p.314-321.

ANNEXES

Liste des annexes

	page
ANNEXE I	
Dessins du variateur mécanique de vitesse	106
ANNEXE II	
Programme du calculateur symbolique	111
ANNEXE III	
Torseurs employés lors de l'analyse du variateur mécanique de vitesse	134

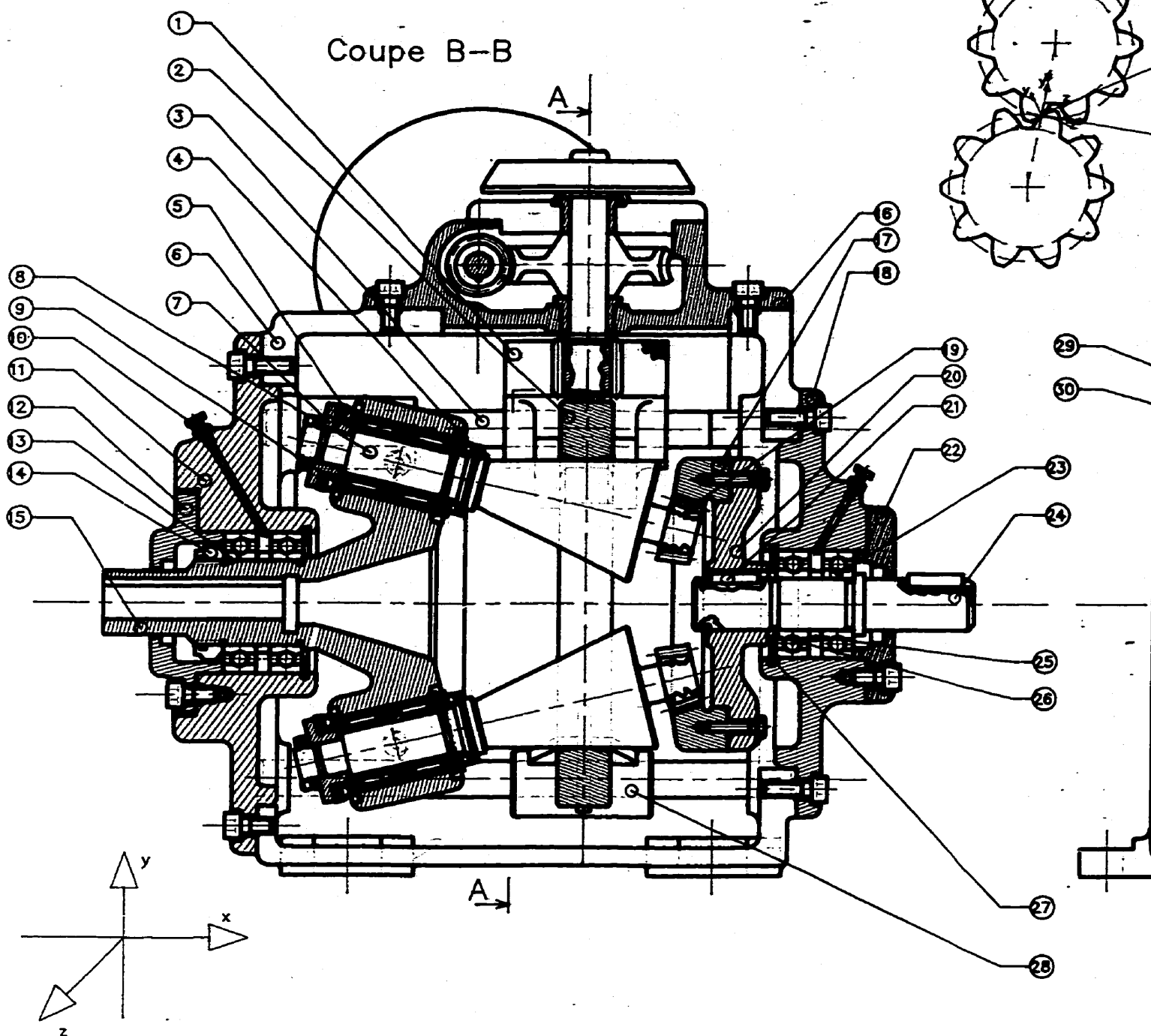
ANNEXE I

DESSINS DU VARIATEUR MÉCANIQUE DE VITESSE

Liste des dessins

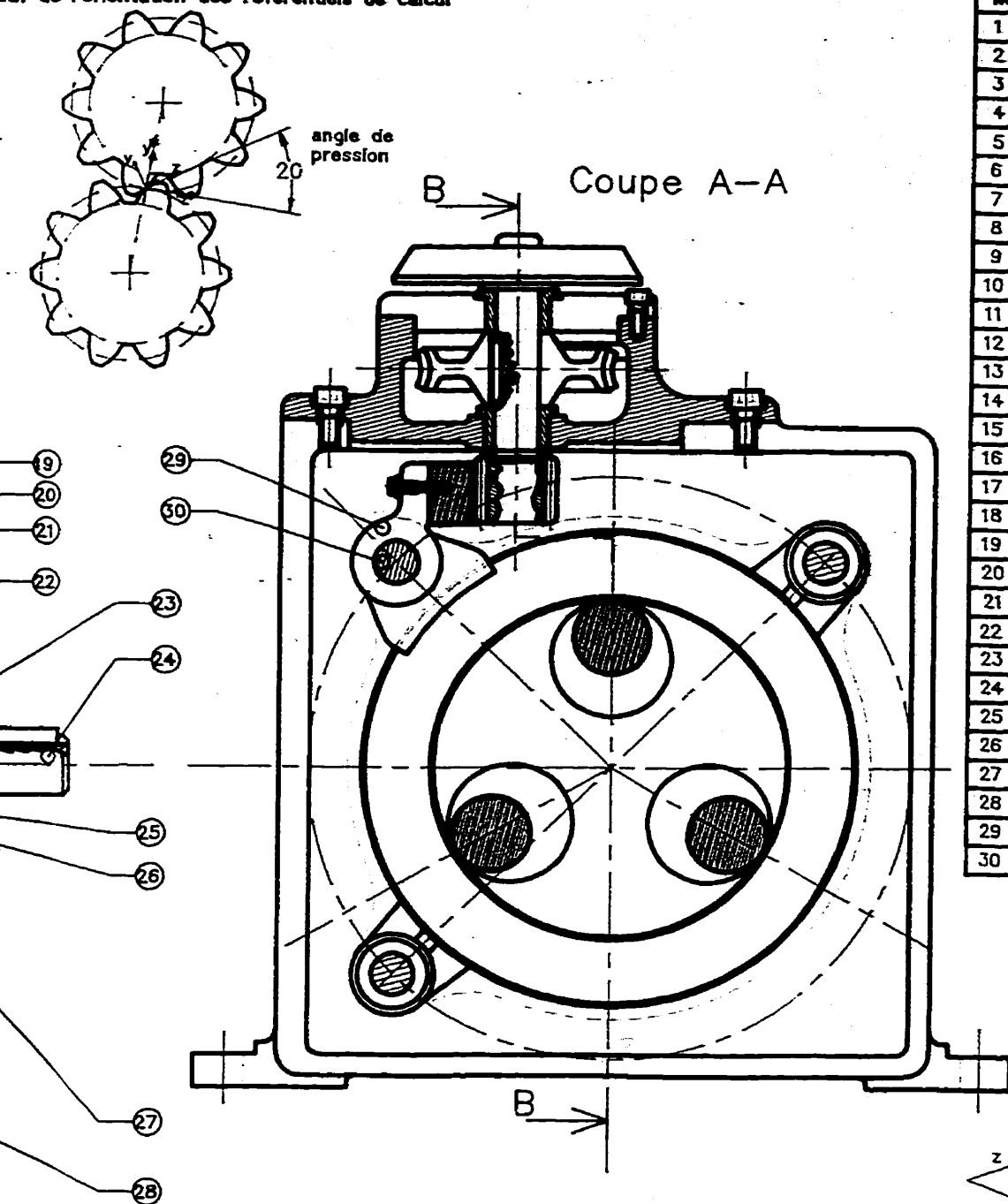
	page
I.1 Dessin d'assemblage	107
I.2 Dessin du cycle fonctionnel 1	108
I.3 Dessin des composantes 1	109
I.4 Dessin des composantes 2	110

I.1 Dessin d'assemblage



NOTA:
Seuls les details necessaires a la
comprehension du dessin ont ete
representes.

CARACTERISTIQUES GENERALES
Puissance transmissible maximum :
Couple resistant maximum autorise
Possibilite d'inversion du sens de r
Rapport de reduction : $1/3 \leq r <$



NO.	DESCRIPTION	QT.	MAT.
1	Chariot de controle	1	acier
2	Anneau de controle	1	acier
3	Tige d'alignement	2	acier
4	Roulement a billes (av.)	3	Multi
5	Roulement a billes (ar.)	3	Multi
6	Boltier	1	acier
7	Gaine de soutien	3	acier
8	Rouleau de transmission	3	acier
9	Roulement a rouleaux	6	Multi
10	Valve de graissage	2	Multi
11	Couvercle 1	1	acier
12	Couvercle 2	1	acier
13	Roulement a billes	2	Multi
14	Ecrou de blocage	1	acier
15	Arbre d'entree	1	acier
16	Couvercle 3	1	acier
17	Couronne d'engrenage	1	acier
18	Couvercle 4	1	acier
19	Engrenage conique	3	acier
20	Disque de support	1	acier
21	Cle	1	acier
22	Couvercle 5	1	acier
23	Roulement a billes	2	Multi
24	Arbre de sortie	1	acier
25	Rondelle	2	acier
26	Rondelle	1	acier
27	Cardclip	1	acier
28	Glissiere intermediaire	1	acier
29	Glissiere de controle	1	acier
30	Tige de controle	1	acier

ENERALES

le maximum : 4 CV
 imum autorise : 2.5 mdaN
 du sens de rotation
 $1/3 \leq r < \infty$

Analyse des erreurs -
 M.Sc.A. 1985-1987

Variateur mecanique de vitesse 15/11/88
 M.Sc.A. 1985-1987

A3-01

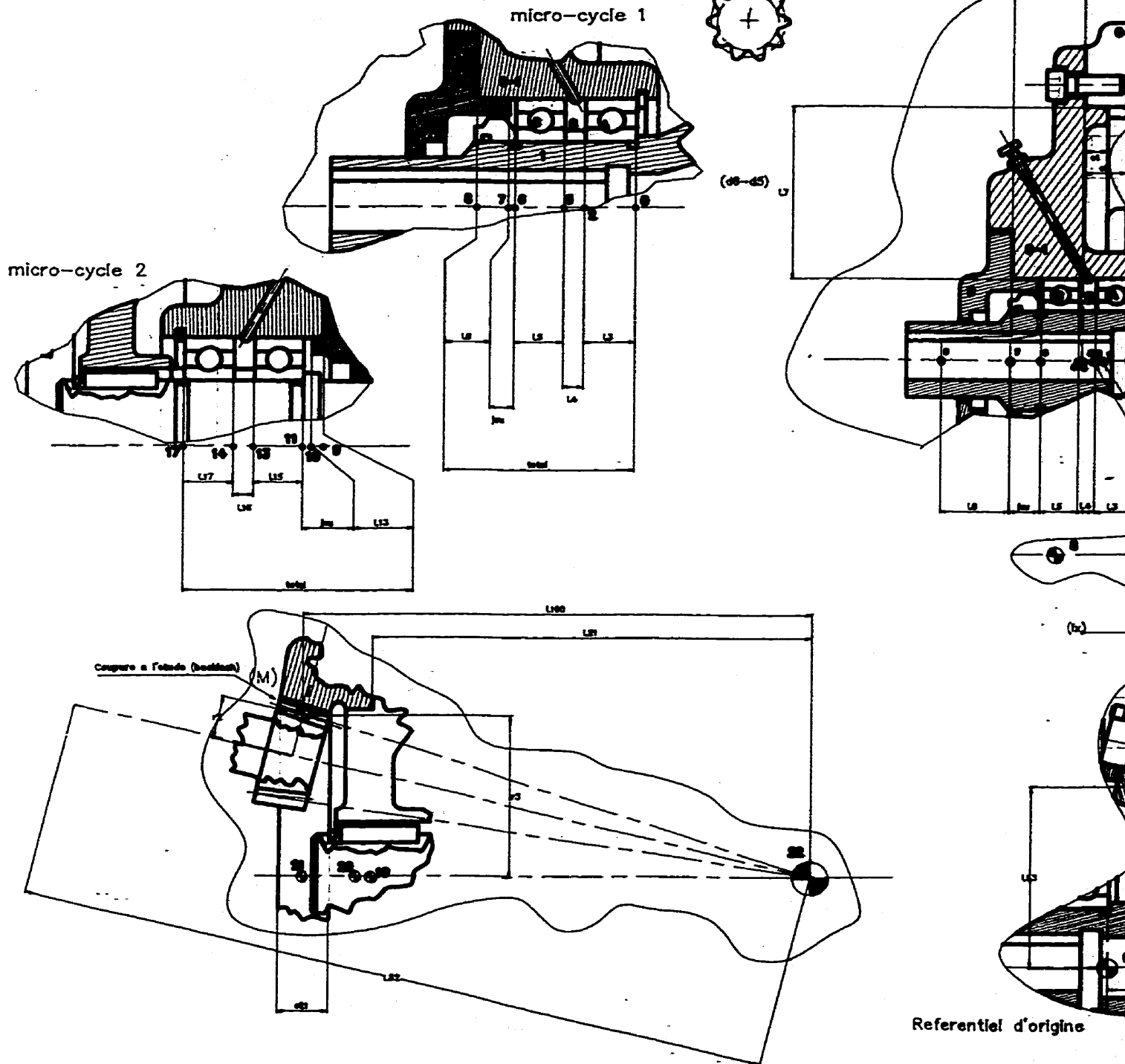
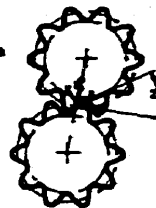
I.2 Dessin du cycle fonctionnel 1

Analyse des erreurs geometriques et des erreurs dimensionnelles - sur le variateur mecanique de vitesse de maniere a connaitre l'effet des jeux et des erreurs sur le backlash de l'engrenage conique.

Demi-boucle 1

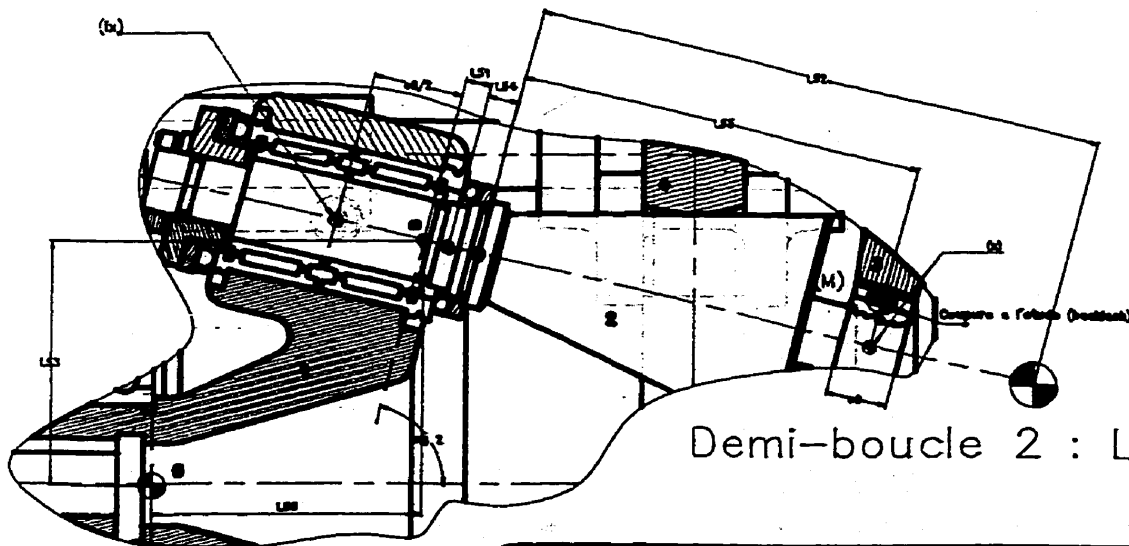
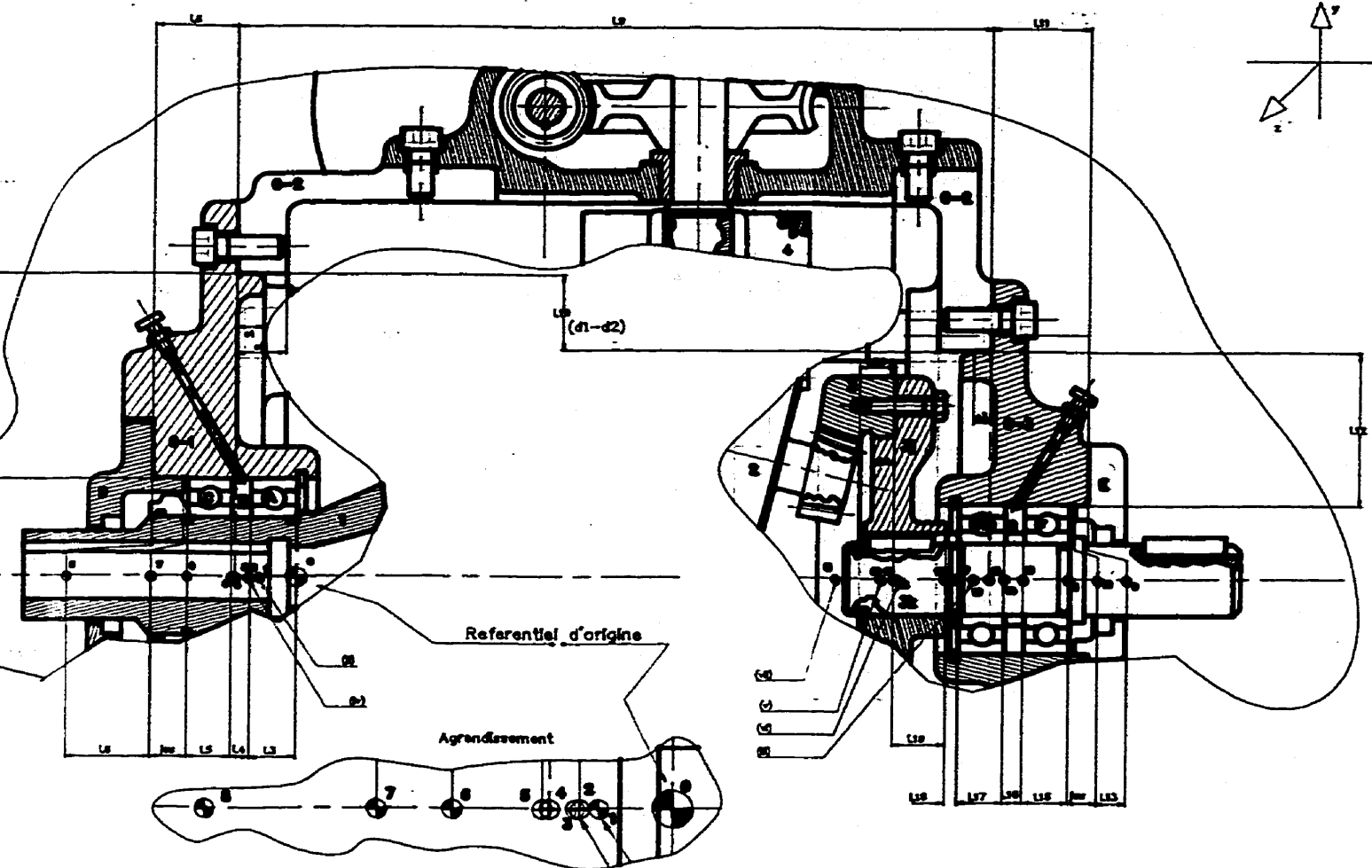
Les vues presentees sur ce dessin sont des coupes simplifiees du plan original.

Schema illustrant les referentiels de calcul au contact des dents de l'engrenage conique



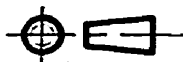
Demi-boucle 1 : L1 - L23

NO.	DESCRIPTION	ET.	MAT.



Demi-boucle 2 : L50 - L54

Referentiel d'origine



Analyse des erreurs -
maîtrise (M.Sc.A.)
1995-1997

Verificateur mécanique de vitesse 20/01/97

Revisé par
Benoit Dupont

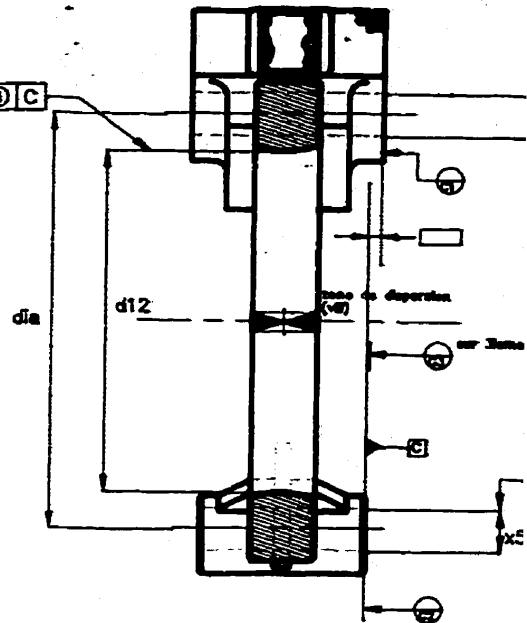
Revisé par
S.A.

AS - 02

I.3 Dessin des composantes 1

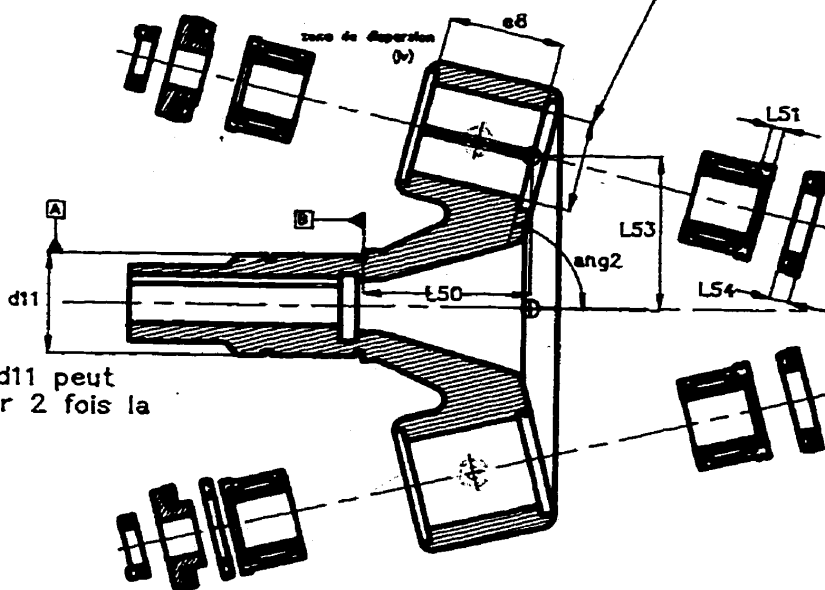
I.4 Dessin des composantes 2

(viii) $\oplus \otimes t, \oplus A \otimes B \oplus C$



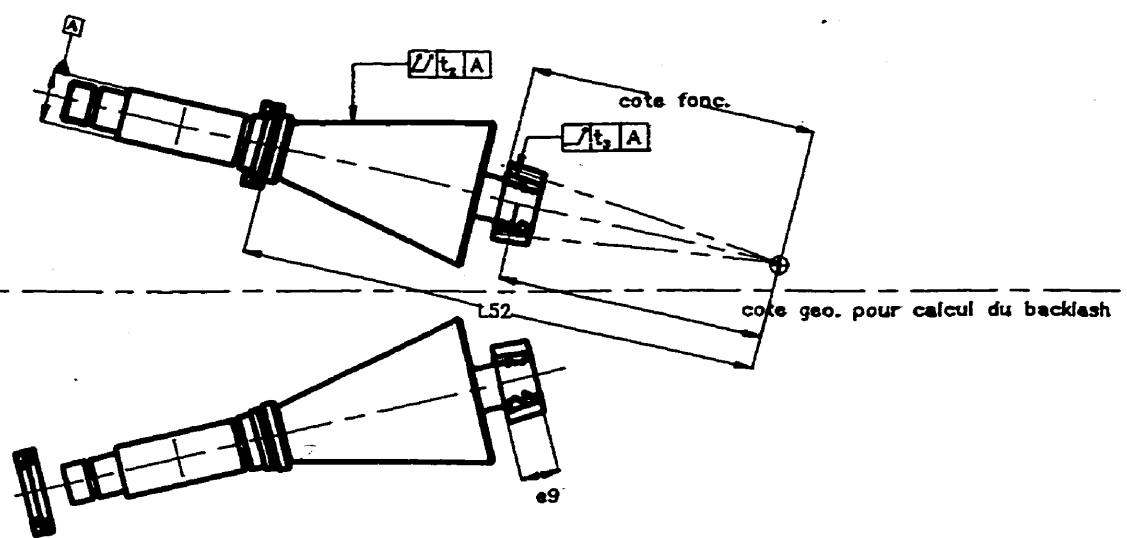
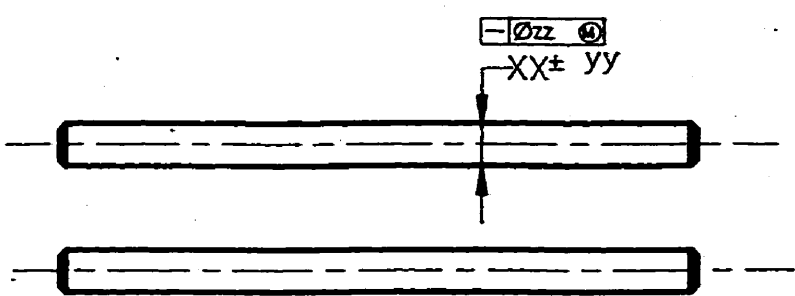
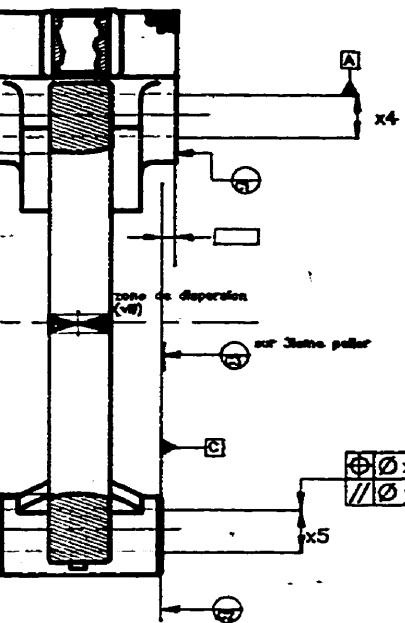
3 x trou

$\oplus \otimes t, \oplus A \otimes B$



Le diamètre d11 peut s'exprimer par 2 fois la longueur L1
($d11 = 2L1$)

NO.	DATE	DESCRIPTION	QT.	MAT.



Analyse des erreurs -		Vérification mécanique de vitesse		20/01/97
M.Sc.A. 1985-1986		Bureau d'études		A3 - 04

ANNEXE II

PROGRAMME DU CALCULATEUR SYMBOLIQUE

Liste des programmes

	page
II.1 Programme C utilisé pour la validation de la méthodologie	112
II.2 Descriptions des fonctions principales du calculateur symbolique	119

II.1 Programme C utilisé pour la validation de la méthodologie

VALIDATION.C

```

/* ----- fichiers inclus ----- */

#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "structurebd.h"
#include "stuctorsor.h"
#include "structure.h"

#include "externe.h"
#include "superieur.h"

#include "torsor.h"
#include "torchaine.h"
#include "creftrans.h"
#include "maitrise.h"

extern parametre *pointeur;
extern torliste *TT;

/* ----- symbole de précompilation (define) ----- */
/* ----- Variables globales ----- */

/* ***** */
/* D E B U T */
/* ***** */

int main(int argc, char **argv)
{
    size_t memdim    = 0;

    char *v1          = '\0';
    char *v2          = '\0';
    char *v3          = '\0';
    char *v4          = '\0';
    char *v5          = '\0';
    char *v6          = '\0';
    char *v7          = '\0';
    char *norm1       = '\0';
    char *norm2       = '\0';
    char *norm3       = '\0';
    char *norm4       = '\0';
    char *norm5       = '\0';
    char *norm6       = '\0';
    char *norm7       = '\0';
    char *normr1      = '\0';
    char *normr2      = '\0';
    char *valv1       = '\0';
    char *valv2       = '\0';
    char *valv3       = '\0';
    char *valv4       = '\0';
    char *valv5       = '\0';
    char *valv6       = '\0';
    char *valv7       = '\0';
    char *valvr1      = '\0';
    char *valvr2      = '\0';
    char *valvR1      = '\0';
    char *valvR2      = '\0';
    char *nl          = '\0';

```

```

char *n2      = '\0';
char *r1      = '\0';
char *r2      = '\0';
char *R1      = '\0';
char *R2      = '\0';
char *normR1   = '\0';
char *normR2   = '\0';
char *t1      = '\0';
char *t2      = '\0';
char *last1    = '\0';
char *last     = '\0';

char *f1, *f2;

double ang3, ang4;

double angle2  = 0.0;
double angle3  = 0.0;
double angle4  = 0.0;

double val1,
       val2,
       val3,
       val4,
       val5,
       val6,
       val7,
       valr1,
       valr2,
       valR1,
       valR2;

double rr1,
       rr2,
       rr3,
       rr4;

double AA,
       BB,
       CC,
       DD;

double phim,
       phip;

double AA2,
       BB2,
       CC2,
       DD2;

double phim2,
       phip2;

double ang;
double d3, d4, dd3, dd4, a3, a4;
double z1, z2, z3, z4;

const double multi = 1.0;
const double pi = 3.14159265359;

/* note: comment stocker les parametres autres que les torseurs....*/

memdim      = sizeof(char);

normR1      = (char*)malloc(memdim);
normR2      = (char*)malloc(memdim);
last        = (char*)malloc(memdim);
last1       = (char*)malloc(memdim);
v1          = (char*)malloc(memdim);
v2          = (char*)malloc(memdim);
v3          = (char*)malloc(memdim);
v4          = (char*)malloc(memdim);
v5          = (char*)malloc(memdim);
v6          = (char*)malloc(memdim);

```



```

v7      = (char*)malloc(memdim);
norm1   = (char*)malloc(memdim);
norm2   = (char*)malloc(memdim);
norm3   = (char*)malloc(memdim);
norm4   = (char*)malloc(memdim);
norm5   = (char*)malloc(memdim);
norm6   = (char*)malloc(memdim);
norm7   = (char*)malloc(memdim);
normr1  = (char*)malloc(memdim);
normr2  = (char*)malloc(memdim);
valv1   = (char*)malloc(memdim);
valv2   = (char*)malloc(memdim);
valv3   = (char*)malloc(memdim);
valv4   = (char*)malloc(memdim);
valv5   = (char*)malloc(memdim);
valv6   = (char*)malloc(memdim);
valv7   = (char*)malloc(memdim);
valvr1  = (char*)malloc(memdim);
valvr2  = (char*)malloc(memdim);
valvR1  = (char*)malloc(memdim);
valvR2  = (char*)malloc(memdim);

r1      = (char*)malloc(memdim);
r2      = (char*)malloc(memdim);
R1      = (char*)malloc(memdim);
R2      = (char*)malloc(memdim);

t1      = (char*)malloc(memdim);
t2      = (char*)malloc(memdim);

n1      = (char*)malloc(memdim);
n2      = (char*)malloc(memdim);

inittorseur("mattor.dat");

/* ***** */
/* definition des torseurs */
/* ***** */

puttorseur("E1", "{rx1}[u + {ry1}[v + {rz1}[w" , "{dr1}[u + {my1}[v +
{mz1}[w", 1);
puttorseur("E2", "{rx2}[u + {ry2}[v + {rz2}[w" , "{dr2}[u + {my2}[v +
{mz2}[w", 1);
puttorseur("E3", "{rx3}[u + {ry3}[v + {rz3}[w" , "{dr3}[u + {my3}[v +
{mz3}[w", 1);
puttorseur("E4", "{rx4}[u + {ry4}[v + {dtheta2}[w" , "{mx4}[u + {my4}[v +
{mz4}[w", 1);
puttorseur("E5", "{rx5}[u + {ry5}[v + {dtheta3}[w" , "{mx5}[u + {my5}[v +
{mz5}[w", 1);
puttorseur("E6", "{rx6}[u + {ry6}[v + {rz6}[w" , "{dr4}[u + {my6}[v +
{mz6}[w", 1);
puttorseur("E7", "{rx7}[u + {ry7}[v + {dtheta4}[w" , "{mx7}[u + {my7}[v +
{mz7}[w", 1);

z1 = atof(argv[1]);
z2 = atof(argv[2]);
z3 = atof(argv[3]);
z4 = atof(argv[4]);

/* z1 = 0.05; z2 = 0.05 ; z3 = 0.05 ; z4= 0.05; */

put_T_num ("E1", z1 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0);
put_T_num ("E2", z2 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0);
put_T_num ("E3", z3 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0);
put_T_num ("E4", 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0); /* 0.06981317 */
put_T_num ("E5", 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0); /* angle */
put_T_num ("E6", z4 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0);
put_T_num ("E7", 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0); /* angle */

```

```

/* ***** */
/* elimination des variables inutiles */
/* ***** */

put_nulle_var("rz1");
put_nulle_var("rx1");
put_nulle_var("ry1");
put_nulle_var("my1");
put_nulle_var("mz1");
put_nulle_var("rz2");
put_nulle_var("rx2");
put_nulle_var("ry2");
put_nulle_var("my2");
put_nulle_var("mz2");
put_nulle_var("rz6");
put_nulle_var("rx6");
put_nulle_var("ry6");
put_nulle_var("my6");
put_nulle_var("mz6");
put_nulle_var("rz3");
put_nulle_var("rx3");
put_nulle_var("ry3");
put_nulle_var("my3");
put_nulle_var("mz3");
put_nulle_var("mx4");
put_nulle_var("rx4");
put_nulle_var("ry4");
put_nulle_var("my4");
put_nulle_var("mz4");
put_nulle_var("mx5");
put_nulle_var("rx5");
put_nulle_var("ry5");
put_nulle_var("my5");
put_nulle_var("mz5");
put_nulle_var("mx7");
put_nulle_var("rx7");
put_nulle_var("ry7");
put_nulle_var("my7");
put_nulle_var("mz7");

/* ***** */
/* transports */
/* ***** */
TRANSPORT
("E1", "[trans(u,r3),rot(w,t3-t2),trans(u,r2),rot(w,t2),trans(u,-r1)]", "E1F");
TRANSPORT("E2", "[trans(u,r3),rot(w,t3-t2),trans(u,r2)]", "E2T");
TRANSPORT("E3", "[trans(u,r3)]", "E3T");
TRANSPORT("E4", "[trans(u,r3),rot(w,t3-t2),trans(u,r2),rot(w,t2)]", "E4T");
TRANSPORT("E5", "[trans(u,r3),rot(w,t3)]", "E5T");
TRANSPORT("E6", "[trans(u,r4)]", "E6T");
TRANSPORT("E7", "[trans(u,r4),rot(w,t4)]", "E7F");

/* ***** */
/* changements de referentiels */
/* ***** */

CR("E2T", "[trans(u,r1),rot(w,t2)]", "E2F");
CR("E3T", "[trans(u,r1),rot(w,t2),trans(u,-r2),rot(w,-t2+t3)]", "E3F");
CR("E4T", "[trans(u,r1)]", "E4F");
CR("E5T", "[trans(u,r1),rot(w,t2),trans(u,-r2),rot(w,-t2)]", "E5F");
CR("E6T", "[rot(w,t4)]", "E6F");

/* ***** */
/* les nominaux */
/* ***** */

vecteur_deplacement
("[trans(u,r3),rot(w,t3-t2),trans(u,r2),rot(w,t2),trans(u,-r1)]", &n1);
vecteur_deplacement("[trans(u,r4),rot(w,t4)]", &n2);

```

```

/* ***** */
/* les vecteurs d'erreurs */
/* ***** */

getsym_moment("E1F",&v1);
getsym_moment("E2F",&v2);
getsym_moment("E3F",&v3);
getsym_moment("E4F",&v4);
getsym_moment("E5F",&v5);
getsym_moment("E6F",&v6);
getsym_moment("E7F",&v7);

/* ***** */
/* les deux vecteurs d'erreurs (chemin 1 et chemin 2) */
/* ***** */

/* vecteur r1 */

addition_vecteur(v1,v2,&t1);
addition_vecteur(t1,v3,&t2);

addition_vecteur(t2,v5,&r1);

/* vecteur r2 */

addition_vecteur(v6,v7,&r2);

printf("dr1: %2.4lf dr2: %2.4lf dr3: %2.4lf dr4: %2.4lf \n\n",z1,z2,z3,z4);

printf("nominiaux : n1 : \n %s \n\n",n1);
printf("nominiaux : n2 : \n %s \n\n",n2);

printf("erreurs \n");
printf("erreur 1: \n %s \n\n",v1);
printf("erreur 2: \n %s \n\n",v2);
printf("erreur 3: \n %s \n\n",v3);
printf("erreur 4: \n %s \n\n",v4);
printf("erreur 5: \n %s \n\n",v5);
printf("erreur 6: \n %s \n\n",v6);
printf("erreur 7: \n %s \n\n",v7);

printf("erreur total du chemin 1: \n %s \n\n",r1);
printf("erreur total du chemin 2: \n %s \n\n",r2);

eqn2(r1,r2,&last,'u');
eqn2(r1,r2,&last1,'v');

eqn1(n1,&f1);
eqn2(n1,n2,&f2,'u');

/* ***** NUMERIQUE ***** */

/* ***** */
/* initialisation de la resolution numerique */
/* ***** */

printf("dr1: %3.2lf dr2: %3.2lf dr3: %3.2lf dr4: %3.2lf \n",z1,z2,z3,z4);

init_calcul("inctransfo.dat");

/* ***** */
/* definition des inconnues manquantes */
/* ***** */

/* introduction de variables dans la liste des variables du parseur */

put_inc("r1",1.00*multi);
put_inc("r2",0.4*multi);
put_inc("r3",1.2*multi);

```

```

put_inc("r4",1.03923*multi);
put_inc("t2",0.0);
put_inc("t3",0.0);
put_inc("t4",0.0);

/* variables pour le calcul analytique de la grandeur theta 4 */

rr1 = 1.00*multi;
rr2 = 0.4*multi;
rr3 = 1.2*multi;
rr4 = 1.03923*multi;

/* ***** */
/* Boucle de resolution pour une rotation de 360 degres autour du pivot O1 */
/* ***** */

for(angle2 = 0.0;angle2 <= 2.1 *(pi);angle2 += (pi/36.0))
{
  if (!(modifie_inc("t2",angle2))) return 0;

  rr1 = 1.0      * multi;
  rr2 = 0.40     * multi;
  rr3 = 1.2      * multi;
  rr4 = 1.03923 * multi;

  AA  = sin(angle2);
  BB  = cos(angle2) - rr1/rr2;
  CC  =
    (rr1*rr1 + rr2*rr2 - rr3*rr3 + rr4*rr4)/(2.0*rr2*rr4) - (rr1/rr4) *
    cos(angle2);
  DD  = sqrt(AA*AA + BB*BB - CC*CC);
  phim = 2.0*atan((AA - DD)/(BB + CC));
  phip = 2.0*atan((AA + DD)/(BB + CC));
  phim = phim*180.0/pi;
  phip = phip*180.0/pi;

  rr1 = 1.0*multi - z1;
  rr2 = 0.4*multi + z2;
  rr3 = 1.2*multi + z3;
  rr4 = 1.03923*multi + z4;

  AA2  = sin(angle2);
  BB2  = cos(angle2) - rr1/rr2;
  CC2  =
    (rr1*rr1 + rr2*rr2 - rr3*rr3 + rr4*rr4)/(2.0*rr2*rr4) -
    (rr1/rr4)*cos(angle2);
  DD2  = sqrt(AA2*AA2 + BB2*BB2 - CC2*CC2);
  phim2 = 2.0*atan((AA2 - DD2)/(BB2 + CC2));
  phip2 = 2.0*atan((AA2 + DD2)/(BB2 + CC2));
  phim2 = phim2*180.0/pi;
  phip2 = phip2*180.0/pi;

  rr1 = 1.0*multi;
  rr2 = 0.4*multi;
  rr3 = 1.20*multi;
  rr4 = 1.03923*multi;

  if (phim < 0) phim = 360.0 + phim;
  if (phip < 0) phip = 360.0 + phip;
  if (phim2 < 0) phim2 = 360.0 + phim2;
  if (phip2 < 0) phip2 = 360.0 + phip2;

  /* On calcule les angles gamma (t3) et phi (t4) */

  regulafalsi(f1,"t3",0.000001,200,angle2,&angle3);
  regulafalsi(f2,"t4",0.000001,200,angle2,&angle4);

  d3 = d4 = 0.0;

```

```

/* On calcule les erreurs sur gamma et phi */
eqn2_newton(&d3,"dtheta3",&d4,"dtheta4",last,last1, 0.00001,0.00001,150);
modifie_inc("dtheta3",d3);
modifie_inc("dtheta4",d4);

printf("%3.5lf\n",angle2*180.0/pi); /* theta 2 input */
printf("%1f\n",phim); /* theta 4 analytique */
printf("%1f\n",phim2); /* theta 4 analytique */
printf("%1f\n",phim-phim2);
printf("%1f\n",angle4*180.0/pi); /* theta 4 experime */
printf("%3.5lf \n\n",d4*180.0/pi); /* delta theta 4 */
}
endtorseur();
}

```

II.2 Descriptions des fonctions principales du calculateur symbolique

Il existe un grand nombre de fonctions dans les bibliothèques du logiciel de calcul symbolique. Toutefois, plusieurs d'entre-elles ne sont que des fonctions intermédiaires employées par le logiciel lui-même. De ce fait, la liste qui suit regroupe que les fonctions susceptibles d'être employées par les utilisateurs éventuels. Cette liste est divisée en trois catégories: les fonctions associées aux opérations sur les torseurs (sous-section II.2.1), les fonctions associées directement aux opérations sur les vecteurs (sous-section II.2.2) et finalement, les autres fonctions qui ne peuvent être catégorisées dans les deux autres catégories (sous-section II.2.3).

II.2.1 Fonctions associées aux opérations sur les torseurs

Opérations

int inittorseur(char *fichier);

Cette fonction effectue l'initialisation de la liste globale des torseurs (symbolique et numérique). C'est dans cette liste globale que toutes les informations sont stockées (TT). L'appel de cette fonction est obligatoire avant de réaliser toute autre opération sur les torseurs. La fonction reçoit en paramètre le nom du fichier s'il y a lieu. Elle retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int endtorseur(void);

Cette fonction détruit la liste globale contenant l'ensemble des torseurs. Cette liste a préalablement été initialisée par la fonction **inittorseur**. Aucun paramètre est nécessaire lors de l'appel de cette fonction en effet, toutes les informations sont transmises par l'intermédiaire de variables globales. Elle retourne l'entier:

1 : en cas de succès;
 0 : en d'échec.

int scalaire_torseur(char *T1, char *var, double num, char *T2);

Cette fonction permet de multiplier un scalaire à un torseur. Pour ce faire, elle doit recevoir en paramètre le nom du torseur (T1) sur lequel le scalaire est multiplié, le nom symbolique de la variable scalaire (var), la valeur numérique de ce même scalaire (num) et finalement, le nom du torseur de destination (T2). La fonction retourne l'entier:

1 : en cas de succès;
 0 : en cas d'échec.

int soustraction_torseur(char *T1, char *T2, char *T3);

Cette fonction gère la soustraction de deux torseurs (partie symbolique et partie numérique). Pour réaliser cette opération, la fonction doit recevoir en paramètre le torseur sur lequel la soustraction est réalisée (T1), le torseur qui doit être soustrait (T2) et le torseur de destination (T3). Évidemment, d'autres informations sont transmises à la fonction par variables globales. Elle retourne le nouveau torseur (T3) et l'entier:

1 : en cas de succès;
 0 : en cas d'échec.

int addition_torseur(char *T1, char *T2, char *T3);

Cette fonction gère l'addition de deux torseurs (partie symbolique et partie numérique). Pour réaliser cette opération, la fonction doit recevoir en paramètre le torseur sur lequel l'addition est réalisée (T1), le torseur qui doit être additionné (T2) et le torseur de destination (T3). D'autres informations sont transmises à la fonction par variables globales. Elle retourne le nouveau torseur (T3) et l'entier:

1 : en cas de succès;
 0 : en cas d'échec.

int CR(char *T1, char *vecteur, char *T2);

Cette fonction gère les changements de référentiel des torseurs. Elle emploie les outils mathématiques présentés dans ce mémoire. La fonction **CR** doit donc recevoir le nom du torseur qui subit le changement de référentiel (T1), le vecteur déplacement (vecteur) exprimé sous la forme de la symbolique rotation/translation qui a été présenté au chapitre 3.0 et le torseur résultant du changement de référentiel (T2). La fonction reçoit également la liste globale contenant toutes les informations relatives aux torseurs. Elle retourne le torseur résultant (T2) avec ces nouvelles valeurs par l'intermédiaire de cette liste globale et l'entier:

-1 : en cas d'échec;
0 : en cas de succès.

int TRANSPORT(char *T1, char *vecteur, char *T2);

Cette fonction gère le transport d'un torseur. Elle emploie les outils mathématiques présentés dans ce mémoire. Pour réaliser un transport, la fonction doit recevoir en paramètre le nom du torseur à transporter (T1), le vecteur (vecteur) liant l'origine du torseur T1 au point où doit s'exprimer ce dernier après son transport et finalement, le nom du nouveau torseur qui est généré (T2). La fonction reçoit également la liste globale contenant toutes les informations relatives aux torseurs. Elle retourne le résultat du transport par l'intermédiaire de cette même liste globale et l'entier:

-1 : en cas d'échec;
0 : en cas de succès.

Manipulations intermédiaires

int getval_moment(char *T1, double *val1, double *val2, double *val3);

Cette fonction extrait la composante numérique du moment de la résultante du torseur (T1). Ainsi, pour accomplir sa tâche, la fonction doit recevoir en paramètre le nom du torseur (T1). Elle doit également obtenir la liste globale définissant l'ensemble

des torseurs. La fonction retourne les trois composantes numériques du vecteur moment associé au torseur T1 soit val1 pour la composante u, val2 pour la composante v et val3 pour la composante w. De plus, elle retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int getval_resultante(char *T1, double *val1, double *val2, double *val3);

Cette fonction extrait la composante numérique de la résultante du torseur (T1). Ainsi, pour accomplir sa tâche, la fonction doit recevoir en paramètre le nom du torseur (T1). Elle doit également obtenir la liste globale définissant l'ensemble des torseurs. La fonction retourne les trois composantes numériques du vecteur résultante associé au torseur T1 soit val1 pour la composante u, val2 pour la composante v et val3 pour la composante w. De plus, elle retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int getsym_moment(char *t, char **mom);

Cette fonction extrait la composante symbolique du moment de la résultante du torseur (t). Ainsi, pour accomplir sa tâche, la fonction doit recevoir en paramètre le nom du torseur (t). Elle doit également obtenir la liste globale définissant l'ensemble des torseurs. La fonction retourne l'expression symbolique du vecteur moment (mom) et l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int getsym_resultante(char *t, char **res);

Cette fonction extrait la composante symbolique de la résultante du torseur (t). Ainsi, pour accomplir sa tâche, la fonction doit recevoir en paramètre le nom du torseur (t). Elle doit également obtenir la liste globale définissant l'ensemble des torseurs. La

fonction retourne l'expression symbolique du vecteur résultante (res) et l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int init_calcul(char *fichier);

Cette fonction a pour tâche de préparer l'ensemble des listes de variables et paramètres qui sont utilisés lors de la résolution numérique des équations symboliques. Ainsi, elle réalise la création d'une liste chaînée contenant les variables symboliques et les valeurs numériques associées. Une certaine part de ces informations est extraite d'un fichier dont le nom (fichier) est reçu en paramètre. Cette fonction doit impérativement être appelée avant toute demande de résolution. Elle retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

Le format du fichier de lecture doit être le suivant:

```
variable1 1.0
variable2 2.0
variable3 ...
...
$$
```

Remarquons que le symbole "\$\$" doit être donné à la fin du fichier.

int put_inc(char *var,double num);

Cette fonction permet d'introduire une variable symbolique et la valeur numérique associée, dans la liste chaînée contenant tous les paramètres. Elle peut être appelée lorsque la fonction d'initialisation `init_calcul` a été exécutée avec succès. La fonction reçoit le nom symbolique de la variable (var) et sa valeur numérique associée (num). Elle utilise également la liste globale regroupant l'ensemble des variables

contenues dans les équations à résoudre. Elle retourne l'entier: -

1 : en cas de succès;
0 : en cas d'échec.

int resoudre(char* t1);

Cette fonction permet de résoudre numériquement un vecteur exprimé sous la forme qui est exigée par le logiciel. Pour réaliser cette résolution, la fonction reçoit le vecteur (t1) qui doit être défini avec les composantes unitaires u,v et w (espace orthonormé direct). Par la suite, elle décompose le vecteur pour les traitements subséquents (passage dans le "parser" numérique). La fonction reçoit également deux listes globales regroupant la liste des variables et la liste des torseurs. Elle retourne le torseur résolu et l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int put_nulle_var(char *var);

Cette fonction permet d'éliminer les variables nulles des équations symboliques. Cette opération permet d'alléger la formulation symbolique de ces équations. La fonction reçoit le nom de la variable (var) en paramètre. De plus, la fonction utilise la liste globale des variables à éliminer. Elle retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int modifie_inc(char *var, double num);

Cette fonction est employée pour modifier la valeur numérique associée à une variable symbolique existante dans la liste globale des variables. La fonction reçoit en paramètre le nom symbolique de la variable (var) et la nouvelle valeur numérique (num) qui doit être associée à ce nom dans la liste globale des variables. Elle retourne l'entier:

1 : en cas de succès;

0 : en cas d'échec.

int vecteur_deplacement(char *transformation,char **vr);

Cette fonction construit un vecteur liant l'origine d'un référentiel défini dans un espace orthonormé direct dont les composantes unitaires sont **u**, **v** et **w** et un point quelconque dans ce même référentiel. La fonction emploie les transformations exponentielles définies au chapitre 3.0 pour réaliser cette tâche. La fonction doit recevoir le parcours du déplacement (**transformation**) à suivre entre les deux points sous un format défini par le logiciel. La fonction retourne le vecteur déplacement (**vr**) et l'entier:

1 : en cas de succès;

0 : en cas d'échec.

Le format du parcours de déplacement contenu dans la variable "**transformation**" est le suivant:

"[expression_1(vecteur,expression_2),...]"

où,

expression_1 : est la transformation géométrique recherchée. Cette expression peut être une "translation" ou une "rotation". La fonction accepte toute contraction de ces mots. Elle accepte autant les majuscules que les minuscules.

vecteur : est la composante unitaire selon laquelle la transformation doit être exécutée. Cette composante peut être **u**, **v** ou **w**.

expression_2 : est une variable définissant l'amplitude de la transformation. L'expression de cette variable est laissée au choix de l'utilisateur.

Exemple d'expression contenue dans la variable **transformation**:

"[trans(v,l),rot(u,alpha)]"

int resoudre_eqn(char *vsym, double *vnum);

Cette fonction calcule l'expression numérique d'une équation définie symboliquement. Pour réaliser cette opération, la fonction emploie les bibliothèques d'un "parser". La fonction reçoit en paramètre l'expression symbolique de l'équation à résoudre (vsym). La liste globale des variables contenues dans les équations est utilisée. La fonction retourne le résultat numérique à l'aide d'une variable (vnum) ainsi que l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int puttorseur(char *T2, char *r, char *m,int flag);

Cette fonction permet d'ajouter un torseur a la liste des torseurs. Si cette liste n'existe pas, elle en effectue l'initialisation. La fonction reçoit en paramètre le nom du torseur à ajouter (T2), le vecteur resultante associé à ce torseur (r), son vecteur moment (m) et un entier qui doit conserver la valeur 1. Les vecteurs moment et résultante doivent être définis selon la norme choisie pour le logiciel. La fonction emploie également la liste globale des torseurs. Elle retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec;
-1 : si le nom du torseur (T2) existe déjà, la fonction abandonne l'opération.

Voici le format qui doit être employé pour définir le vecteur résultante et le vecteur moment:

"({composante_1}[u + {composante_2}[v + {composante_3}[w]"

où,

composante_1, composante_2,

composante_3 : sont les expressions des composantes associées à chacun des vecteurs unitaires du vecteur.

II.2.2 Fonctions associées aux opérations sur les vecteurs

Opérations

int addition_vecteur(char *v1, char *v2, char **vr);

Cette fonction permet l'addition de deux vecteurs exprimés symboliquement. Pour réaliser cette opération, la fonction reçoit un premier vecteur (v1) et un second vecteur (v2). Ces deux vecteurs sont définis symboliquement dans un format propre au logiciel. Le résultat de l'addition est retourné par l'intermédiaire d'une variable (vr). La fonction retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int soustraction_vecteur(char *v1, char *v2, char **vr);

Cette fonction permet la soustraction de deux vecteurs exprimés symboliquement. Pour réaliser cette opération, la fonction reçoit un premier vecteur (v1) et un second vecteur (v2). Ces deux vecteurs sont définis symboliquement dans un format propre au logiciel. Le résultat de la soustraction est retourné par l'intermédiaire d'une variable (vr). La fonction retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int norme_vecteur(char *v1, char **vr);

Cette fonction calcule la norme d'un vecteur. Ce vecteur est défini symboliquement selon un format choisi par convention. La fonction reçoit en paramètre le vecteur (v1) dont la norme est à calculer. Ce résultat est retourné sous une forme symbolique grâce à la variable (vr). De plus, la fonction retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

Voici une description sommaire de la procédure de calcul:

$$((x*x) + (y*y) + (z*z))^{(1/2)} = \text{sqrt}(\text{proscalaire}(\text{vecteur1}, \text{vecteur1})).$$

int scal_mul_vecteur(char *v1, char *scal, char **vr);

Cette fonction permet de multiplier un vecteur et d'un scalaire qui sont définis symboliquement. La fonction reçoit le vecteur (v1) sur lequel le scalaire (scal) doit être multiplié. Le résultat symbolique est retourné par l'intermédiaire de la variable (vr).

La fonction retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int scal_div_vecteur(char *v1, char *scal, char **vr);

Cette fonction permet de diviser symboliquement le scalaire d'un vecteur. La fonction reçoit le vecteur (v1) et le scalaire (scal). Le résultat symbolique est retourné par l'intermédiaire de la variable (vr). La fonction retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int provectoriel(char *v1, char *v2, char **vr, int tidx);

Cette fonction effectue le produit vectoriel entre deux vecteurs. Cette tâche est réalisée à la suite de la réception du premier vecteur (v1), du second vecteur (v2) et d'un entier (tidx). Cet entier permet d'éliminer les signes négatifs (1: pas de signe -, 0: respect de la règle de la main droite). La fonction retourne le vecteur solution (vr) en respectant la convention choisie:

$$.(\{\text{xxxxxxx}\}[u + \{\text{yyyyyyy}\}[v + \{\text{zzzzzzzz}\}][w) .$$

La fonction retourne également l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int proscalaire(char *v1, char *v2, char **vr);

Cette fonction effectue le produit scalaire des deux vecteurs (v1 et v2) reçus en paramètre. Le résultat du produit vectoriel est retourné en paramètre grâce à la variable vr. La fonction retourne également l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int projection_vecteur(char *v1, char *v2, char **vr);

Cette fonction effectue la projection du vecteur v1 sur un second vecteur v2. La fonction retourne le résultat de cette projection grâce à la variable vr. Elle retourne également l'entier:

1 : en cas de succès;
0 : en cas d'échec.

Manipulations intermédiaires

int resoudre_vecteur(char *vsym, char **vnum);

Cette fonction résout numériquement un vecteur grâce à la bibliothèque de fonction du "parser". Pour réaliser cette tâche, la fonction reçoit le vecteur (vsym) sous une forme symbolique définie par convention ($\{composante_1\}[u + \{composante_2\}[v + \{composante_3\}[w]]$). Il est impératif que chacune des variables contenues dans l'expression du vecteur soit associée à une valeur numérique dans la liste globale des variables. Cette liste est reçu par la fonction. Les résultats des calculs sont renvoyés de la fonction grâce à la variable "vnum". La fonction retourne également l'entier:

1 : en cas de succès;
0 : en cas d'échec.

II.2.3 Autres fonctions

int obtention_angles(double t2, char *n1, char *n2, double *angle3, double *angle4);

Cette fonction est employée pour la résolution du générateur d'équations à quatre membres (chapitre 5.0) par la méthode de la sécante. Cette méthode numérique est dérivée de la méthode de Newton/Raphson associée aux équations non-linéaires. Les valeurs des angles θ_3 (angle3) et θ_4 (angle4) sont obtenus connaissant les grandeurs des membrures r_1 , r_2 , r_3 , r_4 et l'angle d'entrée θ_2 (t2). Les variables "n1" et "n2" sont employées dans le fonctionnement interne de la méthode numérique. La fonction retourne l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int eqn2(char *n1, char *n2, char **f2, char x);

Cette fonction est employée lors de la validation de la méthodologie au chapitre 5.0. Elle permet de construire la seconde équation qui est résolue par la méthode de la sécante au cours de la recherche de la valeur numérique de l'angle θ_4 . La fonction reçoit en paramètre le vecteur nominal 1 (n1) et le vecteur nominal 2 (n2). La fonction retourne l'équation à résoudre par la méthode de la sécante (f2) et l'entier:

1 : en cas de succès;
0 : en cas d'échec.

int eqn1(char *n1, char **f1);

Cette fonction génère la première équation pour l'algorithme de la sécante. Elle est employée lors de la validation de la méthodologie (chapitre 5.0). Pour réaliser sa tâche, la fonction reçoit l'équation nominale du chemin de fonctionnalité 1 (n1). Le résultat est retourné par la variable "f1". La fonction retourne également l'entier:

1 : en cas de succès;

0 : en cas d'échec.

int secante(char *f1, char *var, double d1, double d2, double tol, int N, double t2, double *res);

Cette fonction utilise la méthode de la sécante pour résoudre la fonction non-linéaire (f1) qu'elle reçoit (trouve θ_3 ou θ_4). Pour accomplir cette tâche, la fonction reçoit en paramètre l'ensemble des variables de contrôles (d1, d2, tol et N) associées à la méthode numérique soient les deux valeurs de départ, la tolérance à la convergence et le nombre maximum d'itérations. Elle retourne le résultat des calculs s'il y a convergence ou dans le cas contraire, un message d'erreur. La fonction retourne également l'entier:

1 : en cas de succès;
 0 : en cas d'échec;
 2 : en cas de divergence des calculs.

int cherche_borne(char *eqn, char *var, double t2, double *borne1, double *borne2, double precision);

Cette fonction est employée par la fonction **regulafalsi** (méthode "Regula Falsi") pour calculer une borne positive et une borne négative afin d'initier les calculs d'optimisation. La fonction reçoit l'équation symbolique (eqn), la variable symbolique (var), la valeur numérique de l'angle d'entrée θ_2 (t2). La fonction retourne la valeur numérique des bornes min (borne1) et max (borne2) de manière à obtenir: $f(a)f(b) < 0$. La fonction retourne également l'entier:

1 : en cas de succès;
 0 : en cas d'échec.

int regulafalsi(char *eqn, char *var, double tol, int M, double angle2, double *root);

Cette fonction calcule l'inconnue d'une équation non-linéaire par la méthode

REGULA FALSI. Pour accomplir cette tâche, la fonction reçoit la tolérance définissant la convergence (tol), le nombre d'itération maximum (M), l'expression de l'équation non-linéaire (eqn), la représentation symbolique de la variable (var). La fonction retourne la racine de l'équation (root) et l'entier:

- 1 : en cas de succès ("nous avons atteint la précision désirée");
- 0 : en cas d'échec;
- 2 : en cas de succès ("nous avons atteint le nombre maximum d'itération").

int newton(double *xo, char *var, char *eqn, double xterm, double fterm, int N);

Cette fonction effectue la résolution d'une équation non-linéaire par la méthode de Newton. Pour accomplir cette tâche, la fonction reçoit une valeur numérique de départ, des bornes numériques contrôlant la convergence, le nombre d'itération maximum permis (N), l'expression symbolique de l'équation à résoudre, l'expression symbolique de l'inconnue et un index indiquant l'état de la résolution. La fonction retourne l'index d'état et le résultat de l'approximation. La fonction retourne également l'entier:

- 1 : en cas de sortie due à fterm;
- 2 : en cas de sortie due à xterm;
- 3 : en cas de sortie due à N (nombre d'itérations)

int eqn2_newton(double *x0, char *varx, double *y0, char *vary, char *eqn1, char *eqn2, double xterm, double fterm, int N);

Cette fonction effectue la résolution de systèmes de deux équations non-linéaires par la méthode de Newton (version 2). Pour accomplir cette tâche, la fonction reçoit les valeurs numériques de départ (x0 et y0), des valeurs numériques contrôlant l'arrêt des itérations, le nombre d'itérations maximum alloué (N), les expressions symboliques des deux équations (eqn1 et eqn2), les expressions symboliques des deux inconnues recherchées (varx et vary) et un index indiquant l'état de la résolution. La fonction retourne l'index d'état, le résultat de l'approximation et l'entier:

- 1 : en cas de sortie de la fonction due à fterm;
- 2 : en cas de sortie de la fonction due à xterm;
- 3 : en cas de sortie de la fonction due au nombre d'itération (N).

double Fonct(double xo, char *var, char *eqn);

Cette fonction permet d'évaluer numériquement l'expression symbolique d'une équation grâce au "parser". Il est impératif que la liste globale des variables contiennent l'ensemble des variables contenues dans l'équation à résoudre. La fonction reçoit la valeur numérique (xo) et la variable symbolique inconnue associée (var) et l'expression symbolique de l'équation (eqn). La fonction retourne le résultat numérique ou -123456789.45 dans le cas d'un échec de l'évaluation.

double derive1_fonction(double xo, char *var, char *eqn);

Cette fonction évalue numériquement la dérivée d'une fonction ("Richardson extrapolation and numerical differentiation"). Pour accomplir cette tâche, la fonction doit recevoir en paramètre la variable symbolique (var) selon laquelle la dérivée doit être calculée, la valeur numérique de cette variable (xo) et l'équation symbolique qui est à dériver (eqn). La fonction retourne le résultat numérique de la dérivée ou un message indiquant l'échec des calculs ("-123456789.45").

ANNEXE III

TORSEURS EMPLOYÉS LORS DE L'ANALYSE DU VARIATEUR

MÉCANIQUE DE VITESSE

Liste des torseurs

	page
III.1 DEMI-BOUCLE 1	135
III.1.1 Torseurs d'erreurs dimensionnels	135
III.1.2 Tolérances géométriques (demi-boucle 1)	149
III.1.3 JEU (demi-boucle 1)	155
III.2 DEMI - BOUCLE 2	157
III.2.1 Torseurs d'erreurs dimensionnelles	157
III.2.2 Torseurs d'erreurs géométriques (demi-boucle 2)	160
III.3 BACKLASH	163

III.1 DEMI-BOUCLE 1

III.1.1 Torseurs d'erreurs dimensionnels

{T3} (référentiel 0)

$$\begin{Bmatrix} -X3 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-L3), T(u,-L4), T(u,-L5), T(u,-jeu1), T(u,-L6), T(u,L8), T(u,L9), T(u,L11),
T(u,-L13), T(u,-jeu2), T(u,-L15), T(u,-L16), T(u,-L17), T(u,-L18), T(u,-L19),
T(u,L21), T(u,-L100), T(v,r3)

$$d3 = (dx3)u + (dy3)v + (dz3)w$$

$$dx3 = -L3 - L4 - L5 - jeu1 - L6 + L8 + L9 + L11 - L13 - jeu2 - L15 - L16 - \\ L17 - L18 - L19 + L21 - L100$$

$$dy3 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X3 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T4} (référentiel 2)

$$\begin{Bmatrix} -X4 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-L4), T(u,-L5), T(u,-jeu1), T(u,-L6), T(u,L8), T(u,L9), T(u,L11), T(u,-L13),
T(u,-jeu2), T(u,-L15), T(u,-L16), T(u,-L17), T(u,-L18), T(u,-L19), T(u,L21),
T(u,-L100), T(v,r3)

$$d4 = (dx4)u + (dy4)v + (dz4)w$$

$$dx4 = -L4 - L5 - jeu1 - L6 + L8 + L9 + L11 - L13 - jeu2 - L15 - L16 - L17 - \\ L18 - L19 + L21 - L100$$

$$dy4 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X4 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T5} (référentiel 5)

$$\begin{Bmatrix} -X5 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-L5), T(u,-jeu1), T(u,-L6), T(u,L8), T(u,L9), T(u,L11), T(u,-L13), T(u,-jeu2),
T(u,-L15), T(u,-L16), T(u,-L17), T(u,-L18), T(u,-L19), T(u,L21), T(u,-L100),
T(v,r3)

$$d5 = (dx5)u + (dy5)v + (dz5)w$$

$$dx5 = -L5 - \text{jeu1} - L6 + L8 + L9 + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - \\ L19 + L21 - L100$$

$$dy5 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X5 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T6} (référentiel 7)

$$\begin{Bmatrix} -X6 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-L6), T(u,L8), T(u,L9), T(u,L11), T(u,-L13), T(u,-jeu2), T(u,-L15), T(u,-L16),
T(u,-L17), T(u,-L18), T(u,-L19), T(u,L21), T(u,-L100), T(v,r3)

$$d6 = (dx6)u + (dy6)v + (dz6)w$$

$$dx6 = -L6 + L8 + L9 + L11 - L13 - jeu2 - L15 - L16 - L17 - L18 - L19 + L21 \\ - L100$$

$$dy6 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X6 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T8} (référentiel 8)

$$\begin{Bmatrix} dx8 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,L8), T(u,L9), T(u,L11), T(u,-L13), T(u,-jeu2), T(u,-L15), T(u,-L16), T(u,-L17),
T(u,-L18), T(u,-L19), T(u,L21), T(u, -L100), T(v,r3)

$$d8 = (dx8)u + (dy8)v + (dz8)w$$

$$dx8 = L8 + L9 + L11 - L13 - \text{jeu2} - L15 - L16 - L17 - L18 - L19 + L21 - L100$$

$$dy8 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} dx8 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T9} (référentiel 4)

$$\begin{Bmatrix} x_9 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,L9), T(u,L11), T(u,-L13), T(u,-jeu2), T(u,-L15), T(u,-L16), T(u,-L17),
T(u,-L18), T(u,-L19), T(u,L21), T(u,-L100), T(v,r3)

$$d_9 = (dx_9)u + (dy_9)v + (dz_9)w$$

$$dx_9 = L_9 + L_{11} - L_{13} - \text{jeu2} - L_{15} - L_{16} - L_{17} - L_{18} - L_{19} + L_{21} - L_{100}$$

$$dy_9 = r_3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} x_9 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T11} (référentiel 15)

$$\begin{Bmatrix} x_{11} & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,L11), T(u,-L13), T(u,-jeu2), T(u,-L15), T(u,-L16), T(u,-L17), T(u,-L18),
T(u,-L19), T(u,L21), T(u,-L100), T(v,r3)

$$d_{11} = (dx_{11})u + (dy_{11})v + (dz_{11})w$$

$$dx_{11} = L_{11} - L_{13} - \text{jeu2} - L_{15} - L_{16} - L_{17} - L_{18} - L_{19} + L_{21} - L_{100}$$

$$dy_{11} = r_3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} x_{11} & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T13} (référentiel 9)

$$\begin{Bmatrix} -X13 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-L13), T(u,-jeu2), T(u,-L15), T(u,-L16), T(u,-L17), T(u,-L18), T(u,-L19),
T(u,L21), T(u,-L100), T(v,r3)

$$\mathbf{d13} = (\mathbf{dx13})\mathbf{u} + (\mathbf{dy13})\mathbf{v} + (\mathbf{dz13})\mathbf{w}$$

$$\mathbf{dx13} = -\mathbf{L13} - \mathbf{jeu2} - \mathbf{L15} - \mathbf{L16} - \mathbf{L17} - \mathbf{L18} - \mathbf{L19} + \mathbf{L21} - \mathbf{L100}$$

$$\mathbf{dy13} = \mathbf{r3}$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X13 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T15} (référentiel 11)

$$\begin{Bmatrix} -X15 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-L15), T(u,-L16), T(u,-L17), T(u,-L18), T(u,-L19), T(u, L21), T(u,-L100),
T(v,r3)

$$d15 = (dx15)u + (dy15)v + (dz15)w$$

$$dx15 = -L15 - L16 - L17 - L18 - L19 + L21 - L100$$

$$dy15 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X15 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T16} (référentiel 13)

$$\begin{Bmatrix} -X16 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-L16), T(u,-L17), T(u,-L18), T(u,-L19), T(u,L21), T(u,-L100), T(v,r3)

$$d16 = (dx16)u + (dy16)v + (dz16)w$$

$$dx16 = -L16 - L17 - L18 - L19 + L21 - L100$$

$$dy16 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X16 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T17} (référentiel 14)

$$\begin{Bmatrix} -X17 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-L17), T(u,-L18), T(u,-L19), T(u,L21), T(u,-L100), T(v, r3)

$$\begin{aligned} d17 &= (dx17)u + (dy17)v + (dz17)w \\ dx17 &= -L17 - L18 - L19 + L21 - L100 \\ dy17 &= r3 \end{aligned}$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X17 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T18} (référentiel 17)

$$\begin{Bmatrix} -X18 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

$T(u, -L18), T(u, -L19), T(u, L21), T(u, -L100), T(v, r3)$

$$d18 = (dx18)u + (dy18)v + (dz18)w$$

$$dx18 = -L18 - L19 + L21 - L100$$

$$dy18 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X18 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T19} (référentiel 18)

$$\begin{Bmatrix} -X19 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-L19), T(u,L21), T(u,-L100), T(v,r3)

$$d19 = (dx19)u + (dy19)v + (dz19)w$$

$$dx19 = -L19 + L21 - L100$$

$$dy19 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X19 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T21} (référentiel 19)

$$\begin{Bmatrix} x_{21} & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

$T(u, L_{21}), T(u, -L_{100}), T(v, r_3)$

$$d_{21} = (dx_{21})u + (dy_{21})v + (dz_{21})w$$

$$dx_{21} = L_{21} - L_{100}$$

$$dy_{21} = r_3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} x_{21} & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

III.1.2 Tolérances géométriques (demi-boucle 1)

{Tgé0 1} (référentiel 1 - identificateur (ii) sur les plans)

$$\begin{Bmatrix} 0 & 0 \\ Y_{g1} & \beta_{g1} \\ Z_{g1} & \gamma_{g1} \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-e2/2), T(u,L9), T(u,L11), T(u,-L13), T(u,-jeu2), T(u,-L15), T(u,-L16),
T(u,-L17), T(u,-L18), T(u,-L19), T(u,L21), T(u,-L100), T(v,r3)

$$dgé01 = (dx_{g1})u + (dy_{g1})v + (dz_{g1})w$$

$$dx_{g1} = (-e2/2 + L9) + L11 - L13 - jeu2 - L15 - L16 - L17 - L18 - L19 + L21 \\ - L100$$

$$dy_{g1} = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -\gamma_{g1}dy & 0 \\ Y_{g1} + \gamma_{g1}dx_{g1} & \beta_{g1} \\ Z_{g1} - \beta_{g1}dx_{g1} & \gamma_{g1} \end{Bmatrix}$$

{Tgé0 3} (référentiel 3 - identificateur (iv) sur les plans)

$$\begin{Bmatrix} 0 & 0 \\ Y_{g3} & \beta_{g3} \\ Z_{g1} & \gamma_{g3} \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,-e1/2), T(u,L9), T(u,L11), T(u,-L13), T(u,-jeu2), T(u,-L15), T(u,-L16),
T(u,-L17), T(u,-L18), T(u,-L19), T(u,L21), T(u,-L100), T(v,r3)

$$dgé03 = (dxg3)u + (dyg3)v + (dzg3)w$$

$$dxg3 = (-e1/2 + L9) + L11 - L13 - jeu2 - L15 - L16 - L17 - L18 - L19 + L21 \\ - L100$$

$$dyg3 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -\gamma_{g3}dyg3 & 0 \\ Y_{g3} + \gamma_{g3}dxg3 & \beta_{g3} \\ Z_{g3} - \beta_{g3}dxg3 & \gamma_{g3} \end{Bmatrix}$$

{Tgéol2} (référentiel 12 - identificateur (vi) sur les plans)

$$\begin{Bmatrix} 0 & 0 \\ Y_{g12} & \beta_{g12} \\ Z_{g12} & \gamma_{g12} \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

$T(u, e_{12}/2)$, $T(u, -L_{19})$, $T(u, L_{21})$, $T(u, -L_{100})$, $T(v, r_3)$

$$dgéol2 = (dx_{g12})u + (dy_{g12})v + (dz_{g12})w$$

$$dx_{g12} = (e_{12}/2) - L_{19} + L_{21} - L_{100}$$

$$dy_{g12} = r_3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -\gamma_{g12}dy_{g12} & 0 \\ Y_{g12} + \gamma_{g12}dx_{g12} & \beta_{g12} \\ Z_{g12} - \beta_{g12}dx_{g12} & \gamma_{g12} \end{Bmatrix}$$

{Tgé0 20} (référentiel 20 - identificateur (v) sur les plans) -

$$\begin{Bmatrix} 0 & 0 \\ Y_{g20} & \beta_{g20} \\ Z_{g20} & \gamma_{g20} \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

$T(u, e4/2), T(u, L21), T(u, -L100), T(v, r3)$

$$dgé020 = (dxg20)u + (dyg20)v + (dzg20)w$$

$$dxg20 = e4/2 + L21 - L100$$

$$dyg20 = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -\gamma_{g20}dyg20 & 0 \\ Y_{g20} + \gamma_{g20}dxg20 & \beta_{g20} \\ Z_{g20} - \beta_{g20}dxg20 & \gamma_{g20} \end{Bmatrix}$$

{Tgé0 16} (référentiel 16 - identificateur (iii) sur les plans)

$$\begin{Bmatrix} 0 & 0 \\ Y_{g16} & \beta_{g16} \\ Z_{g16} & \gamma_{g16} \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u, e3/2), T(u, L11), T(u, -L13), T(u, -jeu2), T(u, -L15), T(u, -L16), T(u, -L17),
T(u, -L18), T(u, -L19), T(u, L21), T(u, -L100), T(v, r3)

$$dgé016 = (dx_{g16})u + (dy_{g16})v + (dz_{g16})w$$

$$dx_{g16} = e3/2 + L11 - L13 - jeu2 - L15 - L16 - L17 - L18 - L19 + L21 - L100$$

$$dy_{g16} = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -\gamma_{g16}dy_{g16} & 0 \\ Y_{g16} + \gamma_{g16}dx_{g16} & \beta_{g16} \\ Z_{g16} - \beta_{g16}dx_{g16} & \gamma_{g16} \end{Bmatrix}$$

{Tgé0 21} (référentiel 21 - identificateur (vii) sur les plans)

$$\begin{Bmatrix} 0 & 0 \\ Y_{g21} & \beta_{g21} \\ Z_{g21} & \gamma_{g21} \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

$T(u, e_{21/2}), T(u, L_{20}), T(u, L_{21}), T(u, -L_{100}), T(v, r_3)$

$$dgé021 = (dx_{g21})u + (dy_{g21})v + (dz_{g21})w$$

$$dx_{g21} = e_{21/2} + L_{20} + L_{21} - L_{100}$$

$$dy_{g21} = r_3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -\gamma_{g21}dy_{g21} & 0 \\ Y_{g21} + \gamma_{g21}dx_{g21} & \beta_{g21} \\ Z_{g21} - \beta_{g21}dx_{g21} & \gamma_{g21} \end{Bmatrix}$$

III.1.3 JEU (demi-boucle 1)

{Tjeu1} (référentiel 6)

$$\begin{Bmatrix} -Xj1 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u, -jeu1), T(u, L6), T(u, L8), T(u, L9), T(u, L11), T(u, -L13), T(u, -jeu2),
T(u, -L15), T(u, -L16), T(u, -L17), T(u, -L18), T(u, -L19), T(u, L21), T(u, -L100),
T(v, r3)

$$d_{jeu1} = (dx_{jeu1})u + (dy_{jeu1})v + (dz_{jeu1})w$$

$$dx_{jeu1} = -jeu1 - L6 + L8 + L9 + L11 - L13 - jeu2 - L15 - L16 - L17 - L18 \\ - L19 + L21 - L100$$

$$dy_{jeu1} = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -Xj1 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

{T_{jeu2}} (référentiel 10)

$$\begin{Bmatrix} -X_{j2} & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u, -jeu2), T(u, -L15), T(u, -L16), T(u, -L17), T(u, -L18), T(u, -L19), T(u, L21),
T(u, -L100), T(v, r3)

$$d_{jeu2} = (dx_{jeu2})u + (dy_{jeu2})v + (dz_{jeu2})w$$

$$dx_{jeu2} = -jeu2 - L15 - L16 - L17 - L18 - L19 + L21 - L100$$

$$dy_{jeu2} = r3$$

Changement de référentiel:

Aucun

Résultat:

$$\begin{Bmatrix} -X_{j2} & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

III.2 DEMI - BOUCLE 2

III.2.1 Torseurs d'erreurs dimensionnelles

{T51} (référentiel 23)

$$\begin{Bmatrix} x_{51} & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u, L51), T(u, L54), T(u, L55), T(v, r2)

$$d_{51} = (dx_{51})u + (dy_{51})v + (dz_{51})w$$

$$dx_{51} = L_{51} + L_{54} + L_{55}$$

$$dy_{51} = r_2$$

Changement de référentiel:

R(w, λ)

Résultat:

$$\begin{Bmatrix} x_{51} & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

Après changement de référentiel:

$$\begin{Bmatrix} X51 \cos(\lambda) & 0 \\ -X51 \sin(\lambda) & 0 \\ 0 & 0 \end{Bmatrix}$$

{T54} (référentiel 24)

$$\begin{Bmatrix} X54 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

$T(u, L54), T(u, L55), T(v, r2)$

$$d54 = (dx54)u + (dy54)v + (dz54)w$$

$$dx54 = L54 + L55$$

$$dy54 = r2$$

Changement de référentiel:

$R(w, \lambda)$

Résultat:

$$\begin{Bmatrix} X54 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

Après changement de référentiel:

$$\begin{Bmatrix} X54 \cos(\lambda) & 0 \\ -X54 \sin(\lambda) & 0 \\ 0 & 0 \end{Bmatrix}$$

{T52} (référentiel 25)

$$\begin{Bmatrix} X52 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u, L55), T(v, r2)

$$d52 = (dx52)u + (dy52)v + (dz52)w$$

$$dx52 = L55$$

$$dy52 = r2$$

Changement de référentiel:

R(w,λ)

Résultat:

$$\begin{Bmatrix} X52 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$$

Après changement de référentiel:

$$\begin{Bmatrix} X52\cos(\lambda) & 0 \\ -X52\sin(\lambda) & 0 \\ 0 & 0 \end{Bmatrix}$$

III.2.2 Torseurs d'erreurs géométriques (demi-boucle 2)

{Tgé_o 26} (référentiel 26 - identificateur (ix) sur les plans)

$$\begin{Bmatrix} 0 & 0 \\ Y_{g26} & \beta_{g26} \\ Z_{g20} & \gamma_{g26} \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(u,e8/2), T(u,L51), T(u,L54), T(u,L55), T(v,r2)

$$dgé_{o26} = (dx_{g26})u + (dy_{g26})v + (dz_{g26})w$$

$$dx_{g26} = e8/2 + L51 + L54 + L55$$

$$dy_{g26} = r2$$

Changement de référentiel:

R(w,λ)

Résultat:

$$\begin{Bmatrix} -\gamma_{g26} dy_{g26} & 0 \\ Y_{g26} + \gamma_{g26} dx_{g26} & \beta_{g26} \\ Z_{g26} - \beta_{g26} dx_{g26} & \gamma_{g26} \end{Bmatrix}$$

Après changement de référentiel:

$$\begin{Bmatrix} (-\gamma_{g26} dy_{g26} \cos \lambda - (Y_{g26} + \gamma_{g26} dx_{g26}) \sin(\lambda)) & -(\beta_{g26}) \sin(\lambda) \\ (-\gamma_{g26} dy_{g26} \sin \lambda + (Y_{g26} - \gamma_{g26} dx_{g26}) \cos(\lambda)) & +(\beta_{g26}) \cos(\lambda) \\ Z_{g26} - \beta_{g26} dx_{g26} & \gamma_{g26} \end{Bmatrix}$$

{Tgéomé 27} (référentiel 27 - identificateur (x) sur les plans)

$$\begin{Bmatrix} 0 & 0 \\ Y_{g27} & \beta_{g27} \\ Z_{g27} & \gamma_{g27} \end{Bmatrix}$$

note : Le système de contraintes est défini au chapitre 6.0

Transport à la coupure:

T(v, r2)

$$dgéomé27 = (dx_{g27})u + (dy_{g27})v + (dz_{g27})w$$

$$dx_{g27} = 0$$

$$dy_{g27} = r2$$

Changement de référentiel:

R(w, λ)

Résultat:

$$\begin{pmatrix} -\gamma_{g27} dy_{g27} & 0 \\ Y_{g27} + \gamma_{g27} dx_{g27} & \beta_{g27} \\ Z_{g27} - \beta_{g27} dx_{g27} & \gamma_{g27} \end{pmatrix}$$

Après changement de référentiel:

$$\begin{pmatrix} (-\gamma_{g27} dy_{g27} \cos \lambda - (Y_{g27} + \gamma_{g27} dx_{g27}) \sin(\lambda)) & -(\beta_{g27} \sin(\lambda)) \\ (-\gamma_{g27} dy_{g27} \sin \lambda + (Y_{g27} - \gamma_{g27} dx_{g27}) \cos(\lambda)) & +(\beta_{g27} \cos(\lambda)) \\ Z_{g27} - \beta_{g27} dx_{g27} & \gamma_{g27} \end{pmatrix}$$

III.3 BACKLASH

$$\mathbf{B} = (\text{back1})\mathbf{u} + (\text{back2})\mathbf{v} + (\text{back3})\mathbf{w}$$