



HAL
open science

Vers une assistance à la synthèse de dispositifs technologiques assurant une liaison mécanique

Cédric Masclet

► **To cite this version:**

Cédric Masclet. Vers une assistance à la synthèse de dispositifs technologiques assurant une liaison mécanique. Mécanique [physics.med-ph]. INSA de Toulouse, 2002. Français. NNT: . tel-00008049

HAL Id: tel-00008049

<https://theses.hal.science/tel-00008049>

Submitted on 13 Jan 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Numéro d'ordre : 670

THÈSE

présentée devant

L'Institut National des Sciences Appliquées de Toulouse

en vue de l'obtention du

DOCTORAT DE L'INSAT

Spécialité : GÉNIE MÉCANIQUE

(arrêté en date du 30 mars 1992)

par

Cédric MASCLET

Vers une Assistance à la Synthèse de Dispositifs Technologiques Assurant une Liaison Mécanique

Soutenue publiquement le 18 décembre 2002 devant le jury composé de :

M. Christian BES	Professeur d'université (Toulouse III)	Examineur
M. Claude BOHATIER	Professeur d'université (Montpellier II)	Rapporteur
M. Alain DAIDIÉ	Maître de conférences (INSA de Toulouse)	Examineur
<u>M. Jean GUILLOT</u>	Professeur d'université (INSA de Toulouse)	Directeur de thèse
M. Daniel PLAY	Directeur scientifique de Federal-Mogul	Rapporteur
M. Marc SARTOR	Professeur d'université (INSA de Toulouse)	Examineur

Laboratoire de Génie Mécanique de Toulouse (LGMT)

Département de Génie Mécanique, INSA Toulouse, 135 Avenue de Ranguéil
31077 TOULOUSE, FRANCE, Tel : 05.61.55.97.18 Fax: 05.61.55.97.00

E-mail : Cedric.Masclet@insa-tlse.fr

NOM : MASCLET

Prénom : Cédric

VERS UNE ASSISTANCE À LA SYNTHÈSE DE DISPOSITIFS TECHNOLOGIQUES
ASSURANT UNE LIAISON MÉCANIQUE
202 pages

Thèse de doctorat (arrêté en date du 30 mars 1992), Spécialité Génie Mécanique
INSA de Toulouse, 2002, n° d'ordre 670

RÉSUMÉ

Ce travail propose une méthode de conception des mécanismes basée sur une décomposition hiérarchique des tâches de synthèse. Parmi les tâches identifiées, nous nous intéressons à celle qui effectue le choix et le dimensionnement optimal des « Dispositifs Technologiques assurant les Liaisons Mécaniques ». Ces DTLM sont des features de haut niveau qui possèdent tous les attributs nécessaires à un traitement indépendant. La recomposition du système peut cependant révéler des conflits entre les liaisons. La synthèse d'ensemble est exprimée sous la forme d'un problème d'optimisation hiérarchique pour lequel nous proposons plusieurs méthodes de résolution dont une méthode d'optimisation collaborative. Nous présentons un outil d'assistance qui, couplé à un logiciel de CAO commercial, nous a permis de valider les concepts. Des exemples illustrent notre propos.

MOTS CLÉS :

conception intégrée, CAO, synthèse, optimisation collaborative, structure objet, liaison mécanique, features, relaxation sélective.

JURY : Soutenue le 18/12/2002 à l'INSA de Toulouse devant un jury composé de :

Président :	M. Christian BES	Professeur à l'université de Toulouse.
Membres :	M. Claude BOHATIER	Professeur à l'université de Montpellier.
	Alain DAIDIÉ	Maître de Conférences à l'INSA de Toulouse.
	Jean GUILLOT	Professeur à l'INSA de Toulouse.
	Daniel PLAY	Directeur scientifique de Federal Mogul.
	Marc SARTOR	Professeur à l'INSA de Toulouse.

Dépôt à la bibliothèque universitaire en 4 exemplaires.

Remerciements

Je tiens en premier lieu à remercier MM. Claude BOHATIER et Daniel PLAY pour l'honneur qu'ils m'ont fait en acceptant de rapporter sur ces travaux de thèse.

Je suis profondément reconnaissant à M. Christian BES d'avoir assumé la délicate tâche de présider le jury de soutenance.

Mes sincères remerciements vont à M. Jean GUILLOT qui m'a accueilli au sein de son équipe de recherche. J'ai, grâce à lui, découvert une autre vision (ainsi que l'esprit) de la conception mécanique. Qu'il sache à quel point j'ai pu m'enrichir à son contact, et espère avoir encore l'occasion de le faire.

Je veux exprimer toute ma gratitude à M. Marc SARTOR. Il est pour moi le symbole du dévouement et de l'abnégation. Toujours disponible, il a dirigé nos travaux avec une sérénité que je lui envie. A ses côtés, j'apprends quotidiennement le métier d'enseignant-chercheur et je mesure la chance de pouvoir profiter d'un tel exemple.

Mes sincères remerciements vont à Alain DAIDIÉ pour son soutien, sa convivialité ainsi que son expertise dans le domaine de la CAO.

Je salue le personnel du département de génie mécanique. Les secrétaires méritent un éloge particulier, et je m'exécute donc. Mesdames, vos sourires et votre bonne humeur illuminent les journées les plus grises. Je vous renouvelle mes hommages.

L'équipe COSAM est une petite famille. Je voudrais saluer les doctorants passés, présents et à venir pour la chaleureuse ambiance qui règne au laboratoire, parfois bien tard dans la nuit : Feras Alkatan, Hazem Aziz, El Hassan Benkhira, Mohammed Boutaïb, Bertrand Doucet, Jean-Christophe Fauroux, Xavier Hernot, Hassan Lakiss, Dimitri Leray, Renaud Lavabre, Manuel Martinez-Martinez, Carlos Martins, Manuel Paredes, Yves Silberberg, Jean-Christophe Wahl...

Merci également aux étudiants stagiaires (DUT, DEA, ingénieurs) qui ont œuvré sur ce projet : Antoine, Bruno, Djamel, Ivan, Jasmin, Olivier, Patrice et Yann. Ceci est en partie votre œuvre.

Je salue la communauté des logiciels libres, les programmeurs de génie et autres bidouilleurs fous qui, sans relâche, conçoivent et améliorent la qualité d'outils tels que \LaTeX , Lyx, Xplot, Tcl-Tk, Xfig, OpenOffice, la librairie OPTSOLVE++, etc... dont j'ai usé, et parfois abusé! Toute ma gratitude à Jérôme pour son initiation aux systèmes "X" ainsi qu'à Jean-Christophe pour m'avoir fait découvrir Tcl-Tk et Linux.

Ces remerciements seraient incomplets sans évoquer mon entourage qui régulièrement s'est inquiété de la progression des travaux. Merci à tous pour votre soutien dans les moments difficiles.

Enfin, je remercie Maud grâce à qui se travail a abouti. Ses encouragements réguliers ont maintes fois relancer la machine. Il s'agit également d'Erin dont la venue toute récente m'a pourvu d'un nouvel élan au moment le plus ultime. J'ai puisé mon énergie dans votre amour. Merci.

À mon grand-père

Introduction

DEVANT la complexité croissante des systèmes mécaniques et dans un contexte de compétitivité industrielle accrue, la conception est incontestablement une étape stratégique dans la vie d'un produit. L'homme cherche donc naturellement à tirer profit de toutes les aides dont il pourrait bénéficier durant cette phase afin de réduire les délais d'études, abaisser les coûts. . . L'assistance informatique est une des voies permettant d'atteindre ces objectifs. Par le biais de la conception intégrée, le concepteur se voit désormais proposer une palette d'outils d'assistance ainsi qu'une certaine automatisation des tâches de conception. Dès les années 1980, Jean Guillot et ses collaborateurs du Laboratoire de Génie Mécanique de Toulouse (LGMT) ont développé un logiciel d'assistance à la conception intégrée de mécanismes. Ces travaux, qui donnèrent lieu à une thèse d'état [Gui87], concrétisaient une longue réflexion menée sur la conception mécanique et sur l'apport considérable que pouvait constituer l'informatique dans ce domaine. Les années ont passé et aux outils initialement testés, se sont substituées des méthodes d'intelligence artificielles [Sil94] [Dam96], de nouveaux langages, sans pour cela parvenir à un résultat suffisamment satisfaisant. Le travail de thèse présenté dans ce mémoire a pour but de poursuivre ces travaux. C'est l'occasion pour nous de reconsidérer, avec un certain recul, le type d'assistance vers lequel nous devons tendre, et de juger de l'apport des nouvelles technologies de l'informatique dans ce domaine.

Ce mémoire présente nos réflexions sur le sujet de la conception et les idées que nous avons développées concernant son organisation, les éléments manipulés et les méthodes employées. Il présente également la maquette logicielle qui nous a permis de confronter la théorie à la pratique. Sa composition est la suivante :

Chapitre 1 : Nous abordons, dans ce premier chapitre, la description du processus de conception. L'analyse de la phase de conception nous conduit à nous interroger sur les points stratégiques des phases de synthèse et d'analyse qui la composent. Après une brève présentation du type de mécanisme servant de support à notre étude, nous nous consacrons plus profondément à la phase de synthèse. Celle-ci fait appel à des modèles de représentation du produit ainsi qu'à des modèles organisationnels divers que nous recensons. Nous définissons alors une organisation opérationnelle, basée sur un découpage fonctionnel multiniveau, sur laquelle s'appuie la suite de nos travaux. À partir de cette organisation, nous nous intéressons au type d'assistance que nous pourrions fournir au concepteur et la forme qu'elle pourrait prendre. Nous abordons ainsi la problématique de l'assistance à la conception et passons en revue quelques systèmes intégrés de conception déjà proposés dans ce domaine, avant de préciser les grandes lignes de ce que doit être notre outil d'assistance.

Chapitre 2 : De l'organisation opérationnelle proposée dans le chapitre précédent, nous avons dégagé des concepts dont celui de DTLM¹. Ces entités sont des briques fonctionnelles qui concrétisent les liaisons mécaniques du système. Elles forment les éléments de base de

1. Dispositifs Technologiques assurant une Liaison Mécanique

notre processus de synthèse de mécanisme. Nous discutons des caractéristiques générales du DTLM (modèles associés, données colportées, etc.). Nous présentons ensuite la structure de données orientée objet qui abrite ces DTLM, en mettant en valeur l'étude fonctionnelle qui a conduit à son élaboration. Enfin, après avoir introduit la problématique du dimensionnement optimal des DTLM, nous traitons un exemple de mise en œuvre. Nous soulignons pour cela les particularités liées au paramétrage et au choix de la méthode d'optimisation (c'est l'occasion d'aborder le problème des variables discrètes dans les composants mécaniques).

Chapitre 3 : Afin de valider les propos et concepts présentés dans les précédents chapitres, nous avons développé une maquette logicielle d'outil d'assistance à la conception. Dans un premier temps, nous recensons les objectifs que nous fixons pour notre outil. La synthèse d'un mécanisme passe par la réunion de plusieurs DTLM. Nous présentons succinctement les deux grandes phases de synthèse de notre outil de conception assistée : la synthèse découplée et la synthèse coordonnée. Nous abordons ensuite la présentation générale de notre outil par les aspects structurel et interfaçage avec le concepteur. Nous traitons finalement de la phase d'initialisation du processus de synthèse de mécanisme à l'aide de *Sicam 2*. Il s'agit essentiellement du travail de renseignement du cahier des charges global grâce à l'ensemble des informations héritées des niveaux amont de la phase de synthèse.

Chapitre 4 : Ce chapitre est entièrement consacré à l'étude détaillée de la phase de synthèse découplée qui constitue le premier niveau d'assistance que fournit notre outil (nous en rappelons les principes directeurs). Durant cette phase, le concepteur définit, à l'aide de différents menus de définition, les cahiers des charges propres à chaque liaison ; il choisit les types de DTLM qu'il désire associer à ces liaisons, les paramètres de fonctionnement et les objectifs locaux d'optimisation. En préalable aux calculs de dimensionnement des DTLM, les torseurs d'effort aux liaisons doivent être déterminés. Nous expliquons l'importance du choix du type de liaison pivot (décomposable ou non) sur la détermination des torseurs, et ses conséquences sur la chronologie des opérations. Le concepteur peut alors lancer les dimensionnements des DTLM sans tenir compte des contraintes de couplage inhérentes à leur réunion au sein du système. Un exemple de synthèse découplée d'une liaison arbre-moyeu complète permet d'apprécier l'aide apportée au concepteur. Dans l'hypothèse où ce dernier désirerait éliminer les contraintes de couplage entre DTLM, il existe une possibilité de guider le dimensionnement de chacune des liaisons. Ceci est réalisable grâce aux valeurs limites qu'impose le concepteur à certains paramètres des cahiers des charges locaux. Nous illustrons par un exemple cette procédure manuelle de synthèse découplée puis nous discutons de l'intérêt et des limitations d'une telle méthodologie.

Chapitre 5 : La phase de synthèse coordonnée constitue la seconde étape de l'assistance fournie par *Sicam 2*. Elle a pour objectif de mener à terme la synthèse globale du système en tenant compte des couplages entre DTLM. Elle s'appuie pour cela sur les cahiers des charges établis lors des phases d'initialisation et de synthèse découplée. L'étude bibliographique des systèmes décomposés et de leurs méthodes d'optimisation nous a conduit à élaborer et tester une structure collaborative. Nous présentons cette structure collaborative et la méthode d'optimisation associée. L'analyse du comportement de cette méthode et des résultats obtenus nous a amené à envisager un processus spécifique de synthèse coordonnée s'appuyant sur le principe de relaxation sélective des contraintes. Sa mise en œuvre fait appel à un objectif d'optimisation global multicritères afin d'établir une comparaison avec

la méthode collaborative. Nous proposons également cette méthode avec des critères d'optimisation homogènes et en évaluons l'impact sur les résultats. A l'issue de ce chapitre, nous tirons les conclusions de l'emploi de ces diverses méthodes de synthèse coordonnée et estimons leur adéquation avec notre problème.

Chapitre 6 : L'élaboration de notre maquette logicielle s'appuie sur des notions et des outils informatiques. Le choix des uns et des autres est en partie sous-tendu par la structure de notre organisation opérationnelle et la nature des DTLM. Nous détaillons ces différents points sous l'angle de l'architecture logicielle en mettant l'accent sur des éléments tels que les flux d'informations et l'ergonomie de l'interface homme/machine.

Chapitre 7 : Ce chapitre clôt le mémoire en présentant nos conclusions sur les travaux menés. À ce titre nous resituons le travail effectué dans ce contexte en proposant les améliorations et extensions possibles de notre outil d'assistance à la conception et du concept de DTLM.

En espérant que ce mémoire puisse contribuer à l'enrichissement du débat sur la conception assistée par ordinateur, nous vous en souhaitons une bonne lecture...

Table des matières

1	Étude de la démarche de conception.	1
1.1	Introduction.	3
1.2	La phase de conception dans le cycle de vie d'un produit.	3
1.3	Les mécanismes de transmission de puissance à un degré de liberté.	6
1.4	Une phase clé dans la définition du produit : la phase de synthèse.	9
1.5	Représentation du produit dans la phase de synthèse.	10
1.6	Définition d'une organisation opérationnelle de la phase de synthèse.	12
1.6.1	Quelques exemples d'organisation de la phase de synthèse.	13
1.6.2	Organisation opérationnelle basée sur une définition fonctionnelle multi-niveaux.	17
1.6.2.1	Construction du modèle cinématique du mécanisme.	17
1.6.2.2	Détermination de la nature des liaisons.	19
1.6.2.3	Synthèse des contraintes de montage.	20
1.6.2.4	Choix et définition des DTLM.	21
1.6.2.5	Habillage des pièces du mécanisme.	21
1.6.3	Remarques sur le découpage proposé.	22
1.7	Assistance informatique aux concepteurs.	24
1.7.1	Historique de la Conception Assistée par Ordinateur.	24
1.7.2	La conception intégrée.	26
1.7.3	Quelques exemples de systèmes intégrés de conception.	27
1.7.3.1	SICAM.	28
1.7.3.2	Les systèmes architecturés autour de MÉCAEXPERT.	30
1.7.3.3	Les systèmes architecturés autour de SMECI.	31
1.7.3.4	Autres systèmes experts.	32
1.7.3.5	Conclusion sur l'utilisation de l'I.A.	32
1.7.4	Les objectifs du nouveau SICAM.	33
2	Concrétisation des liaisons mécaniques : les DTLM.	35
2.1	Composition d'un DTLM.	37
2.1.1	Exemple de composition d'un DTLM.	37
2.1.2	Importance du paramétrage des DTLM.	38
2.1.3	Le DTLM : feature de haut niveau.	39
2.1.4	Intégration du DTLM dans un environnement CAO.	42
2.1.4.1	Définition des points caractéristiques du DTLM.	43
2.1.4.2	Positionnement du modèle 3D.	46
2.1.4.3	Exploitation du modèle 3D.	47
2.2	Aspect fonctionnel du DTLM.	47
2.2.1	Approche orientée objet.	48

2.2.2	Dimensionnement et optimisation d'un DTLM.	54
2.3	Création des DTLM.	58
2.3.1	Exemple de modélisation d'une famille de DTLM : la liaison complète par clavette.	58
2.3.1.1	Présentation du problème.	58
2.3.1.2	Formulation du problème de dimensionnement optimal de la liaison.	59
2.3.1.3	Résolution du problème par la méthode de Johnson.	70
2.3.1.4	Recherche par énumération.	74
2.3.2	Choix des méthodes d'optimisation.	75
2.3.3	Ensemble des DTLM développés.	77
2.4	Conclusions.	79
3	Organisation générale de l'outil d'assistance.	81
3.1	Objectifs de l'outil d'assistance.	82
3.1.1	Synthèse découplée.	82
3.1.2	Synthèse coordonnée.	84
3.2	Présentation générale de l'outil d'assistance.	84
3.2.1	Architecture opérationnelle de l'interface homme/machine.	84
3.2.2	Les interfaces graphiques : menus et I-DEAS [®]	85
3.2.2.1	Dialogue par les menus.	86
3.2.2.2	Dialogue par I-DEAS [®]	86
3.2.3	Menu principal.	86
3.3	Initialisation de la phase de conception : le cahier des charges global.	90
4	Synthèse découplée des DTLM.	93
4.1	Principes de la synthèse découplée.	95
4.2	Préalable à la synthèse de DTLM : détermination des torseurs d'efforts aux liaisons.	95
4.2.1	Liaison pivot décomposable.	96
4.2.2	Liaison pivot non décomposable.	97
4.3	Synthèse des DTLM de la liaison pivot.	98
4.3.1	Cas de deux DTLM indépendants.	98
4.3.2	Cas de deux DTLM dépendants.	100
4.4	Synthèse des DTLM des liaisons complètes arbre/moyeu.	102
4.4.1	Menu de définition commun aux diverses technologies de liaison.	102
4.4.2	Menu de définition caractéristique à la technologie de liaison.	104
4.4.2.1	Liaison complète par clavette.	104
4.4.2.2	Liaison complète par cannelures.	106
4.4.2.3	Liaison complète par frettage.	107
4.4.3	Remarques.	108
4.4.4	Exemple de synthèse découplée d'une liaison complète arbre/moyeu.	109
4.4.4.1	Présentation du problème.	110
4.4.4.2	Étude de la liaison complète par clavette.	110
4.4.4.3	Étude de la liaison complète par cannelures.	110
4.4.4.4	Étude de la liaison complète par frettage.	112
4.4.4.5	Observations.	112
4.5	Procédure de synthèse découplée de l'ensemble.	113
4.5.1	Gestion manuelle des contraintes.	114
4.5.1.1	Les interférences axiales.	114

4.5.1.2	Les interférences de montage.	116
4.5.1.3	Pertinence de la gestion manuelle des contraintes.	119
4.5.2	Aide au choix des DTLM.	119
4.5.3	Exemple de synthèse découplée.	122
4.6	Conclusions.	124
5	Synthèse coordonnée des DTLM.	127
5.1	Identification du problème de synthèse coordonnée.	129
5.1.1	Objectif.	129
5.1.2	Nature du problème.	129
5.1.3	Paramètres initiaux de la synthèse coordonnée.	130
5.2	Les systèmes décomposés et leur optimisation.	131
5.2.1	Les différents types de décomposition.	131
5.2.1.1	Décompositions hiérarchiques.	131
5.2.1.2	Décompositions non-hiérarchiques.	132
5.2.2	L'optimisation pluridisciplinaire (MDO).	133
5.2.3	Optimisation collaborative.	134
5.3	Application de l'optimisation collaborative à notre problème.	137
5.3.1	Formulation initiale du problème sous forme non-hiérarchique.	137
5.3.2	Problème global exprimé sous forme collaborative.	138
5.3.3	Choix d'une méthode pour l'implémentation du niveau système : l'algorithme de Powell.	141
5.3.4	Expression du problème du niveau système en problème sans contrainte.	142
5.3.5	Mise en œuvre de l'algorithme de Powell.	142
5.3.6	Influence du comportement des modules de calcul.	145
5.3.7	Conclusions.	149
5.4	Processus spécifique de synthèse coordonnée.	149
5.4.1	Présentation de la méthode.	149
5.4.2	Description de l'algorithme de relaxation sélective des contraintes.	151
5.4.2.1	Point de départ du processus.	151
5.4.2.2	Recherche d'une solution réalisable.	151
5.4.2.3	Optimisation d'une solution réalisable.	156
5.4.2.4	Remarques	158
5.4.3	Relaxation sélective avec un objectif multicritère.	159
5.4.4	Relaxation sélective avec des objectifs homogènes.	160
5.4.4.1	Caractérisation des problèmes.	160
5.4.4.2	Cas des problèmes semi-contraints.	162
5.4.4.3	Cas des problèmes sous-contraints.	163
5.4.4.4	Remarques.	166
5.5	Conclusions.	168
6	Implémentation informatique.	171
6.1	Introduction.	172
6.2	Architecture logicielle.	172
6.2.1	Choix des langages de programmation.	173
6.2.2	Liaison des modules.	174
6.2.3	L'interface homme/machine.	177
6.2.3.1	Les menus de renseignement du système.	178

6.2.3.2	Visualisation graphique 3D des DTLM.	178
6.3	Interface graphique et ergonomie du système.	184
6.4	Commentaires sur l'interface proposée.	185
7	Conclusions et perspectives.	187
7.1	Conclusions	187
7.2	Perspectives	188
A	Cahiers des charges du problème-support.	199

Table des figures

1.1	Processus de conception de produit.	4
1.2	Définition du produit	6
1.3	Exemple de mécanisme générateur de fonction.	8
1.4	Quelques réalisations industrielles de réducteurs de vitesse.	8
1.5	Caractéristiques des niveaux de définition.	9
1.6	Niveaux de décomposition fonctionnelle selon [Gui87].	11
1.7	Décomposition d'un mécanisme en solutions technologiques selon [Gui87].	12
1.8	Méthode de fractionnement selon [VDI93].	14
1.9	Conception distribuée selon [Deg96].	15
1.10	Formalisation d'un problème de conception distribué selon [Deg96].	16
1.11	Représentation partielle d'un mécanisme de grignoteuse.	17
1.12	Découpage opérationnel : la tâche T1.	18
1.13	Exemples de M.M.E. d'après [Fau99].	18
1.14	Découpage opérationnel : la tâche T2.	19
1.15	Découpage opérationnel : la tâche T3.	20
1.16	Découpage opérationnel : la tâche T4.	21
1.17	Découpage opérationnel : la tâche T5.	22
1.18	Découpage opérationnel de la phase de synthèse.	23
1.19	Copie d'écran de SICAM [Gui87].	29
1.20	Recherche d'une solution optimale dans SICAM.	30
1.21	Exemple de composant fonctionnel complexe selon [Dam96].	32
2.1	Assemblage d'un DTLM.	38
2.2	Différences de paramétrage.	39
2.3	Exemples de DTLM réalisant différents types de liaisons.	40
2.4	Interface graphique menus/CAO de <i>Sicam</i> 2.	43
2.5	Placement des DTLM selon le cahier des charges principal.	44
2.6	Point de référence et trièdre associé d'un modèle géométrique 3D.	44
2.7	Représentation bidimensionnelle d'un problème tridimensionnel.	45
2.8	Point de référence et centre d'effort.	45
2.9	Positionnements possibles d'un DTLM par rapport au centre géométrique de la liaison.	47
2.10	Formalisme objet selon [Boo94].	49
2.11	Décomposition d'une liaison pivot.	51
2.12	Architecture interne d'une liaison pivot.	51
2.13	Rotule unilatérale fournissant ses services.	52
2.14	Exemple d'objets non divisibles.	53
2.15	Exemples de DTLM disponibles.	53
2.16	Contraintes d'encombrement sur un DTLM.	55

2.17	Montage de deux DTLM.	56
2.18	Contraintes de montage d'un DTLM.	56
2.19	Différentes géométries de clavettes.	58
2.20	Combinaisons d'arrêts axiaux.	59
2.21	DTLM liaison complète par clavette, entretoise et épaulement.	60
2.22	Exemple de paramétrage de l'arbre du DTLM.	60
2.23	Détail du paramétrage de la clavette.	60
2.24	Paramétrage de l'entretoise et du moyeu.	61
2.25	Dimensions normalisées des assemblages clavetés (extraits de la norme NF E 22-175 et NF E 22-177).	61
2.26	Définition des relations dimensionnelles.	62
2.27	Exemples de paramétrages sur l'arbre de la liaison clavetée.	64
2.28	Quotient de sollicitation au matage des flancs de clavette.	67
2.29	Discrétisation du domaine des solutions.	70
2.30	Représentation graphique du domaine des solutions.	71
2.31	Couples solutions pour une minimisation du diamètre.	72
2.32	Couples solutions pour une minimisation de la longueur de clavetage.	73
2.33	Cas d'une fonction contrainte non monotone.	73
2.34	Structure objet des DTLM.	77
2.35	Étagement des dimensions de roulement à billes radiales.	79
3.1	Exemple de composition d'un super-assemblage.	83
3.2	Chronologie opérationnelle de <i>Sicam</i> 2.	85
3.3	Actigramme du système.	85
3.4	Arborescence des fenêtres.	87
3.5	Interface graphique du logiciel I-DEAS®.	88
3.6	Menu principal.	88
3.7	Fenêtre de saisie du nom de projet.	89
3.8	Cahier des charges global.	90
4.1	Exemple d'un cas de statique associé à une synthèse de DTLM.	96
4.2	Menu de définition des DTLM d'une liaison pivot : cas des DTLM indépendants.	98
4.3	Fenêtres de définition complémentaires pour DTLM indépendants.	99
4.4	Modèles 3D de DTLM associés à une liaison pivot.	100
4.5	Menu de définition des DTLM d'une liaison pivot : cas des DTLM dépendants.	100
4.6	Fenêtre de définition complémentaire pour DTLM dépendants.	101
4.7	Modèles 3D de DTLM à base de roulements à rouleaux coniques.	102
4.8	Cahier des charges pour la liaison complète arbre/moyeu.	103
4.9	Liste des options du menu-bouton « choix de matériau ».	103
4.10	Menu de définition caractéristique de la liaison complète arbre/moyeu par clavette.	104
4.11	Fenêtre de visualisation d'un fichier « résultats de calcul ».	105
4.12	Modèle 3D d'un DTLM à base de clavette.	105
4.13	Menu de définition caractéristique de la liaison arbre/moyeu par cannelures.	106
4.14	Modèle 3D d'un DTLM à base de cannelures.	107
4.15	Menu de définition caractéristique de la liaison arbre/moyeu par frettage.	108
4.16	Modèle 3D d'un DTLM de liaison arbre/moyeu par frettage.	109
4.17	Configuration de l'exemple de synthèse découplée d'une liaison arbre/moyeu.	110
4.18	Menu de définition commun de la liaison complète arbre/moyeu.	111

4.19	Menu de définition caractéristique du DTLM à base de clavette et modèle 3D résultant.	111
4.20	Menu de définition caractéristique du DTLM à base de cannelures et modèle 3D résultant.	112
4.21	Menu de définition caractéristique du DTLM à base de frettage et modèle 3D résultant.	113
4.22	Interférences axiales entre DTLM.	114
4.23	Fenêtre de topologie du mécanisme.	115
4.24	Exemple de définition «à risque» des espaces de DTLM.	115
4.25	Exemple de résolution manuelle de conflits axiaux.	117
4.26	Géométries d'arbres liées aux sens de montage des DTLM.	117
4.27	Diagrammes des moments.	120
4.28	Fenêtre de visualisation de l'encombrement de tous les DTLM solutions.	121
4.29	Fenêtre de visualisation des diamètres d'arbre de tous les DTLM solutions.	121
4.30	Résultat (tronçons d'arbre) d'une première passe de calculs.	122
4.31	Après restriction des encombrements axiaux.	123
4.32	Après modification de la valeur limite sur le diamètre d'arbre.	123
4.33	Modélisation des flux d'informations de la phase de synthèse découplée.	125
5.1	Fenêtre du menu de synthèse coordonnée.	130
5.2	Décomposition hiérarchique d'un problème d'optimisation.	131
5.3	Décomposition non-hiérarchique d'un problème d'optimisation.	132
5.4	Exemple de mise en place de paramètres de liaison.	133
5.5	Problème multidisciplinaire couplé	135
5.6	Formulation en optimisation collaborative	136
5.7	Découpage du problème d'optimisation collaborative.	137
5.8	Découpage hiérarchique du problème d'optimisation collaborative.	140
5.9	Exemple d'application de l'algorithme de Powell sur un problème bidimensionnel.	142
5.10	Diagramme d'évolution des consignes du niveau système.	144
5.11	Dimensions de solutions concurrentes d'un DTLM pour une minimisation de la largeur.	146
5.12	Réponses d'un module local aux consignes du niveau système.	147
5.13	Influence du type de DTLM sur la réponse d'un module local.	147
5.14	Dépassement de consignes pour le DTLM 3.	148
5.15	Paramètres d'entrée et de sortie d'un module de dimensionnement de DTLM.	149
5.16	Découpage hiérarchique du problème pour la méthode de relaxation sélective.	150
5.17	Conflit axial à l'issue de la première étape malgré une optimisation de l'encombrement axial.	152
5.18	Algorithme de création d'une première proposition.	152
5.19	Espaces axiaux inter-DTLM.	153
5.20	Évolution du conflit radial.	154
5.21	Tableaux variationnels des DTLM.	155
5.22	Influence de la vitesse d'évolution d'un DTLM sur la résolution des conflits radiaux.	156
5.23	Algorithme de construction d'une solution réalisable.	157
5.24	Principe de la mise à jour des contraintes.	157
5.25	Algorithme d'optimisation.	158
5.26	Influence du positionnement des limites axiales en problème semi-contraint.	161
5.27	Cas typique de limites axiales d'un problème semi-contraint.	162

5.28	Première proposition du problème semi-contraint.	163
5.29	Solution finale du problème semi-contraint.	163
5.30	Modèle 3D complet de la solution optimisée.	164
5.31	Cas typique de limites axiales d'un problème sous-contraint.	165
5.32	Ensemble des DTLM solutions d'un problème sous-contraint.	165
5.33	Première proposition et solution après optimisation.	166
5.34	Représentation complète de la solution proposée au problème sous-contraint. . .	167
5.35	Visualisation d'une étape du processus de synthèse coordonnée par algorithme. .	167
5.36	Diagramme de Pareto représentant la relation entre l'investissement en temps et la qualité de la solution obtenue.	169
6.1	Datagramme du système.	175
6.2	Actigramme du système.	176
6.3	Architecture simplifiée de CORBA.	179
6.4	Interface graphique du logiciel I-DEAS®.	180
6.5	Composition d'un assemblage racine.	180
6.6	Algorithme primaire du module de visualisation.	181
6.7	Exemple de codage de DTLM de liaison rotule.	182
6.8	Répartition moyenne des temps d'actions du module de visualisation 3D.	182
6.9	Paramétrage simplifié du modèle 3D de l'arbre claveté épaulé.	183
6.10	Transfert maximal des paramètres dimensionnels.	183
6.11	Transfert minimal des paramètres dimensionnels.	183
A.1	Fenêtre du cahier des charges global.	199
A.2	Fenêtre de définition de la liaison pivot.	200
A.3	Fenêtre de définition complémentaire de la liaison A.	200
A.4	Fenêtre de définition complémentaire de la liaison B.	200
A.5	Fenêtre du menu de définition commun de la liaison arbre-moyeu 1.	201
A.6	Fenêtre du menu de définition caractéristique de la liaison arbre-moyeu 1.	201
A.7	Fenêtre du menu de définition commun de la liaison arbre-moyeu 2.	202
A.8	Fenêtre du menu de définition caractéristique de la liaison arbre-moyeu 2.	202

Chapitre 1

Étude de la démarche de conception.

Résumé

LA conception est une étape importante de la vie d'un produit. Nous précisons tout d'abord la place qu'elle occupe dans le cycle de vie. Cette étape met en jeu une phase de synthèse et une phase d'analyse que nous détaillons. La phase de synthèse peut encore faire l'objet d'un découpage selon des critères fonctionnels que nous précisons. L'un des stades de la phase de synthèse, domaine précis de notre étude, fait appel à des outils regroupant méthodologies et modèles de représentation. Nous présentons quelques-uns de ces outils avant de dévoiler l'organisation opérationnelle de la phase de synthèse que nous avons retenue pour la suite de nos travaux. Dès lors, nous pouvons aborder la problématique de l'assistance à la conception durant cette phase de synthèse. Nous présentons quelques exemples de systèmes informatiques déjà proposés dans ce domaine puis nous définissons ce que devra être notre système d'aide à la conception.

Contenu du chapitre

1.1	Introduction.	3
1.2	La phase de conception dans le cycle de vie d'un produit.	3
1.3	Les mécanismes de transmission de puissance à un degré de liberté.	6
1.4	Une phase clé dans la définition du produit : la phase de synthèse.	9
1.5	Représentation du produit dans la phase de synthèse.	10
1.6	Définition d'une organisation opérationnelle de la phase de synthèse.	12
1.6.1	Quelques exemples d'organisation de la phase de synthèse.	13
1.6.2	Organisation opérationnelle basée sur une définition fonctionnelle multi-niveaux.	17
1.6.2.1	Construction du modèle cinématique du mécanisme.	17
1.6.2.2	Détermination de la nature des liaisons.	19
1.6.2.3	Synthèse des contraintes de montage.	20

1.6.2.4	Choix et définition des DTLM.	21
1.6.2.5	Habillage des pièces du mécanisme.	21
1.6.3	Remarques sur le découpage proposé.	22
1.7	Assistance informatique aux concepteurs.	24
1.7.1	Historique de la Conception Assistée par Ordinateur.	24
1.7.2	La conception intégrée.	26
1.7.3	Quelques exemples de systèmes intégrés de conception.	27
1.7.3.1	SICAM.	28
1.7.3.2	Les systèmes architecturés autour de MÉCAEXPERT.	30
1.7.3.3	Les systèmes architecturés autour de SMECI.	31
1.7.3.4	Autres systèmes experts.	32
1.7.3.5	Conclusion sur l'utilisation de l'I.A.	32
1.7.4	Les objectifs du nouveau SICAM.	33

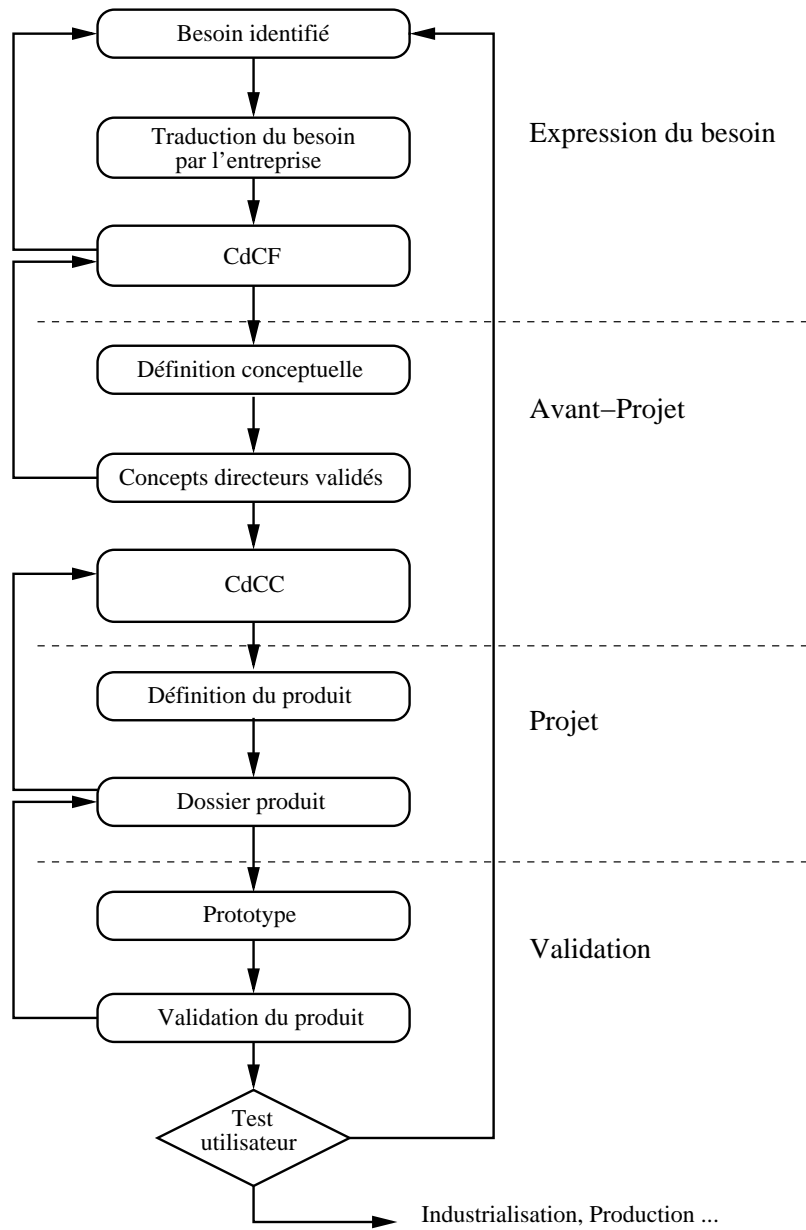
1.1 Introduction.

La conception mécanique est actuellement l'objet de nombreux travaux de recherche. Elle doit cet engouement international à la place qu'elle occupe au sein de l'ensemble des activités du domaine de la mécanique. C'est tout d'abord l'activité la plus stratégique puisqu'elle se situe au début du cycle de vie du produit et, de ce fait, conditionne grandement l'avenir du produit. Elle est aussi l'activité la moins facilement modélisable, car elle repose pour un certain temps encore, sur les compétences humaines. Enfin c'est la discipline qui est longtemps restée la plus occulte et pour laquelle la plupart des mécanismes cognitifs restent à découvrir. On comprendra dès lors pourquoi tant d'équipes de recherche se penchent sur la conception. Les axes de recherche sont nombreux. Ils vont de la modélisation du processus de conception à la représentation des données manipulées en passant par l'étude des outils et la caractérisation des intervenants (humains ou institutionnels). Il est délicat de dégager une vision générale de la conception, tant les approches sont variées et les méthodes différentes. On peut néanmoins remarquer que la mécanique, à l'instar des autres secteurs industriels, vie au rythme des marchés, devenus mondiaux, et que la recherche tend à développer de nouvelles méthodes de travail adaptées à cette globalisation. On retrouve donc des objectifs communs à de nombreuses industries comme le productivisme et la rentabilité, ainsi que des méthodes communes comme la coopération.

1.2 La phase de conception dans le cycle de vie d'un produit.

Le produit mécanique est avant tout un objet industriel. Parmi tous les points de vue relatifs à l'étude et l'analyse de l'objet industriel [Def85], nous retiendrons celui du concepteur. Comme tout être « vivant », un produit mécanique naît, vie et meurt selon la métaphore biologique. On distingue dans ce cycle de vie des étapes auxquelles on associe des acteurs et des filières de métiers. Le recensement de ces étapes donne le classement chronologique suivant :

- L'expression du besoin. Actuellement, le cycle de vie du produit commence bien plus tôt que les premiers coups de crayons. L'idée du besoin pourrait être considérée comme l'acte de naissance du produit (Celui-ci ne constituant que la réponse matérielle à l'expression de ce besoin). Une exception à cela : l'opération de marketing où la création (ou la révélation du besoin au futur client) n'est qu'un alibi au développement d'un nouveau produit ne découlant pas d'un besoin exprimé explicitement [Tro99]. Le besoin (donc la demande) naît alors de l'offre. Une fois ce besoin exprimé et identifié, il faut le concrétiser en un document appelé le *cahier des charges fonctionnel* (CdCF). Ce document, objet de la norme NF X 50-151 [AFN91], traduit le besoin en situant le produit dans son marché et son contexte industriel, en énonçant les fonctions attendues par le client et en fixant le cadre de réponse du fournisseur. Il n'est pas figé, et peut subir des modifications commandées par l'évolution du besoin, ce qui oblige à une grande clarté dans sa rédaction. Vient ensuite la traduction matérielle des fonctions de services ou techniques exprimées dans le CdCF.
- L'avant-projet. Dès lors que le marché existe pour un produit, et que sa commercialisation s'avère réalisable et fructueuse (sur un plan strictement financier), les études sont lancées. Cette phase vise à créer les solutions conceptuelles du produit à partir de l'expression des besoins et des connaissances heuristiques des concepteurs. C'est une interprétation du CdCF qui est fortement imprégnée de l'identité de l'entreprise. C'est la phase la plus

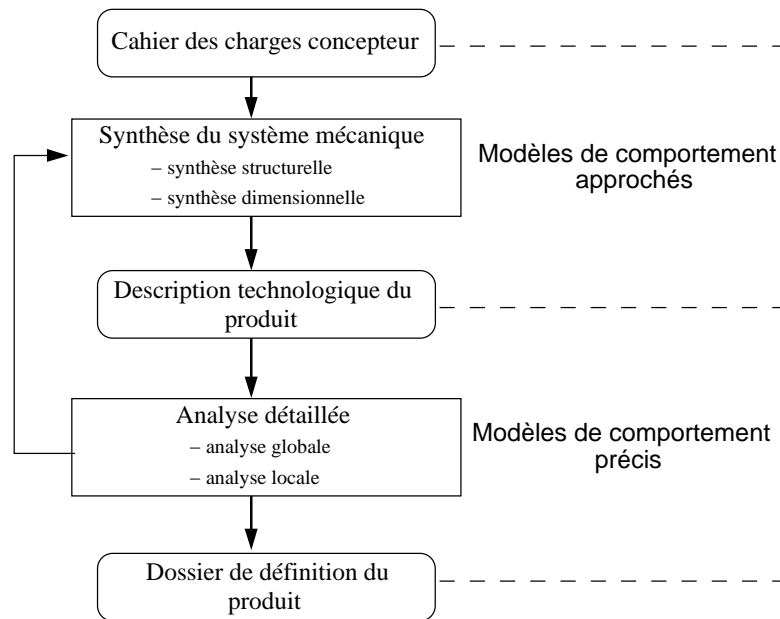
FIG. 1.1 – *Processus de conception de produit.*

difficile à modéliser car elle fait encore appel, en majeure partie, à l'expérience et au jugement humain. Elle conduit à l'élaboration du *cahier des charges concepteur* (CdCC).

- Le projet (ou développement). Voici l'étape où les concepts exprimés précédemment vont être progressivement affinés jusqu'à définir complètement les éléments constitutifs du produit. Nous sommes dans le domaine de prédilection des calculs, à l'interface de l'abstrait et du tangible. Les pièces trouvent leur formes définitives, validées par des calculs de résistance, et les dimensions sont fixées.
- La validation, l'industrialisation et la production. Ces termes que nous serions tentés de regrouper, distinguent trois tâches intimement liées : après une ultime validation par le client, le produit est l'objet d'une étude d'industrialisation qui consiste à choisir et mettre en œuvre les méthodes de fabrication tandis que la phase de production caractérise la mise en œuvre des moyens déterminés dans l'étude d'industrialisation.
- La commercialisation et le suivi (éventuellement l'évolution du produit). Le produit rejoint son lieu d'exploitation. L'acquéreur est en droit d'attendre du fournisseur un service, que l'on qualifie d'*après-vente*, auquel on associe à tort la fonction de dépannage. Ce service peut prendre la forme d'une assistance (téléphonique : les fameuses «Hot Lines»), d'un suivi d'utilisation (programme de révision ou d'entretien), voire d'une mise à jour du produit. Il arrive aussi que le produit fasse l'objet de modifications commandées par une défaillance (campagne de rappel des constructeurs automobiles pour un défaut de conformité par exemple), une expression nouvelle du besoin, ou bien une inadéquation due à l'évolution du besoin (extension de la gamme, ...). On repasse alors par des phases dites de reconception qui s'apparentent aux phases déjà vues précédemment.
- Enfin, comme cela devient le plus souvent le cas, la destruction et/ou le recyclage. Cette prise en compte récente des problèmes d'environnement et de revalorisation des déchets n'est pas encore totalement assimilée par les industriels. La démarche de conception en vue du recyclage tend cependant à se généraliser, en partie grâce (ou à cause) des règles mises en place par les pouvoirs publics. Il est à noter que le recyclage constitue un enjeu économique non négligeable pour les industries gourmandes en matières premières ou dépendantes de matériaux nobles. Ceci explique l'intérêt nouveau de ces industries dans les processus de collecte et de revalorisation des déchets.

La partie qui nous intéresse est la partie projet consacrée à la *définition du produit*. Elle peut être décomposée en deux actions successives et complémentaires qui sont la *synthèse* et l'*analyse*. Ces deux actions, visualisées sur la figure 1.2 sont intimement liées et sont souvent l'objet de réactivations mutuelles.

Phase de synthèse : Elle a pour but de produire une représentation quasi-complète du produit, en regard des fonctions exprimées dans le cahier des charges concepteur. Le fruit de cette phase est un dossier d'avant-projet qui met en place les principes technologiques (fonctions technologiques et opérateurs technologiques [CdP73]) retenus pour satisfaire les fonctions de service du produit. Cette étape est particulièrement difficile à appréhender pour les cognitiens de la conception. Les mécanismes de réflexion mis en jeu sont nombreux et mal expliqués (schèmes d'assemblage d'objets impliquant des mécanismes de reconnaissance de fonction, de substitution, etc). Une explication de cet ésothérisme se situe vraisemblablement dans l'acte de passage d'une représentation abstraite (la fonction) à une représentation concrète (l'objet).

FIG. 1.2 – *Définition du produit*

Phase d’analyse : Les choix issus de la phase de synthèse doivent être individuellement puis globalement validés. Pour cela, des modèles comportementaux sont mis en œuvre afin de simuler le fonctionnement de tout ou partie du système. C’est le monde des outils de simulation tels que la simulation de mécanismes, les éléments finis, etc.

Pourquoi scinder la définition du produit en deux phases? Cela tient au fait que les domaines concernés par ces deux phases présentent des singularités qui tendent naturellement à les distinguer. Des éléments de comparaison sont donnés dans le tableau 1.1. Ces différences sont en premier lieu de nature opératoire. La phase de synthèse est l’étape où se construit la solution par manipulations d’éléments primaires. Les caractéristiques concernant ces éléments sont connues des spécialistes, mais la définition en est souvent imparfaite. C’est toute la force de l’expérience que de pouvoir mener en peu d’essais une synthèse performante. A l’inverse, la phase d’analyse s’attache à décomposer le produit afin d’évaluer la pertinence des divers composants. On procède pour cela à une distribution de l’information au niveau local pour valider la totalité du système.

1.3 Les mécanismes de transmission de puissance à un degré de liberté.

Une des thématiques principales de recherche de notre groupe est l’assistance à la synthèse des mécanismes. Ce domaine étant très vaste, nous avons choisi de nous limiter à l’étude d’un type de mécanisme particulier, les *mécanismes générateurs de fonction* [SM89]. Ce type de mécanismes se caractérise par une fonction de transformation d’une consigne d’entrée en une consigne de sortie. Les grandeurs concernées peuvent être des vitesses, des puissances... La figure 1.3 représente un mécanisme à barres de la famille des générateurs de fonction. Au sein de cette famille, nous nous sommes concentrés sur les mécanismes de transmission de puissance à un degré de liberté parmi lesquels on trouve, par exemple, les mécanismes réducteurs. Ces mécanismes forment une sous-classe particulière des mécanismes générateurs de fonction puisque la fonction en question est une simple réduction de la consigne d’entrée, c’est-à-dire la vitesse de

	Synthèse	Analyse
Entités manipulées	Les entités manipulées sont d'abord abstraites. Elles deviennent progressivement plus proche de la réalité en adoptant des formes et des structures proches du concret.	Les entités manipulées sont des pièces, des volumes, de la matière.
Calculs	Les calculs sont relativement simples. Ils font appel à des modèles agrégés, simplifiés qui représentent sommairement les comportements attendus. La principale préoccupation des concepteurs, à ce niveau, est d'obtenir des ordres de grandeur afin d'opérer des comparaisons entre plusieurs options candidates.	Les modèles employés ici sont extrêmement précis. Ils font appel à une puissance de calcul que seul l'informatique peut fournir. L'objectif de cette phase est d'obtenir des résultats similaires au comportement réel du produit fini. C'est aussi un moyen de diminuer le coût des essais sur prototypes.
Niveau d'expertise	La phase de synthèse requiert une expérience certaine que souvent seuls les «anciens» possèdent. La qualité du produit obtenue est fréquemment proportionnelle à la somme des expériences accumulées par l'équipe de concepteurs.	Plus que des experts, ce sont des spécialistes. Ils connaissent parfaitement les outils employés et maîtrisent les techniques opératoires. Leurs compétences sont souvent liées à la rigueur de leur démarche.
Outils d'assistance	Il est très difficile de cerner un profil type d'outils d'assistance pour la phase de synthèse. On peut néanmoins dégager quelques traits communs. L'interface utilise une sémantique proche du langage humain, que ce soit pour les langages de programmation ou pour les données manipulées. On met en place des bases de règles (systèmes experts) ou/et des bibliothèques de solutions existantes. La clé de ces outils est une représentation symbolique qui synthétise les connaissances sous-tendues par l'utilisation des entités manipulées.	Cette phase est devenue ces dernières années le domaine de prédilection des ordinateurs. Ils remplacent à moindre frais les campagnes d'essais qui validaient jadis les choix effectués en amont. On trouve donc de nombreux logiciels utilisant des codes numériques exploitant au mieux la puissance de calcul des machines.

TAB. 1.1 – *Éléments de comparaison entre les phases de synthèse et d'analyse*

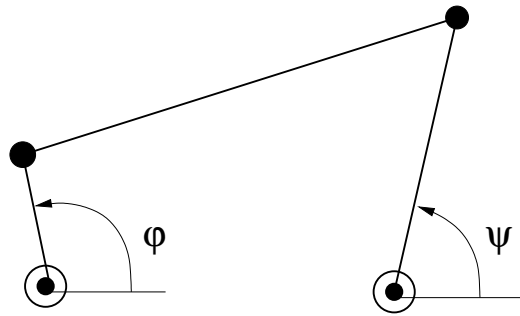


FIG. 1.3 – Exemple de mécanisme générateur de fonction.

rotation, dans un rapport que nous choisissons constant. Malgré leur apparente simplicité, ces mécanismes font l'objet d'une multitude de solutions, tant conceptuelles que technologiques, qu'il nous est impossible d'énumérer complètement. La variété de ces mécanismes est illustrée

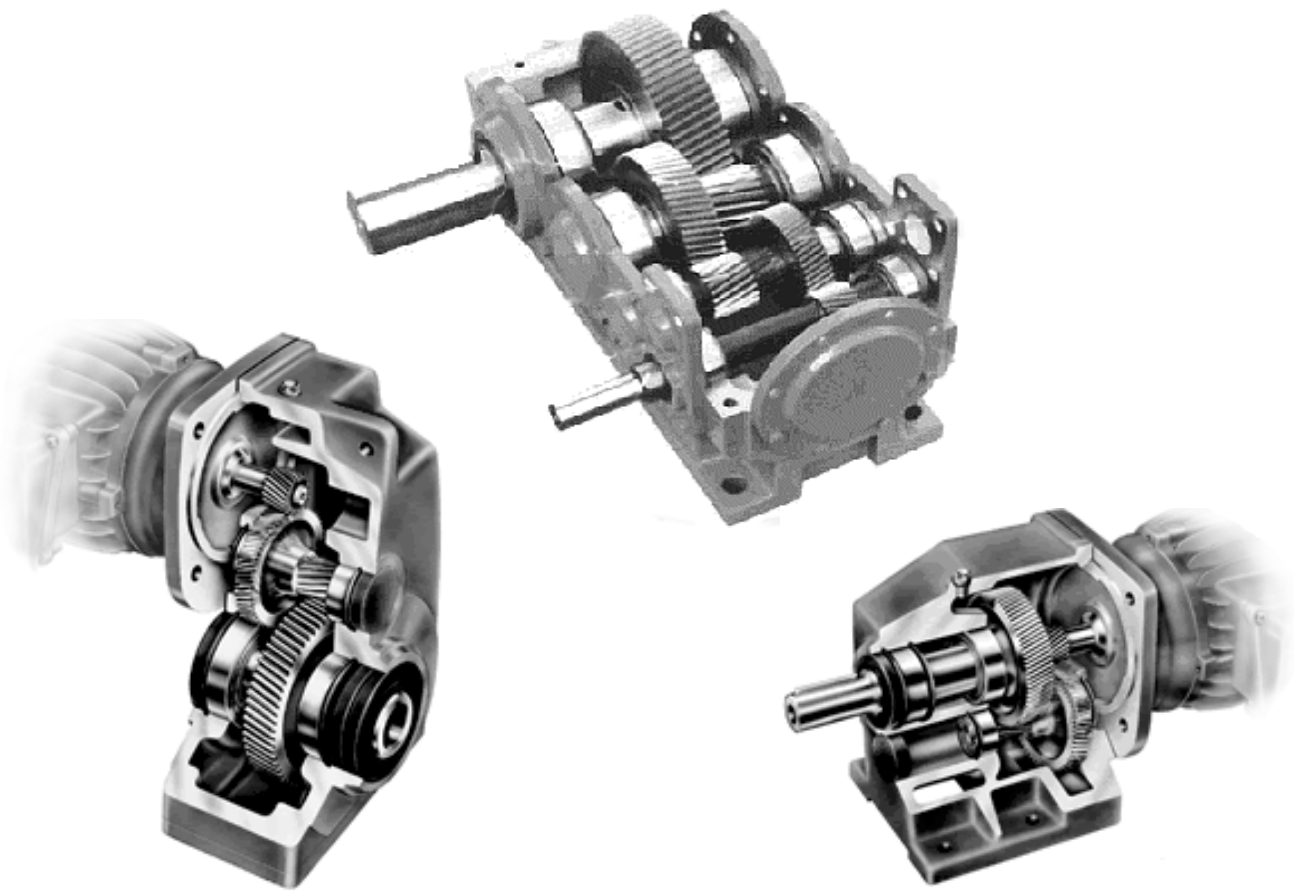


FIG. 1.4 – Quelques réalisations industrielles de réducteurs de vitesse.

sur la figure 1.4 avec quelques réalisations industrielles. On relève cependant des traits communs à tous ces mécanismes, ce qui en facilite le recensement et l'étude la plus exhaustive possible.

Les objectifs principaux de notre travail sont de définir une organisation, des méthodes et des outils d'assistance dédiés à la synthèse de ces mécanismes. Parmi les travaux précurseurs de notre équipe, le système SICAM constitue notre base de réflexion sur les systèmes d'assistance au concepteur. Il découle d'une analyse précise du processus de conception menée par Jean Guillot

et son équipe dès le début des années 80 [GMR81]. Ce système sera détaillé dans le paragraphe 1.7.3.1.

1.4 Une phase clé dans la définition du produit : la phase de synthèse.

La phase de synthèse possède un domaine d'action très étendu. Elle mène le concepteur de la définition la plus macroscopique du système (niveau d'abstraction le plus élevé) à la définition la plus détaillée (niveau d'abstraction le plus bas). Nous pouvons établir une hiérarchie dans la progression de la phase de synthèse qui nous mènera à un découpage en étapes de l'activité de synthèse (cf. figure 1.5). Nous présenterons, au paragraphe 1.6, l'organisation que nous proposons pour cette phase de synthèse (les étapes correspondant alors aux niveaux fonctionnels considérés). Pour chaque étape, nous pouvons encore dégager deux stades de développement qui correspondent à une approche pragmatique de la synthèse. Ces deux temps de la synthèse sont la *synthèse structurelle* et la *synthèse dimensionnelle*. Elles sont consécutives, et reflètent le raison-

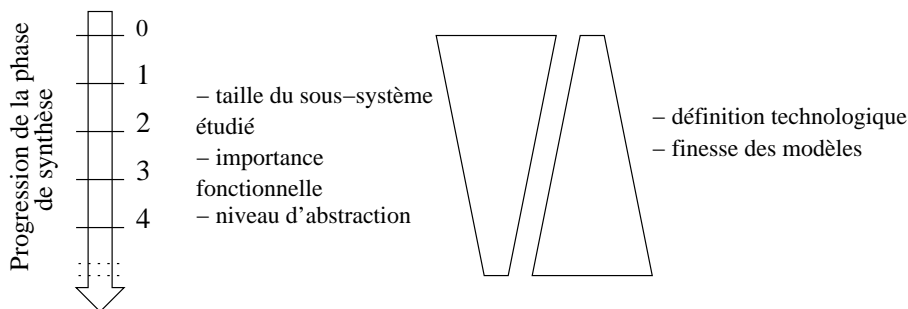


FIG. 1.5 – *Caractéristiques des niveaux de définition.*

nement naturel de l'humain qui d'abord structure le système, puis lui attribue des dimensions cohérentes. Les définitions suivantes prennent pour exemple la toute première étape d'une telle phase de synthèse, qui vise à produire une architecture de système.

synthèse structurelle : Au premier stade de la synthèse, le concepteur manipule des blocs élémentaires (ici des mécanismes élémentaires connus) qu'il combine pour obtenir la ou les fonctions de service requises. L'aspect du système est alors purement schématique, voire technologique, et l'assemblage résultant n'est qu'un assemblage de fonctions. On retrouve souvent une démarche de mise en concurrence de solutions technologiques, pour lesquelles un choix doit être fait. Compte-tenu du peu d'informations dont dispose le concepteur à ce stade du projet, le choix, ou plutôt l'éventuel classement des solutions, s'effectue suivant des critères représentatifs des solutions technologiques manipulées (cinématiques, géométriques...). On ne peut encore évaluer des critères objectifs tels que le coût, la complexité de réalisation. Plusieurs méthodes ont été proposées afin d'apporter de l'aide au concepteur. Dans notre exemple, la plus simple consiste en un archivage de mécanismes simples sous forme de catalogue dans lequel le concepteur puise [Art76]. Cette technique de conception, connue sous le nom de « catalog design » est souvent employée dans les bureaux d'études. Viennent ensuite les méthodes fournissant une assistance à la combinaison de blocs élémentaires. Les techniques employées sont diverses, on citera notamment les méthodes à base de graphes ou plus récemment à base de règles, d'agents multiples ou de

stratégies évolutionnistes [CW96]. Une méthode originale proposée dans [HWW98] propose d'aider le concepteur dans son choix de composants (de toutes natures). Partant du postulat que tout composant est identifiable de façon unique par les propriétés intrinsèques qui définissent son profil caractéristique, les auteurs proposent une méthode graphique qui autorise la comparaison du profil fonctionnel recherché (données issues du CdCF et traduites en contraintes) avec les profils caractéristiques de chaque composant candidat. A l'issue de cette étape de synthèse structurelle, le concepteur dispose d'un schéma de principe compatible avec les fonctions attendues, qu'il doit dimensionner.

synthèse dimensionnelle : On procède, dans cette étape, au dimensionnement des éléments de mécanisme qui ont été retenus lors de la *synthèse structurelle*. Il s'agit, non pas de dimensionner des pièces, mais plutôt de positionner et de définir les dimensions des composants choisis durant la synthèse structurelle. On valide ainsi, non plus la nature des fonctions générées, mais la qualité de ces fonctions. On intègre à ce stade des outils mathématiques de résolution de problèmes contraints tels que les méthodes de gradient, les méthodes stochastiques (algorithmes génétiques, tabou, recuit simulé...).

1.5 Représentation du produit dans la phase de synthèse.

Ce chapitre aborde l'aspect représentation du produit. Précisons immédiatement que la représentation du produit fait appel à deux notions souvent confondues [Loy91] : la modélisation et la visualisation. Tandis que la modélisation a pour but la définition du produit, la visualisation n'a qu'un rôle de représentation restrictive interprétable selon un contexte précis. Nous traitons par la suite de la modélisation du produit.

Le fil conducteur de l'étape de synthèse est bien entendu le produit (le mécanisme). Du début jusqu'à la fin, il va prendre forme d'après les choix réalisés par l'équipe de concepteurs. La représentation finale du produit, à l'issue de la phase de conception, est une représentation graphique, géométrique et dimensionnée, universellement reconnue et appelée dessin de définition. Cette représentation, bien que très réaliste, n'en est pas moins une *modélisation abstraite* du produit, comme toutes les autres représentations. Durant sa conception, le produit est passé d'une description fonctionnelle textuelle, le CdCF, à une description géométrique qui représente au mieux son futur aspect physique. Une multitude d'étapes intermédiaires ont donné lieu à différentes représentations du produit. Ces représentations découlent de la méthodologie de conception retenue. Par exemple, une approche fonctionnelle [Boc98] recourt à des représentations du type *diagramme* comme dans le formalisme SADT¹ ou *tabulaire* comme dans le cas du formalisme QFD². Dans le cas d'une approche comportementale, nous disposerons de représentations schématiques sommaires. Le modèle conceptuel représente le produit par des entités qui caractérisent les concepts mis en œuvre pour satisfaire aux besoins : ce peut être des formalismes objets tels que OMT³, ou relationnels tels que NIAM⁴. Le modèle descriptif est plus orienté vers l'aspect géométrique. Il fournit une description des constituants à l'aide d'entités élémentaires telles que surfaces, arêtes ou bien nœuds et facettes, qui pourra ultérieurement être utilisée dans une étude par éléments finis. Enfin, et nous arrêterons là notre recensement, nous trouvons la

1. Structured Analysis and Design Technique.

2. Quality Functional Deployment.

3. Object Modeling Technique.

4. Nijsen Information Analysis Method.

représentation par « features » ou « entités »⁵. Ces features, géométriques ou non, représentent une information concernant une pièce. L'information peut concerner une fonction de fabrication, une fonction technologique, une fonction de forme, etc. Ils donnent lieu le plus souvent à une représentation géométrique que l'on intègre directement aux logiciels de CAO. Ce concept de features est né du besoin de dépasser le strict cadre de modélisation géométrique lors de la conception en y intégrant des contraintes de fabrication. Il a débouché sur une nouvelle méthodologie de conception nommée « feature-based design »⁶. Les auteurs éprouvent des difficultés à donner une définition cohérente d'un(e) feature, tant les informations contenues peuvent être diverses. Néanmoins, le feature constitue un outil puissant d'intégration de connaissance dans une entité géométrique plus ou moins complexe, et cette capacité à transmettre une intention à partir d'un élément primaire de matière est le fondement même de notre démarche. On peut cependant constater une grande disparité dans la richesse des informations contenues dans les features et on a souvent tendance à les cataloguer en features de haut-niveau ou de bas-niveau selon leur complexité et la nature des informations. Une étape délicate de la conception est la sélection des features au vu des fonctions à remplir.

Il apparaît que ces différentes modélisations prennent des formes très diverses selon leur utilisation. Dans le cadre de la conception préliminaire, on trouvera une description plus détaillée des modèles et outils de représentation et de raisonnement dans [HW98]. Cette diversité permet de transmettre aisément des informations utiles aux opérations menées dans chaque approche (calculs, choix, etc.) en adaptant le modèle. Cette évolution de la représentation traduit aussi le besoin d'intégration de nouvelles connaissances dans le modèle produit. La recherche d'un modèle produit évolutif qui puisse intégrer des connaissances de natures différentes représente alors un enjeu important [Con96]. L'enrichissement de la base d'informations concernant le produit implique une structuration des données pour que le concepteur ne soit pas submergé. En conséquence, les modèles peuvent éluder une partie des informations acquises dans les phases amont qui s'avèreraient inutiles dans la tâche courante.

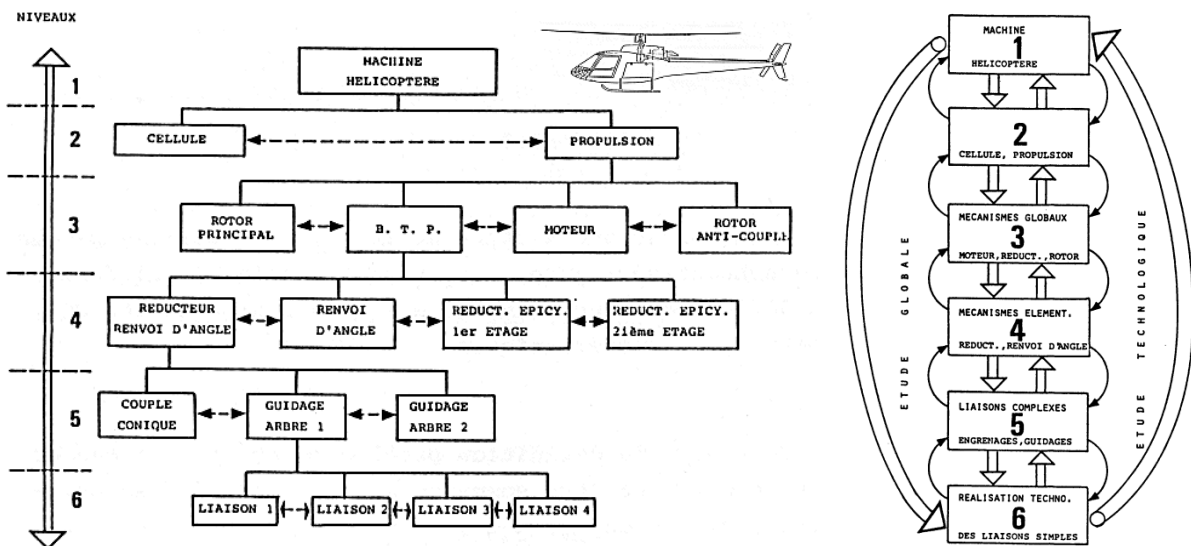


FIG. 1.6 – Niveaux de décomposition fonctionnelle selon [Gui87].

5. Le terme français n'étant pas plus explicite que son homologue anglophone, nous conserverons ce dernier dans la suite de l'exposé.

6. Ou « conception à base de features ».

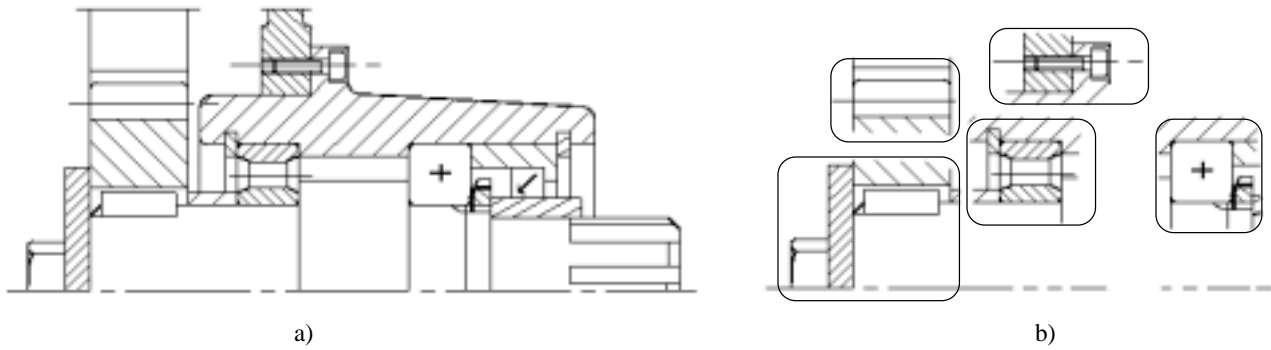


FIG. 1.7 – Décomposition d'un mécanisme en solutions technologiques selon [Gui87].

En prenant pour hypothèse que tout système mécanique peut être complètement décrit à partir de ses liaisons, J. Guillot propose un modèle de représentation de produit structuré en niveaux de définition [Gui87]. On en dénombre six, de complexité fonctionnelle décroissante, concernant l'exemple d'un hélicoptère (figure 1.6). Ces niveaux, dont le nombre ne nous apparaît pas primordial, sont associés à des définitions fonctionnelles du mécanisme. Il est donc évident que l'opération de décomposition n'a de valeur que pour les *systèmes mécaniques connus* pour lesquels on peut précisément identifier des *blocs fonctionnels*. Ces blocs sont des regroupements fonctionnels de haut niveau, telle la propulsion, ou bien de définition plus précise comme la boîte de transmission primaire de l'hélicoptère. Ils sont l'objet d'une connaissance experte et de méthodes de traitement connues et éprouvées. On notera que l'importance de la fonction est inversement proportionnelle à la précision de définition (figure 1.5). Notons encore que le niveau ultime de définition ne concerne pas les pièces constitutives du mécanisme, mais les «solutions technologiques» des liaisons mécaniques [GC84]. Ces éléments représentés sur la figure 1.7.b, sont les entités de base de la décomposition fonctionnelle pour lesquelles une connaissance et des méthodes de sélection existent. L'auteur rappelle que les décompositions sont toujours conditionnées par les niveaux supérieurs, c'est-à-dire que l'on recherche les découpages propres à satisfaire les contraintes issues des niveaux supérieurs. Dans une logique d'optimisation, et plus largement pour le bon déroulement des tâches de synthèse, on privilégie les décompositions qui génèrent le moins de couplage entre les blocs fonctionnels. Il est évident que l'opération de décomposition n'a de valeur que si l'on peut identifier précisément des blocs fonctionnels.

Cette analyse d'un système mécanique par niveaux fonctionnels constitue la base de notre réflexion. Cette analyse n'est pas propre au domaine du génie mécanique puisqu'elle a déjà été appliquée au génie civil dans le cadre d'un système de CAO [Alz93]. Nous avons tenté d'en extraire une méthodologie de travail proposant des opérations liées aux niveaux de définition. Nous présentons cette organisation opérationnelle dans la section 1.6.2.

1.6 Définition d'une organisation opérationnelle de la phase de synthèse.

Les concepteurs sont unanimes pour réclamer une certaine souplesse dans le processus de conception. Cela se traduit par une certaine liberté dans l'ordonnancement des tâches de conception. A ce titre, nous devons rappeler les deux grands courants de pensée qui président actuellement à l'organisation du processus de conception : l'approche « *ingénierie concourante* »⁷ et

7. "Concurrent engineering".

l'approche plus classique que l'on nomme *séquentielle*. La première approche semble plus séduisante compte tenu du déroulement simultané des tâches de conception qui laisse entrevoir une compression évidente des temps de développement. On doit cependant faire face à des problèmes de coordination des activités, et le temps de développement ne se résume jamais au temps de déroulement de l'activité la plus longue. La seconde approche, que l'on pourrait qualifier de traditionnelle, présente une structure figée qui facilite les échanges d'informations mais ne peut éviter des «retours en arrière» liés à des choix qui apparaissent inadéquats dans les phases avalées.

On peut raisonnablement penser qu'une technique efficace consisterait à combiner les deux approches.

1.6.1 Quelques exemples d'organisation de la phase de synthèse.

Certains auteurs ont élaboré des logiciels permettant de définir des niveaux de priorité dans les étapes de synthèse [AA00]. La démarche est basée sur les relations étroites existant entre les fonctions attendues et les paramètres de conception. Ces relations sont révélées par une *matrice de relations* dérivée de la *matrice de corrélation* utilisée dans la méthode QFD. Elles sont alors représentées par un graphique de type sagittal et constituent la base d'élaboration de l'organigramme de conception. Cet organigramme sert de canevas au déroulement de la phase de synthèse en proposant une chronologie d'actions. Ainsi, pour un mécanisme connu, on peut établir une planification des actions de conception visant à rationaliser les temps de développement. Cette méthode risque de devenir relativement lourde avec l'augmentation du nombre de fonctions et de facteurs de conception. De plus, l'évaluation qualitative des interactions fonctions/facteurs et de la prépondérance des fonctions attendues est très subjective et dépend fortement du niveau d'expertise de l'utilisateur.

Dans les premières étapes de la synthèse (particulièrement l'avant-projet), il n'est pas toujours facile de transcrire les fonctions attendues, souvent qualitatives. A. Medland propose une stratégie de décomposition de ces problèmes [MM00]. Les fonctions attendues sont transcrites en contraintes qui seront vérifiées dès qu'un niveau de définition suffisant sera atteint. Le mécanisme prend la forme d'un modèle paramétré dont les paramètres constituent les variables d'un problème contraint. Ce problème est décomposé selon la (ou les) méthode de décomposition la mieux adaptée, tandis que son intégralité est gérée par une méthode de maintien de cohérence. L'étude dévoile différentes stratégies de résolution, du changement minimal à la solution intégralement nouvelle, qui mènent à des solutions fort différentes. On remarque à ce sujet que certaines stratégies conduisent à des architectures du système (un vélo pour l'exemple) opposées à l'architecture primaire, ce qui n'est pas toujours souhaitable.

L'approche proposée par Pahl et Beitz nommée «systematic approach» [PB96] s'articule autour d'une méthode de factorisation qui consiste en un éclatement du système étudié en sous-systèmes plus simples à étudier. La résolution de chaque sous-problème, que l'on tente de rendre parfaitement indépendant, peut constituer une tâche dont le résultat servira à la construction du système (figure 1.8). Cette méthode conduit à la construction d'une structure de fonction grâce à laquelle on pourra mener une recherche des principes de fonctionnement⁸. Cette approche est souvent retenue pour l'automatisation du processus de conception.

La structure de type hiérarchique qui s'appuie sur un classement des fonctions de service de *complexité* décroissante est très fréquente. La *complexité* est souvent liée au manque, volontaire, d'information sur les relations entrées/sorties de cette fonction, ou à la sophistication

8. «working principles». Ce sont des sous-ensembles qui font apparaître les principes physiques ainsi que les géométries et matériaux caractéristiques aptes à remplir une fonction.

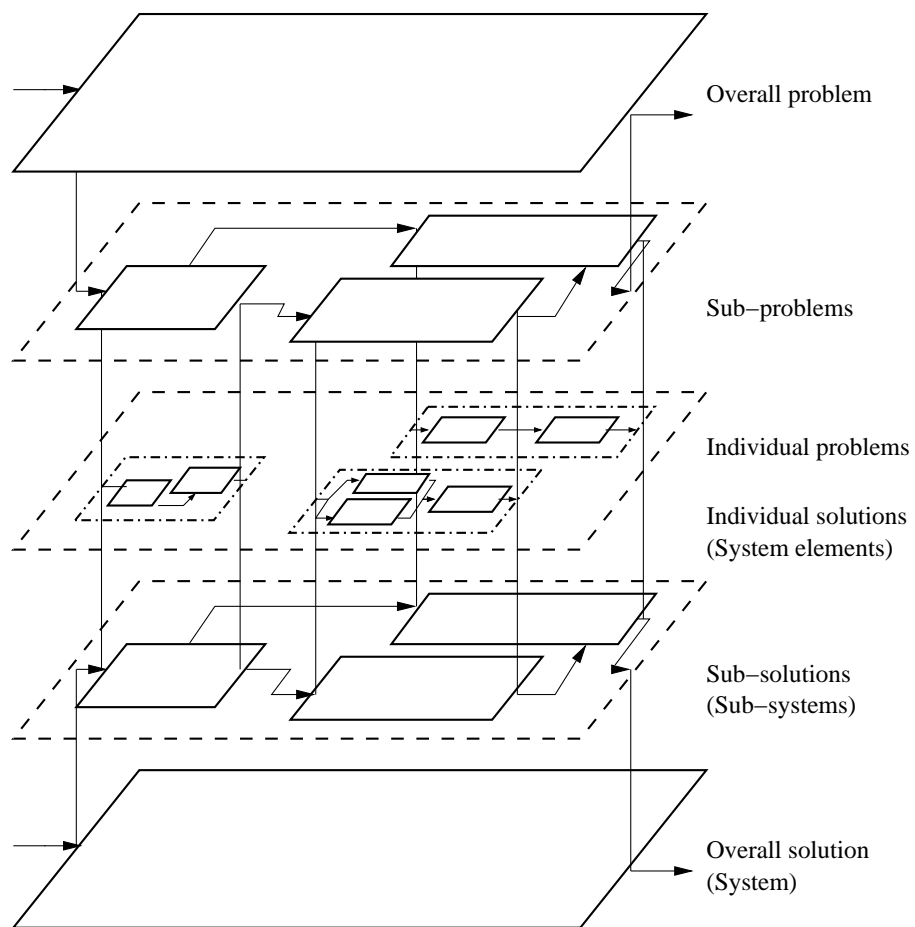


FIG. 1.8 – Méthode de fractionnement selon [VDI93].

de cette fonction. Une telle structure hiérarchique a déjà été proposée dans [Gui87]. Elle s'appuie sur la définition par niveau d'une machine (en l'occurrence un hélicoptère) et débouche sur la proposition d'un processus de conception de machine connue. Ce processus a fait l'objet d'un traitement informatique dont nous reparlerons ultérieurement. L'auteur a démontré qu'en partant d'une définition fonctionnelle globale descendante, qui mène aux caractéristiques particulières des éléments constitutifs, et ce en partant des caractéristiques les plus générales, on atteint un niveau seuil de définition correspondant à la définition des liaisons simples. De ce niveau, il suffit alors de « remonter », par une définition technologique, vers la machine. On note la proportionalité inverse de certaines caractéristiques du sous-système étudié comme, par exemple, la définition technologique qui croît à mesure que la taille du sous-système diminue (fig. 1.5). Dans l'optique d'une optimisation du produit conçu, cette organisation peut mener à un découpage facilitant la résolution du problème d'optimisation [BS70].

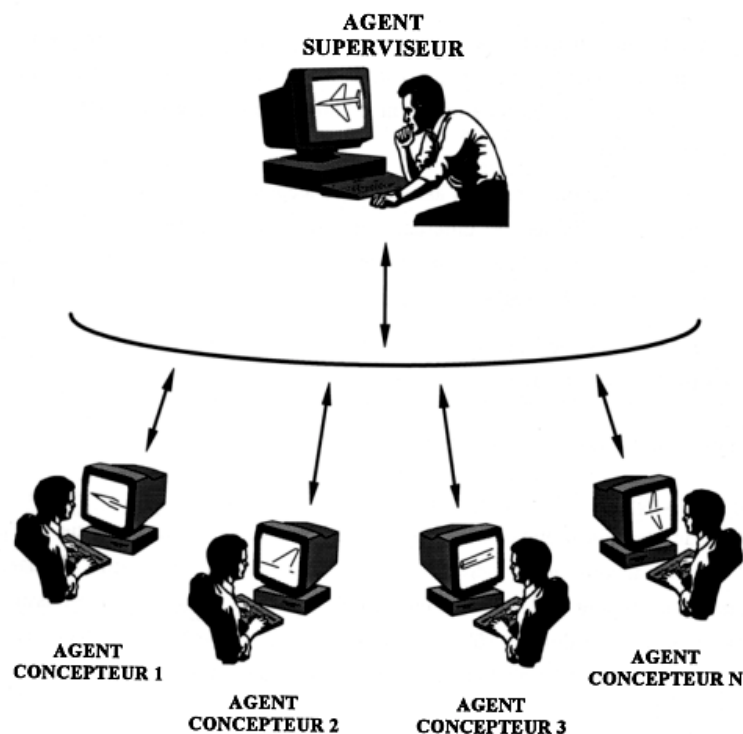


FIG. 1.9 – Conception distribuée selon [Deg96].

Le découpage entraîne un besoin d'échange d'informations durant la phase de conception. Afin de travailler en (quasi-)complète autonomie, les agents requièrent un certain nombre d'informations qui fixent leur cadre d'évolution. La quantité d'informations nécessaires dépend essentiellement du découpage fonctionnel et de la chronologie d'activation des différentes tâches de conception. Ces informations transversales seront caractérisées par des variables de liaison lors de la mise en équation des sous-problèmes. Selon ce principe de conception distribuée, I. Degirmenciyan propose une structure multi-agents [Deg96] dans laquelle le problème de conception est morcelé selon des critères d'expertise métier. Cette approche, apparentée aux concepts de l'intelligence artificielle distribuée, s'appuie sur une architecture « d'entités expertes » indépendantes, nommées *agents concepteurs*, en charge d'une partie de la conception. Les éventuels conflits naissants à l'interface des domaines d'étude de chaque agent concepteur seront pris en charge par un *agent superviseur*. Cela est rendu possible par la structure hiérarchique représen-

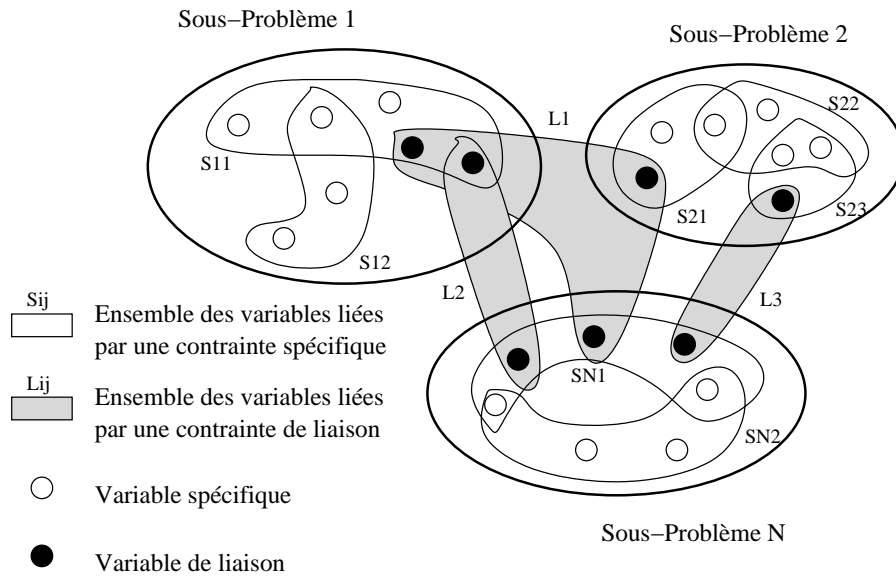


FIG. 1.10 – Formalisation d'un problème de conception distribué selon [Deg96].

tée figure 1.9. Cette structure s'adapte parfaitement au paradigme du choix contraint orienté par objectifs [Zha97]. Le problème de conception est donc formalisé comme un ensemble de sous-problèmes liés entre eux par des contraintes de liaisons (figure 1.10). On adopte, pour la résolution des conflits, une méthode hybride de distribution des contraintes de liaisons sur chaque agent concepteur, et sur l'agent superviseur pour l'arbitrage des conflits. Cette redondance de contraintes a des conséquences en termes de temps de calcul.

Le découpage hiérarchique que nous proposons ci-après est basé sur une analyse de l'ensemble des démarches que nous venons d'évoquer. Les observations qui ont orienté notre choix vers le principe de découpage de l'activité de conception en tâches sont les suivantes :

- La complexité des systèmes conçus est telle que l'on doit morceler le travail afin d'en répartir la charge sur des intervenants multiples.
- La rationalisation des tâches de conception mène à favoriser l'hyper-spécialisation des agents de conception, et donc à multiplier leur nombre.
- La diversité des métiers impliqués dans la conception, qui est un pendant de la complexité des produits développés, implique la présence d'agents concepteurs de compétences variées.
- La conception optimale des systèmes conçus ne peut actuellement se faire de façon globale. Il est nécessaire de produire un ensemble de sous-systèmes modélisables plus ou moins indépendamment, afin de simplifier la formulation mathématique des problèmes d'optimisation.
- Le mode de pensée humaine (mono-tâche) tend naturellement à séparer les activités cognitives. Il serait erroné de penser que la formidable puissance de calcul actuellement disponible permet de venir à bout des problèmes mathématiques de taille et de complexité importantes. Les processus de calculs en parallèle requièrent, eux aussi, une grande rigueur dans l'assignement des tâches.

Nous proposons donc, dans la partie suivante, une organisation opérationnelle qui tente d'allier les avantages du traitement parallèle avec une organisation séquentielle de l'activité de conception.

1.6.2 Organisation opérationnelle basée sur une définition fonctionnelle multi-niveaux.

La nécessité de découpage de l'activité de conception en tâches clairement identifiées est largement reconnue. Un indice de qualité de ce découpage sera son aptitude à isoler des sous-problèmes *indépendants* qui seront traités par les agents concepteurs⁹. L'indépendance se caractérisera par un nombre minimal de variables de liaison liant ces sous-problèmes. Le découpage permet aussi de définir plus rigoureusement le cadre d'étude des sous-problèmes en fixant, entre autres choses, les paramètres qui le régissent. L'autonomie ainsi fournie aux agents concepteurs serait le gage d'une pleine liberté d'investigation dans la recherche d'une solution optimale. Afin d'illustrer notre propos, nous utiliserons l'exemple de la grignoteuse présenté figure 1.11 et qui constitue un mécanisme générateur de fonction à un degré de liberté. Nous présentons ci-après la définition des cinq niveaux (ou tâches, notées T_i) que nous avons retenue pour notre organisation opérationnelle. Nous précisons pour chacun d'eux le mode opératoire, les éléments manipulés et le type de résultats obtenus.

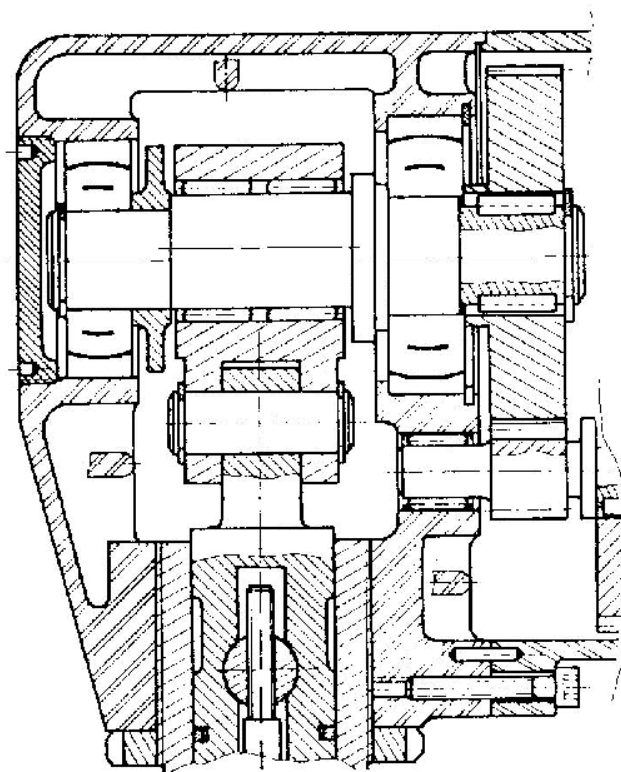


FIG. 1.11 – Représentation partielle d'un mécanisme de grignoteuse.

1.6.2.1 Construction du modèle cinématique du mécanisme.

Cette première tâche, que nous nommerons T_1 , a pour but de produire un modèle cinématique de mécanisme qui répond au mieux aux exigences du CdCF. Les données traitées à ce niveau de la conception sont d'ordre global: couple et vitesse transmissibles, disposition des arbres

9. Dans la suite de ce mémoire, le mot concepteur désignera la personne ou le groupe de travail en charge de l'activité de conception à l'instant précis décrit. On peut donc assister à un changement de concepteur au long du cycle de conception.

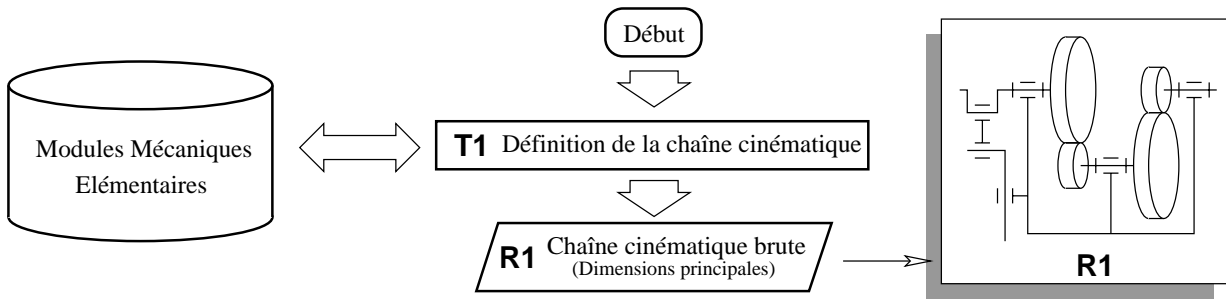


FIG. 1.12 – Découpage opérationnel : la tâche T1.

d'entrées et de sorties, encombrement... L'outil privilégié du concepteur dans cette tâche est l'esquisse. Il reflète la juste part d'approximation et d'indécision à laquelle le concepteur doit faire face. Le concepteur procède à un assemblage de « principes élémentaires de mécanismes » dont la mise en relation doit conduire à un modèle cinématique conforme au CdCF. Il existe des outils capables de fournir une assistance au concepteur à ce niveau de la phase de synthèse. L'un d'eux est présenté ci-après.

J.-C. Fauroux a mis au point un ensemble de modules informatiques [Fau99] qui apportent une aide au concepteur qui doit définir la structure cinématique d'un mécanisme de réduction de vitesse. Cette aide est organisée en trois étapes (une structurale et deux dimensionnelles) définies ci-après.

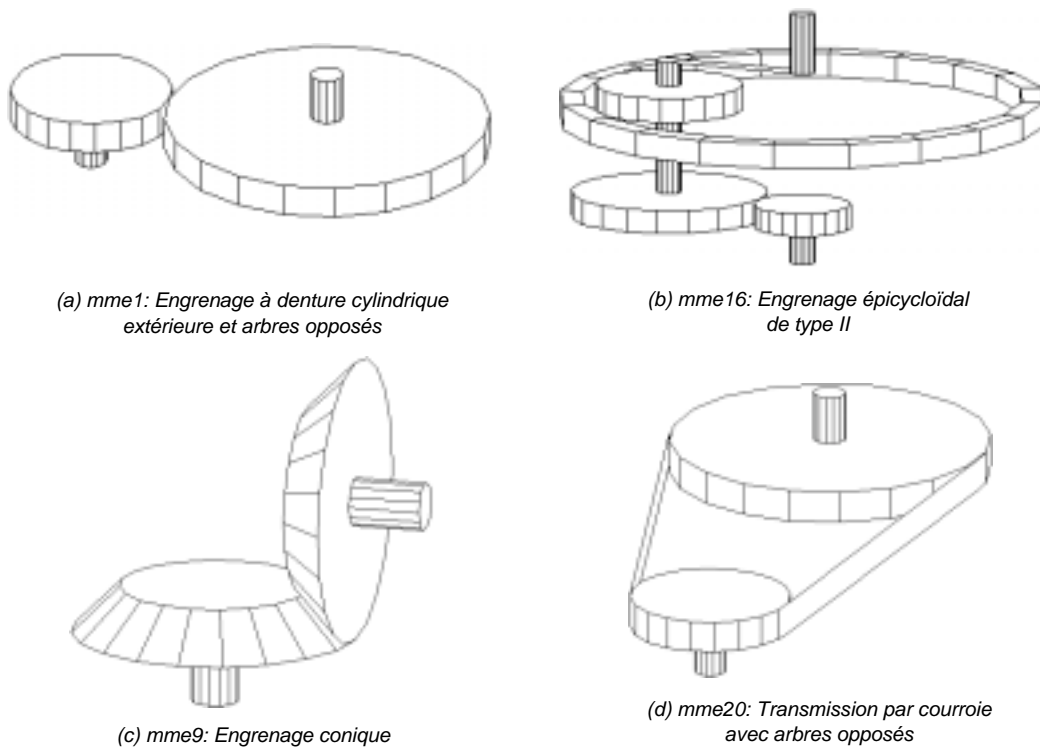


FIG. 1.13 – Exemples de M.M.E. d'après [Fau99].

1. Une étape de synthèse structurale détermine l'ensemble des combinaisons de M.M.E. (Modules Mécaniques Élémentaires) capables de générer la fonction cinématique souhaitée. Les M.M.E. constituent les briques élémentaires du système à ce niveau de la conception.

Leur représentation est schématique, et ils sont répertoriés et classés par fonction (la figure 1.13 présente quelques uns des M.M.E. considérés). L'outil exploite une bibliothèque de ces M.M.E. pour constituer, par association, des structures cinématiques candidates. Les résultats sont présentés sous forme d'un classement des structures, selon des critères qualitatifs de performance. L'extension de ce module aux transformations de mouvement rotation–translation [WGSF01] permet d'envisager le traitement du problème de la grignoteuse. Cette étape de synthèse structurelle pourrait alors mener à la solution {1 engrenage extérieur + 1 engrenage extérieur + 1 bielle-manivelle} représentée figure 1.12.

2. Cette structure est traitée pour donner un modèle géométrique appelé *squelette*. Le concepteur doit alors faire face à l'épineux problème de la fermeture de la chaîne cinématique. C'est ici que sont pris en compte certains éléments tels que les dispositions d'arbres.
3. Enfin, un premier dimensionnement approximatif des éléments primaires, tels que les arbres et les dentures, est réalisé afin de vérifier la conformité au CdCF (encombrement, puissance, etc).

Une représentation globale de cette tâche T1 est donnée figure 1.12. On y voit les données d'entrée issues du CdCF ainsi que le type de données nécessaires au concepteur pour la synthèse de la structure (en l'occurrence une bibliothèque de M.M.E.). Le résultat (R1) obtenu est la structure d'un mécanisme répondant aux exigences de génération de fonction du cahier des charges de la grignoteuse. On remarque qu'à ce stade de la synthèse, les liaisons telles que les pivots ne sont pas détaillées.

L'assistance fournie au concepteur lors de cette étape peut prendre d'autres formes : il peut s'agir d'heuristique d'assemblage, de catalogues de solutions ou de systèmes experts dédiés à la sélection et l'assemblage d'éléments de base. L'utilisation d'un outil informatique permet de fournir un modèle et une visualisation des résultats obtenus. C'est un des caractéristiques de CASYMIR [Fau99] qui propose une représentation 3D de la chaîne cinématique avec une possibilité d'animation de la structure choisie.

1.6.2.2 Détermination de la nature des liaisons.

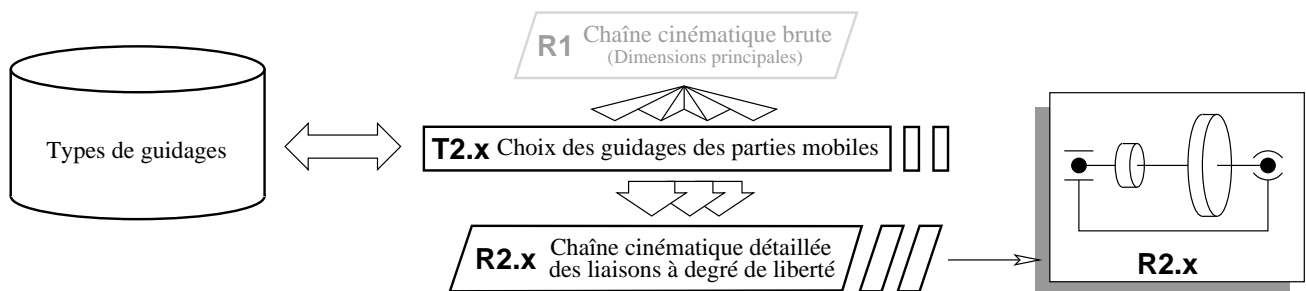



FIG. 1.14 – *Découpage opérationnel : la tâche T2.*

A l'issue de la tâche T1, le concepteur dispose d'un modèle cinématique grossièrement dimensionné. Il va désormais se consacrer à la définition de la nature des liaisons qui assureront le guidage des ensembles mobiles. Nous avons vu précédemment que les liaisons mécaniques étaient peu détaillées. Le concepteur va s'attacher à préciser ces liaisons durant la tâche T2 (figure 1.14). Il va, par exemple, devoir choisir la disposition des paliers qui réaliseront les liaisons pivots. Cela

peut se traduire par un découpage de la fonction *liaison pivot* en un assemblage de *liaison rotule* et *liaison linéaire annulaire*. Le fonctionnement du système sera fortement dépendant de la topologie retenue pour les paliers. Un positionnement approximatif est réalisé ; il sera affiné ultérieurement par la phase de synthèse des contraintes de montage (section 1.6.2.3). Il peut paraître surprenant de ne définir qu'une partie des attributs des liaisons de guidage. Le concepteur pourrait vouloir définir également le type de roulement qu'il compte associer à chaque palier. Ceci est imposé par le principe de découpage qui doit rejeter tout élément non strictement nécessaire dans cette étape du processus, et qui risquerait de sur-contraindre les étapes suivantes.

Le concepteur peut faire appel à des *règles métiers* pour éliminer des solutions qui peuvent lui paraître, où qui se sont avérées par le passé, non satisfaisantes ou irréalisables (montage en porte-à-faux par exemple). Il pratique alors une estimation dimensionnelle pour finir de classer ces solutions. Les calculs menés ici concernent la statique des appuis, et servent à arbitrer le classement selon les objectifs fixés.

Il est important de noter que cette tâche peut être scindée en plusieurs activités (T2.1, T2.2... T2.x), menées en parallèle, qui concernent les sous-ensembles cinématiquement équivalents tels que représentés en R2.x (figure 1.14). Ces sous-ensembles cinématiquement équivalents sont composés d'éléments n'ayant entre eux aucun mouvement relatif (ligne d'arbre par exemple). Ce partitionnement est fortement dépendant des choix faits durant T1. On éclate (symbole ) le système issu de T1 en de multiples sous-systèmes que l'on traite indépendamment, comme les lignes d'arbre.

1.6.2.3 Synthèse des contraintes de montage.

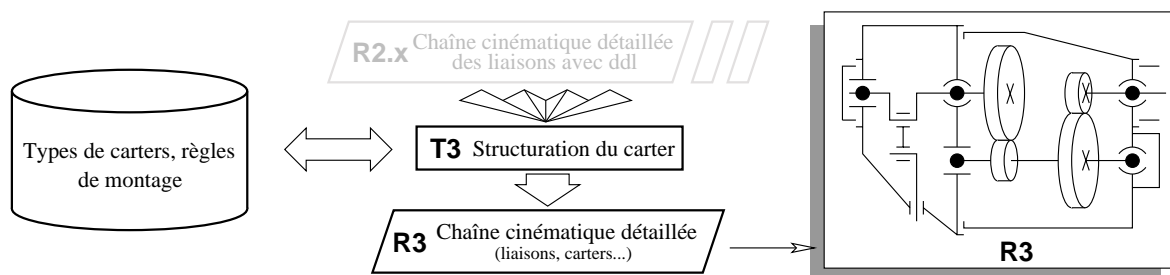



FIG. 1.15 – *Découpage opérationnel : la tâche T3.*

La tâche T3 étudie le montage de l'ensemble du mécanisme. Pour cela, on regroupe les sous-ensembles (symbole ) que l'on a pu former en T2 et l'on considère à nouveau le mécanisme global de la tâche T1, augmenté des attributs définis pendant T2. On s'attache à affiner la position des éléments de guidage en définissant aussi la morphologie du carter et les sens de montage. L'introduction d'un voile dans le carter pour supporter les paliers de deux arbres parallèles entraînera un alignement des paliers sur ce voile (voir figure 1.15). Cette « vision » globale du mécanisme est aussi l'occasion d'appréhender les risques d'interférence entre les composants de guidage des sous-ensembles. Les problèmes de montage, d'encombrement et d'étanchéité sont identifiés dans cette phase. Ceci est rendu possible grâce aux compléments d'information apportés par la définition partielle des guidages. La définition du carter et des règles de montage ne peuvent être menées à bien qu'à partir d'une vision d'ensemble du mécanisme. Ces choix structurels entraînent l'introduction et la définition de nouvelles liaisons qui n'apparaissaient pas dans T1. Il s'agit de liaisons complètes dites démontables (ex : entre une roue dentée et un tronçon d'arbre) qui naissent des options prises concernant les règles de montage.

Ces liaisons, dont on ne connaît par encore la réalisation technologique, permettent le montage du mécanisme. On fixe ainsi la topologie du système et de ses composants. Le résultat obtenu lors de cette tâche est représenté par le schéma noté R3 (figure 1.15) que l'on peut nommer *schéma d'architecture détaillée*.

1.6.2.4 Choix et définition des DTLM.

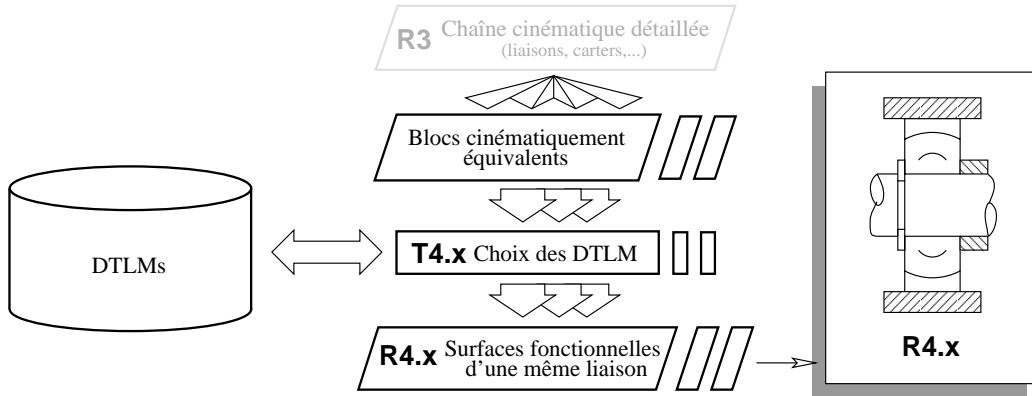



FIG. 1.16 – *Découpage opérationnel : la tâche T4.*

A partir de la structure générale ainsi esquissée, on peut à nouveau éclater le système en sous-ensembles cinématiquement équivalents. On procède, parallèlement pour chaque sous-ensemble, à une définition beaucoup plus précise de toutes les liaisons déjà mises en évidence au moment de la définition des types de guidages (Tâche T2, section 1.6.2.2) et au moment de la définition des liaisons complètes (Tâche T3, section 1.6.2.3). Ce travail se traduit par le choix de « solutions technologiques » que nous nommerons DTLM, pour *Dispositifs Technologiques réalisant une Liaison Mécanique*, et leur dimensionnement. Il s'appuie sur une base de données qui fournit, pour chacune des fonctions susceptibles d'être étudiées durant cette tâche, l'ensemble le plus exhaustif possible de DTLM. Un DTLM est un ensemble d'éléments mécaniques capable de concrétiser matériellement une liaison mécanique conventionnelle¹⁰. A titre d'exemple, la figure 1.16 représente, en R4.x, un DTLM qui concrétise une liaison linéaire annulaire. Il est constitué d'un roulement-rotule à billes, d'un tronçon d'arbre, d'une partie de carter ainsi que d'une entretoise et d'un anneau élastique. Ce groupement de petits volumes de matière met en place les surfaces fonctionnelles. Ces volumes peuvent correspondre à des composants standards complets (roulement, anneau élastique...) ou ne constituer que des parties de pièces spécifiques, celles-ci n'étant pas encore définies. L'étude des DTLM et de leur mode opératoire est l'objet du chapitre suivant. La figure 1.16 décrit l'organisation de la tâche T4. La définition des DTLM peut être faite en parallèle, ceux-ci disposant des contraintes d'encombrement issues des niveaux supérieurs.

1.6.2.5 Habillage des pièces du mécanisme.

La tâche T5 est la tâche de plus bas niveau, la plus détaillée de l'étape de synthèse. Elle est consacrée à la concrétisation des pièces (figure 1.17). Elle est précédée d'une redistribution (symbole ) des formes et volumes fonctionnels issus des DTLM définis dans T4. Ces éléments sont réunis par entités pièces, déjà partiellement définies, et complétées par un ajout de matière reliant les volumes d'une même pièce. Cette redistribution autorise un traitement en parallèle

10. Parmi la liste des liaisons conventionnelles figurent les liaisons pivot, pivot glissant, glissière, complète, etc.

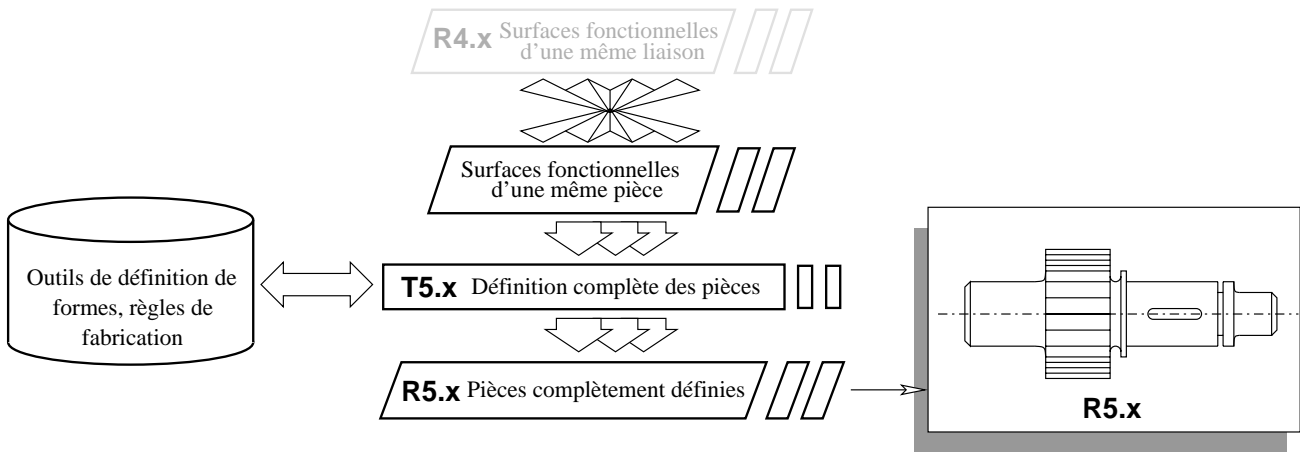


FIG. 1.17 – *Découpage opérationnel : la tâche T5.*

de l'activité. Cette tâche fait appel à des compétences dans le domaine du calcul des structures (Résistance des Matériaux, Éléments Finis) et aussi de la fabrication. En effet, dans cette phase de définition de formes, les connaissances du métier des fabricants peuvent jouer un rôle important. Ceci s'explique par le fait que, hormis les contraintes de type mécanique, le choix des formes est libre puisque non fonctionnel du point de vue concepteur. Les résultats issus de ce niveau sont les dessins de définition des pièces (voir R5.x).

1.6.3 Remarques sur le découpage proposé.

Cette proposition d'organisation de la phase de conception est reconstituée sur la figure 1.18 (organigramme et types de résultats obtenus pour chaque tâche). Cette organisation nous servira de canevas pour la suite de notre travail. Cette phase de synthèse est bien entendu suivie de la phase d'analyse qui va reconstituer le système par une démarche ascendante, en vérifiant la cohérence de la solution retenue. Rappelons que cette approche par niveaux autorise un retour en arrière à chaque étape de sa progression. Il est donc possible de remettre en cause les options choisies précédemment si elles ne s'avèrent pas aussi judicieuses ou performantes qu'escomptées. L'intérêt de ce découpage réside cependant dans la hiérarchisation de l'importance des fonctions qui doit tendre à limiter les « remontées » par une synthèse progressive du système. Ceci peut s'expliquer par la structure arborescente de l'arbre de décision. A chaque niveau, le concepteur établit une collection de solutions qu'il classe. Il s'engage alors sur une voie pour laquelle le niveau suivant proposera une nouvelle collection de solutions. C'est une combinatoire qui peut aboutir à l'issue du dernier niveau de définition à des millions de combinaisons. L'avantage du découpage est donc de proposer l'exploration de toutes les solutions du niveau directement supérieur à celui du cas d'échec. Cela implique une certaine systématisation dans la recherche des solutions qui permet d'envisager toutes les solutions du niveau (n) avant de remettre en cause le niveau ($n - 1$). On respecte donc le niveau d'importance des options prises en amont du processus (ceci peut éviter les situations absurdes où l'on remet en cause la structure du système parce qu'une clavette, par exemple, s'avère trop longue!). Le découpage induit une certaine rationalisation des tâches. Compte tenu de la part importante que prend la conception routinière dans toute activité de conception, le découpage permet la centralisation des connaissances juste nécessaires à un niveau donné. Leur exploitation future n'en sera que facilitée [KS96]. Le traitement de chaque tâche peut être réalisé avec la méthode la plus adaptée, c'est-à-dire celle qui est réputée avoir les meilleures performances en terme de production de solutions. On conçoit alors que ce

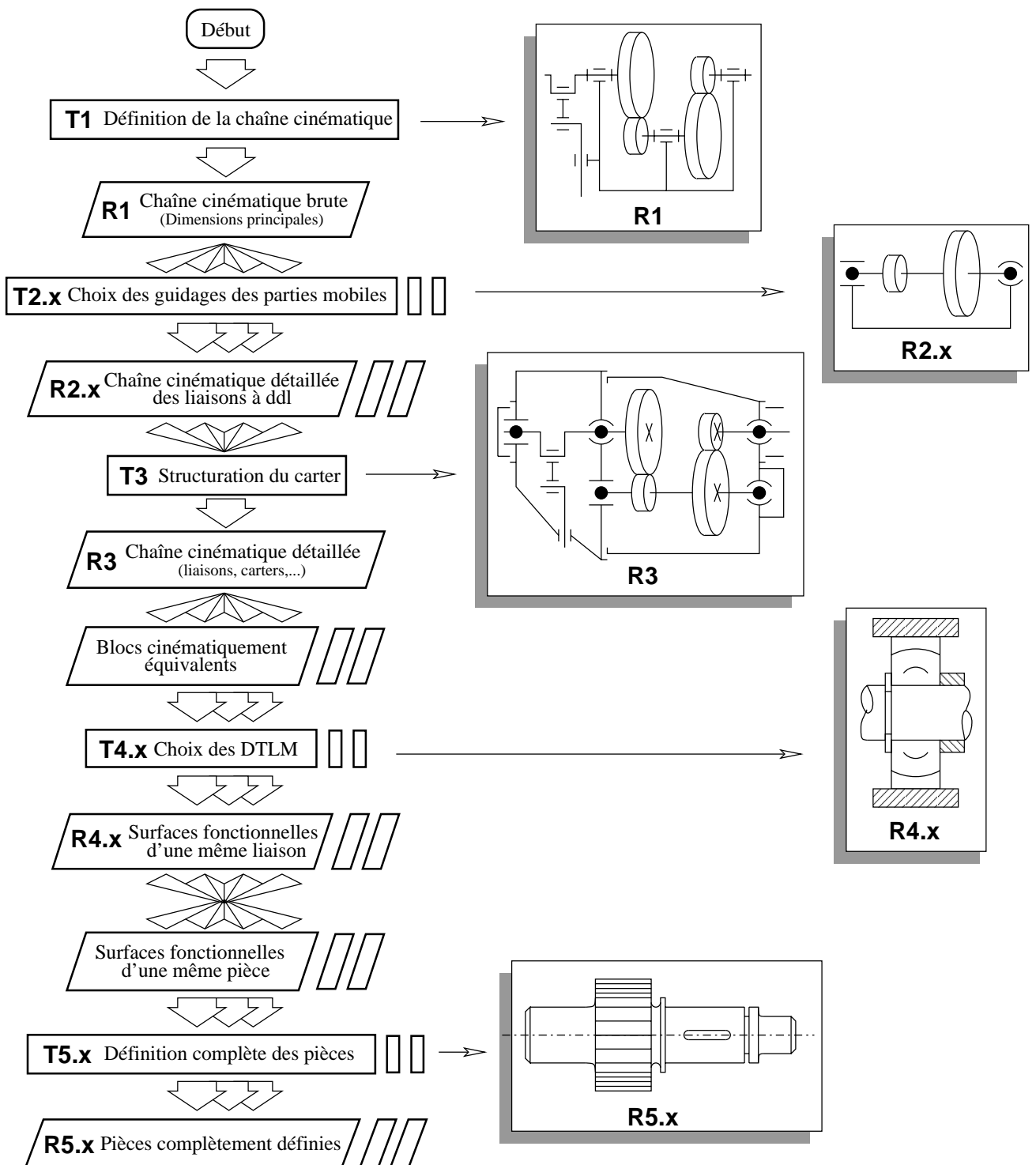


FIG. 1.18 – Découpage opérationnel de la phase de synthèse.

découpage favorise l'hétérogénéité des méthodes de résolution de problèmes.

Nous allons maintenant nous pencher sur la problématique de l'assistance aux concepteurs. Les questions essentielles que nous pouvons nous poser sont :

- Quand doit-on assister le concepteur? Il s'agit de préciser les phases de conception où l'apport de l'informatique peut être appréciable.
- Comment doit-on assister le concepteur? Quelle forme doit prendre l'outil d'assistance pour être assuré d'une efficacité optimale?

Avant de tenter de répondre à ces questions, nous allons préciser le rôle et l'importance de l'ordinateur dans la conception mécanique ainsi que ses champs d'actions.

1.7 Assistance informatique aux concepteurs.

Les méthodes de conception ont évolué avec l'avènement de l'ordinateur. Le processus de conception a recours à de nombreux calculs utiles à la prévision des comportements du mécanisme. Les modèles mathématiques mis en œuvre sont de complexité croissante. Dès lors, il n'est pas surprenant de constater que l'informatique a pris une place importante dans cette activité, et ce depuis une trentaine d'années. L'étude historique de la CAO nous montre à quel point l'assistance informatique aux concepteurs a muté. Elle tend aujourd'hui vers la conception intégrée, notamment en essayant d'intégrer des outils récents comme l'intelligence artificielle. Nous fixerons, en fin de chapitre, les objectifs que vise notre système d'aide à la conception intégrée.

1.7.1 Historique de la Conception Assistée par Ordinateur.

Originellement dédié au calcul, l'ordinateur a évolué vers un outil de représentation novateur en introduisant ce que le dessin technique a toujours peiné à représenter : la troisième dimension. L'apparition de l'ordinateur a bien entendu entraîné son cortège d'espoirs et de désillusions. Il a généré de nombreux excès dont une nouvelle sémantique, la «x.A.O.», où l'on peut, à loisir, remplacer le x par à peu près tout ce que l'on veut (C, D, F, CM, TG...). Dans la suite de cet ouvrage, nous parlerons de CAO, bien que celle-ci ne soit qu'une entité réduite de la CFAO.

Nous assistons ces dernières années à une remise en question fondamentale de nos méthodes de conception. Les concepteurs ont bien compris qu'avec ce nouvel outil, il était sûrement possible d'adopter de nouvelles méthodes de travail, mais la question de l'existence de telles méthodes reste posée. Lorsque sont apparus les premiers logiciels de DAO, il y a environ 30 ans, le but clairement avoué était double : pouvoir modéliser de nouvelles géométries et apporter un gain de temps considérable à la principale activité du bureau d'études, à savoir le dessin. Les pionniers que furent Fayet de Casteljaud et Bezier ont définitivement installé, non sans peine, l'informatique dans les bureaux d'études [Poi89] grâce à l'introduction des surfaces complexes. Les services rendus par ces logiciels (Unisurf) peuvent désormais paraître dérisoires mais ils n'en ont pas moins contribué à imposer comme « assistant de l'ingénieur » le logiciel de DAO, notamment en diminuant sensiblement le temps dévolu à la représentation graphique, jusque-là assurée manuellement ! Quelles différences séparent donc le DAO et la CAO? Elle tiennent en plusieurs points :

- Le DAO est une forme de représentation purement mathématique des entités géométriques que manipule l'ingénieur mécanicien. En ce sens, il est l'héritier direct de la géométrie

descriptive. Le DAO est souvent vu comme l'informatisation du dessin technique. La CAO regroupe les divers métiers de la conception dont le dessin technique. Elle forme un ensemble cohérent d'outils informatiques d'assistance au concepteur en proposant des applications utiles à tous les stades du développement du produit. Elle manipule des modèles exploitables selon les différents points de vue de spécialistes.

- Le DAO (qui disparaît progressivement) n'intervient que tardivement dans la phase de conception, alors que les formes sont pratiquement définies. Il constitue une ultime étape de validation des choix effectués auparavant. La CAO intervient beaucoup plus tôt dans la phase de conception. Elle permet, par exemple, de valider des choix de structure par une analyse cinématique avant même d'avoir défini précisément les pièces du mécanisme.
- Enfin, la CAO est une organisation pluri-disciplinaire qui intègre le DAO en tant que module de visualisation et de représentation géométrique.

Les logiciels actuels fournissent beaucoup plus de services aux concepteurs. Des disciplines comme l'étude par éléments finis doivent leur existence à la percée des calculateurs. Les modeleurs 3D se sont généralisés, certains livrant même leur code source au domaine public [EM01]. Ce mode de représentation se généralise mais pose encore quelques problèmes de compréhension aux concepteurs habitués à une représentation 2D. Ces difficultés sont principalement causées par le manque de repères face à ces nouvelles représentations : alors que le concepteur expérimenté visualise instantanément les éléments de base d'un dessin 2D classique (et donc les intentions de conception), il éprouve plus de difficulté à lire un modèle 3D, principalement à cause du manque d'habitude de ce mode de représentation. Ce handicap est vite surmonté moyennant une période d'apprentissage. L'offre en matière d'analyse de mécanisme se standardise, et cela dans un contexte de démocratisation croissante. Les grands fabricants de logiciel déclinent des versions individuelles limitées de leurs logiciels phares (I-DEAS[®], CATIA[®]) ou proposent des logiciels spécifiques (SOLIDWORKS[®], THINK3[®]). Ce développement du logiciel de CAO dit *économique* représente 1/4 du marché en valeur (estimé à 1 Milliard de \$ en 2000) et 1/2 du marché en nombre d'unités [Pre00]. Les arguments décisifs lors d'un tel investissement sont :

- Le coût. Il consiste principalement en une sommation des coûts d'investissement et d'exploitation du matériel ainsi que des logiciels. Les études récentes sur le marché du logiciel de CAO montrent à quel point l'enjeu économique est important pour le client et les offres « allégées » séduisent de plus en plus de P.M.E.
- Le temps. Plus difficilement quantifiable, il donne une notion d'intégration de l'ordinateur dans les méthodes de travail. Ce chiffre est cependant délicat à exploiter pour les raisons suivantes. La phase de conception englobe de plus en plus d'activités et s'étale dans le temps. A ce titre, il devient difficile de cerner la durée et le contenu exact de l'intervention de l'ordinateur dans la phase de conception. De plus, les logiciels utilisés sont de natures diverses et variées (gestion, dessin, calcul, ergonomie, stylisme¹¹...). Ceci est principalement lié à la diversité des acteurs intervenant dans la phase de conception et à la spécificité de leur métier¹².

11. Nous emploierons ici le mot *stylisme* plutôt que *design* pour deux raisons. La première, qui peut paraître évidente, est notre attachement à la langue française, tandis que la seconde est plus sémiologique: le mot *design* recouvre un sens beaucoup plus général pour les anglo-saxons que le sens « esthétique » que lui accorde la langue française.

12. On entend par métier la spécialisation de la tâche accomplie par l'acteur et les méthodes s'y rapportant.

L'ordinateur est aussi un moyen de communication multimédia qui autorise l'échange d'informations de toutes natures. Les premiers catalogues informatiques ont vu le jour dans les années quatre-vingts, d'abord proposés sur support disquette, puis sur cédéroms pour profiter de l'augmentation de la capacité de stockage. Notons que cet accroissement du volume de stockage n'a pas contribué à l'augmentation du nombre de références proposées, mais à un enrichissement des modèles et informations stockées. Les composants mécaniques qui profitent de ces supports sont surtout les composants standards de la mécanique tels que les roulements [SKF01], les composants de fixation ou les composants d'étanchéité. La sélection de matériaux est aussi à l'honneur avec des logiciels tels que CAMBRIDGE ENGINEER SELECTOR [Ash00] ou son dérivé FUZZYMAT [BAS98], exploitant des indices de performances des matériaux pour effectuer une optimisation « multicritère ». Plus simplement, MATWEB propose une base de données matériaux en libre accès [AC01]. Plusieurs griefs sont souvent formulés à l'encontre de ce genre de supports :

- Ils représentent un coût non négligeable pour l'entreprise, même s'ils sont bien meilleur marché que leurs équivalents papiers.
- La rapidité d'évolution des produits et de leur disponibilité sur le marché s'accomode mal de mises à jour semestrielles ou annuelles. Ce manque de réactivité constitue un handicap certain pour les entreprises innovantes dont l'offre évolue à une fréquence élevée.
- Ils mobilisent un ordinateur pour leur exploitation, limitant ainsi considérablement le champ d'utilisation de cette ressource.
- Ils n'autorisent qu'un mode de communication unidirectionnelle, du fournisseur vers le client, selon les attentes supposées de ce dernier.

Des fournisseurs ont donc décidé de se lancer dans l'aventure de l'internet, par le biais du protocole `world wide web` (ou `www`) et proposent des composants de plus haut niveau tels que des moteurs électriques, vérins ou autres éléments de liaison cinématique sur leurs sites propres, ou sur des sites en commun tel celui de MDCYBERCAD. Plusieurs avantages à cette technologie : pour le client, une rapidité d'accès à des données réactualisées fréquemment et la possibilité de récupérer les seules informations qu'il juge utiles (et non le catalogue intégral). Pour le fournisseur, une maintenance aisée de la base de données, une plus grande couverture géographique, et un coût d'exploitation moindre.

Les données proposées sur les cédéroms sont souvent des fiches descriptives des caractéristiques principales du produit, accompagnées de modèles 2D ou 3D qui tentent d'utiliser un format d'exportation standard (DXF, IGES, STEP). Certains intègrent des logiciels d'aide au choix qui permettent un accès indirect à la base de données. On reproche cependant une pauvreté des informations qui ne permet pas au concepteur de mener à bien une étude intégrant ces produits sans une information complémentaire. Le Web autorise une interactivité plus grande entre le client et le fournisseur, à condition d'intégrer des architectures matérielles et logicielles adéquates [Tum00].

Outre la communication par le réseau, l'enjeu de la CAO de demain repose sur la *conception intégrée*. Nous étudions dans le chapitre suivant ce que peut être un système de conception intégrée, et quelques réalisations présentées comme telles.

1.7.2 La conception intégrée.

Pour M. Loyer, le logiciel de CAO/DAO ne passera du statut de « boîtes à outils » à celui de « système CAO » que lorsqu'il aura appris le métier de l'entreprise. Cet apprentissage passe

inévitablement par une intégration des méthodes de travail de l'entreprise. Pour S. Tichkiewitch, *l'intégration* doit signifier une intégration de tout le cycle de vie du produit dans la phase de conception [Tic94]. Elle doit aussi signifier l'intégration du contexte et de l'environnement de développement du produit. Enfin, elle signifie l'implication de tous les corps de métier dans la phase de conception. En conséquence, le domaine est tellement vaste, que la réalisation d'un système universel comprenant les outils nécessaires à une telle réalisation s'avèrerait vite impossible à envisager. On doit donc se résoudre à aborder le problème de la conception en proposant une palette de solutions fortement orientées métiers. La mise en commun de tous ces systèmes pourra éventuellement se faire grâce aux travaux menés sur les modèles de représentation et d'échanges de données qui autoriseront la communication entre ces systèmes. Cela ne règlera pas pour autant le problème de l'organisation des tâches.

Nombreux sont les concepteurs qui constatent l'absence, pour l'heure, de plates-formes informatique répondant aux critères de la conception intégrée. S'il est difficile de donner une définition de la conception intégrée, on peut néanmoins recenser les traits caractéristiques que s'accordent à reconnaître les acteurs de la conception intégrée, et sans aller jusqu'à présenter les quinze points soulevés dans [SK93], nous retiendrons les suivants :

- l'intégration de toutes les activités de conception et leurs contraintes afférentes ;
- une modélisation permettant la représentation et l'exploitation de toutes les connaissances investies dans la conception ;
- un processus de maintien de la consistance du projet ;
- une gestion des conflits dans les processus coopératifs qui implique un dialogue efficace et pertinent entre agents concepteurs ;
- une certaine souplesse dans le cheminement de la conception (possibilité d'itérations, de retour en arrière) qui peut se traduire par une structure ouverte du système ;
- le traitement et l'archivage des données qui peuvent être remobilisées et réinvesties dans le cadre d'une reconception ou d'une extension de la gamme de produits ;
- une interface puissante homme-machine qui soit apte à dialoguer en termes d'expertise avec le concepteur ;
- une transparence complète des modes décisionnels et de leurs éléments (critères de choix, règles d'évaluations, classement des solutions concurrentes. . .)

Cette liste n'est pas exhaustive, mais elle constitue un embryon d'objectif idéal vers lequel doit tendre tout système intégré de conception.

1.7.3 Quelques exemples de systèmes intégrés de conception.

Nous ne présenterons pas de logiciels commerciaux dans ce chapitre, car il ne reflètent pas encore, à notre sens, ce que doit être un système intégré de conception. Ces logiciels présentent une collection d'outils utiles à différents stades de la conception, mais n'apportent que sporadiquement une aide au concepteur. La plupart ne disposent d'aucune base de connaissances, ce qui peut paraître logique compte tenu de la spécificité métier des utilisateurs potentiels, mais ne proposent pas non plus la possibilité d'en constituer.

Nous détaillerons dans un premier temps un système de conception intégré présentant un traitement de type algorithmique : il s'agit de SICAM. D'autres travaux ont porté sur des systèmes intégrés basés sur des moteurs d'inférence de systèmes experts. Ceci relève du domaine de l'Intelligence Artificielle, ou I.A. Les systèmes experts apportent au concepteur un savoir-faire « métier » qui leur permet de consacrer à la conception innovante le temps gagné sur les tâches de conception routinière (que l'on estimait il y a peu, à 80% du volume de travail d'un bureau d'études) [Bea98]. Les systèmes experts que nous présentons ci-après ont la particularité de s'appuyer, pour les uns sur un moteur d'inférence développé au sein même du laboratoire de recherche (MÉCAEXPERT), et pour les autres sur un moteur d'inférence commercialisé (SMECI).

1.7.3.1 SICAM.

Le système SICAM¹³ met en application l'analyse faite par J. Guillot d'un processus de conception de machine connue (voir chapitre 1.5) pour les niveaux 5 et 6 (figure 1.6). Il « permet la conception de tous mécanismes composés d'engrenages montés sur des paliers à roulement et comprenant des liaisons arbres/moyeux par clavettes ou par cannelures. . . ». Le principe de SICAM est de considérer la conception d'un mécanisme comme étant l'assemblage des solutions technologiques des différentes liaisons. A chaque solution technologique correspond un algorithme de recherche des solutions optimales propres [Joh80]. Ces différentes solutions technologiques sont archivées dans un catalogue accessible grâce au Système de Gestion de Base de Donnée (SGBD) SOMINE®.

L'architecture de SICAM est modulaire. Chaque module est en charge de la conception d'un type de liaison. Le déroulement de la séquence de conception est, à peu de choses près, fixé par l'ordre de montage des éléments sur l'arbre. Cette technique permet de gérer facilement les contraintes de montage par propagation directe entre solutions technologiques voisines. Elle présente cependant l'inconvénient d'imposer une chronologie de définition des solutions technologiques au concepteur. Le système propose toutefois un retour en arrière pour modifier les solutions technologiques définies précédemment qui pourraient bloquer la recherche de solutions, mais les données associées aux solutions technologiques en cours de définition sont alors perdues. Cette chronologie rigoureuse dans la définition des solutions technologiques permet une gestion simplifiée des contraintes de montage. La gestion deviendra plus problématique dès lors que le concepteur sera libre de choisir la chronologie des opérations de définition des solutions technologiques. Un des avantages de SICAM est l'assistance « locale » au concepteur dans le choix même de ses solutions technologiques. Nous l'expliquons de la façon suivante. Lorsque le concepteur choisit une solution technologique, il base sa réflexion sur son expérience, i.e. principalement son intime conviction que cette solution technologique *s'adaptera* bien dans le contexte qu'il a créé. La synthèse dimensionnelle peut confirmer ou infirmer cette conviction. Elle sert aussi à guider le concepteur dans le choix à venir des solutions technologiques en lui donnant un aperçu qualitatif de ce que sera son système final à partir d'échantillons que constituent les solutions technologiques déjà définies. Cette notion de progression dans la définition est, à notre sens, primordiale lors de la phase de conception préliminaire. C'est une aide que ne peuvent fournir les logiciels qui traitent le problème de façon globale.

Les principaux griefs que nous pouvons formuler à l'encontre de SICAM concernent son manque de souplesse (dû au traitement algorithmique) et son domaine d'application restreint. Le système SICAM, programmé sur BULL® DPS68M n'est plus exploitable techniquement par suite de l'évolution normale du matériel informatique. L'utilisation du langage FORTRAN, couplé à des modules graphiques spécifiques au calculateur, limitait aussi considérablement les possi-

13. Système Intégré de Conception Assistée de Mécanismes.

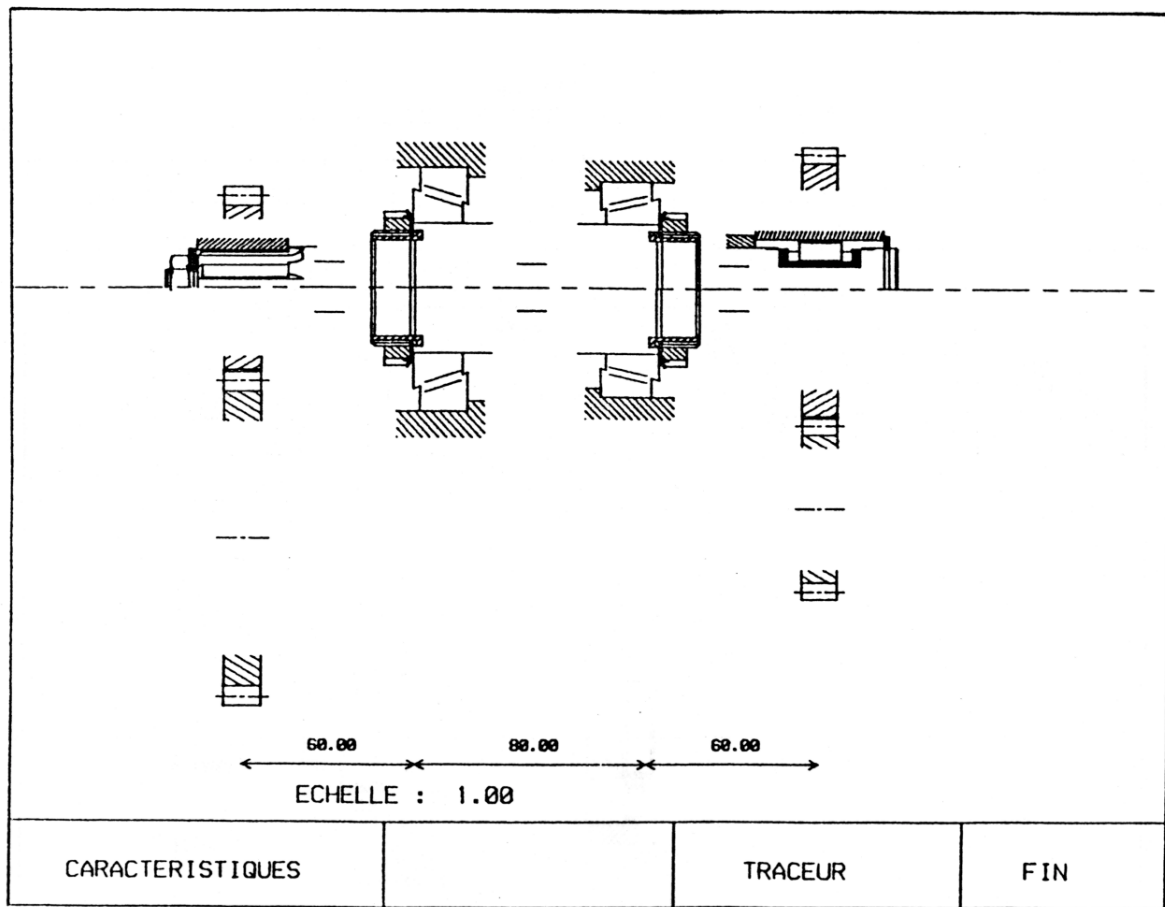


FIG. 1.19 – Copie d'écran de SICAM [Gui87].

bilités d'évolution du code informatique. On peut voir sur la figure 1.19 que les sorties écran et traceur que fournissait SICAM étaient d'une grande utilité aux concepteurs malgré une interface qui peut paraître désormais désuète.

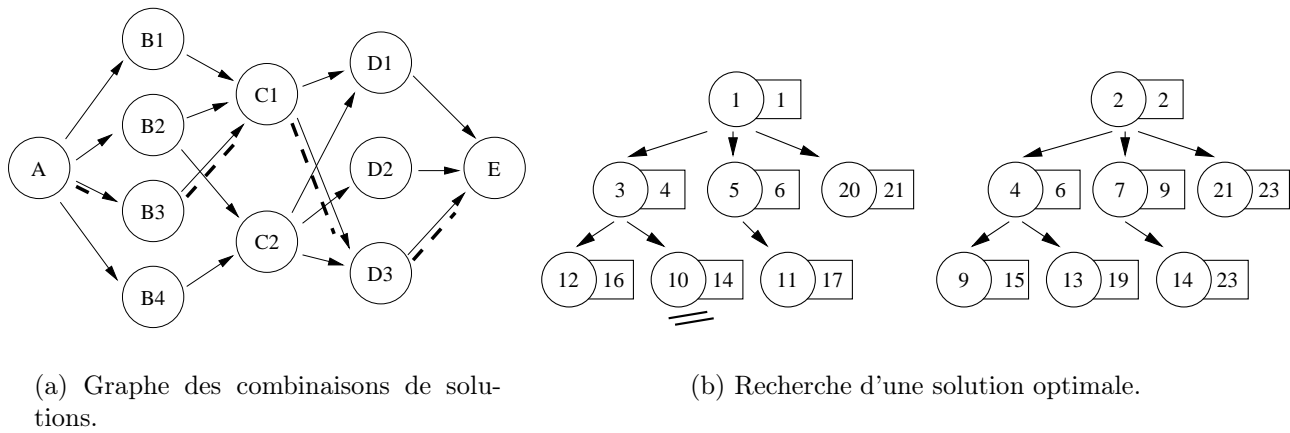


FIG. 1.20 – Recherche d'une solution optimale dans SICAM.

Il a bien été tenté une approche automatique du problème en y intégrant une logique d'optimisation [BD85]. Le principe utilisé était relativement simple puisqu'il proposait de parcourir un arbre de combinaisons formé par les ensembles de solutions de chaque liaison pour en dégager une solution globale optimale (figure 1.20.a). Le nombre de combinaisons potentielles devenant vite rédhibitoire, l'emploi d'une méthodologie efficace de recherche de solution optimale s'imposait. Les auteurs ont retenu un algorithme de séparation et évaluation progressive représenté sur la figure 1.20.b. La progression dans l'arborescence se fait grâce à l'évaluation, étape par étape, du critère d'optimisation retenu (en l'occurrence un critère de masse minimale). Dans le cas proposé, la solution optimale est formée de la combinaison 1–3–10. Cette approche, qui n'a pas été retenue dans SICAM, limite certainement le nombre d'évaluation et assure de trouver la solution optimale, mais sa rentabilité reste fortement tributaire du nombre de combinaisons à tester.

1.7.3.2 Les systèmes architecturés autour de MÉCAEXPERT.

Les systèmes experts constituent une famille de logiciels de l'I.A. Ils ont pour avantage, une aptitude à formaliser, stocker et utiliser des connaissances expertes, sous forme de règles, que le concepteur enregistre et peut mettre à jour régulièrement (système évolutif). Leur application à la conception mécanique a engendré de nombreuses études dont l'un des objectifs était de pallier aux insuffisances des systèmes algorithmiques.

Le système proposé par D. Benchaouine est basé sur un moteur d'inférences nommé MÉCAEXPERT [Ben93], programmé en langage LE-LISP. Les connaissances représentées dans le système sont les règles de production, les faits initiaux, les modules de calculs, et la base de données. Le système est appliqué au choix des liaisons mécaniques, illustré par une application au choix de liaisons par roulement. Les règles de conception mises en place dans ce système sont « classiques » et correspondent aux intentions que pourrait avoir un concepteur face à de tels problèmes (choix des roulements, des arrêts, de l'étanchéité...). L'auteur constate une difficulté certaine à capitaliser les connaissances. De plus, un ensemble de règles implicites, souvent relatives à des points de détail, remettent en cause la cohérence globale de la solution proposée. Le

système n'évite pas au concepteur le choix d'éléments technologiques *a priori*. On peut regretter que la procédure de choix soit figée selon un algorithme, au demeurant logique.

Dans le cadre d'une conception plus ambitieuse, c'est-à-dire un système de taille plus importante, on peut se questionner sur l'aptitude d'un tel système à construire et exploiter une base de connaissances. C'est une des problématiques soulevées par les travaux de K. Mehdi qui reprend le système MÉCAEXPERT afin de proposer un système intégré de conception de boîtes de transmission par engrenages [Meh95]. Après une analyse détaillée de la méthode de conception des boîtes de vitesses, et une classification de celles-ci, K. Mehdi propose une formalisation objet des différents composants standards d'une boîte de vitesses. Ces objets sont classés par famille selon trois types d'attributs : géométriques, fonctionnels et relationnels. Ceci oblige l'auteur à recréer des liens entre objets qui devraient être implicites, comme par exemple une condition d'engrènement entre dentures. Cette classification par famille est discutable car elle entraîne une surcharge relationnelle du modèle, ce qui a poussé l'auteur à étudier une version simplifiée du système. Le système proposé ne traite que du montage des roulements, dentures et éléments de liaison sur des arbres, mais il laisse apparaître une complexité liée vraisemblablement au degré de définition des composants mécaniques représentés (clavettes, roulements, épaulement...).

1.7.3.3 Les systèmes architecturés autour de SMECI.

Le laboratoire de Génie Mécanique de Toulouse a également développé cet axe de recherche. A la suite de SICAM, il a été décidé d'adapter des méthodes d'intelligence artificielle au principe méthodologique qu'avait exposé J. Guillot. Plusieurs systèmes intelligents d'aide à la conception ont ainsi été réalisés en s'appuyant sur le moteur d'inférences SMECI[®] commercialisé par la société ILOG[®]. Y. Silberberg en propose un, en 1994, appliqué aux arbres de transmission [Sil94]. L'interface graphique est assurée par les générateurs MASAI[®] et AÏDA[®]. L'auteur constitue une base de données à partir d'éléments constitutifs d'arbres de transmissions, à savoir des dentures d'engrenages, des roulements, des liaisons complètes arbre/moyeu et des dispositifs d'étanchéité et intègre des règles expertes (montage, immobilisation des roulements, lubrification, etc.). Une fois le cahier des charges renseigné, le moteur d'inférence génère un arbre d'états à partir d'un état initial, en parcourant le graphe des tâches et en exécutant les règles activables pour l'état courant. Une procédure de choix est alors engagée pour sélectionner le prochain état courant qui servira de base à la poursuite du raisonnement.

Malgré la souplesse introduite par l'utilisation de l'I.A. par rapport à la programmation algorithmique, notamment en matière d'évolution du système par intégration de nouveaux modules, la phase d'alimentation du système expert reste fastidieuse et délicate. Le système autorise l'étude en parallèle de plusieurs alternatives de conception et conserve un historique du raisonnement à des fins de compréhension ou de réutilisation. L'ampleur de l'entreprise a cependant pesé sur les résultats obtenus : la multiplicité des règles liées à l'assemblage des composants fonctionnels rend leur recensement difficile. La possibilité de laisser des champs du cahier des charges non remplis paraît séduisante, mais elle augmente l'opacité du raisonnement. Enfin, la liberté de positionnement donnée aux composants fonctionnels engendre des processus itératifs qui peuvent amener à un bouclage du système (d'où la présence de mécanismes d'arrêt).

Devant la complexité du système obtenu, des études ont été menées sur la spécialisation d'un tel système aux seules liaisons pivot par roulement [Dam96] et à la détermination des engrenages cylindriques [Dai96]. Dans le premiers cas, elles ont donné naissance au système SCLIP (Système de Conception de Liaisons Pivot par roulements). Conçu comme un sous-système expert de conception de mécanismes, SCLIP s'appuie aussi sur SMECI et emprunte l'interface graphique d'I-DEAS[®]. Le système utilise des *composants fonctionnels complexes* bâtis à partir de *composants*

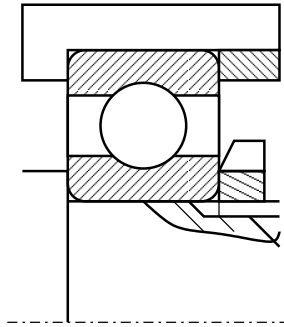


FIG. 1.21 – Exemple de composant fonctionnel complexe selon [Dam96].

fonctionnels simples. Dans l'exemple de la figure 1.21, le composant fonctionnel complexe « liaison rotule » est réalisé grâce aux composants fonctionnels simples roulement, élément de carter, entretoise... Ceci est justifié par le fait que seuls les composants fonctionnels simples possèdent un composant physique associé. Ce niveau supplémentaire de découpage autorise la modification du composant fonctionnel complexe par échange de ses composants fonctionnels simples. Ceci est réalisé par le système, en accord avec les règles activées lors du raisonnement. L'accent a été mis sur l'interactivité entre le concepteur et le système. L'intervention éventuelle du concepteur lors du raisonnement constitue une avancée significative dans la volonté de clarification de l'*intention de conception*. SCLIP se caractérise par une meilleure gestion des interférences géométriques entre les composants, tant radiales (pour le sens de montage des composants sur l'arbre) qu'axiales.

1.7.3.4 Autres systèmes experts.

Le système expert STATEX proposé par Aberšek et al. optimise le dimensionnement de paires d'engrenages par la méthode des algorithmes génétiques et la fabrication de ceux-ci [AFB96]. L'objectif retenu est la minimisation de la somme des volumes formés par les cylindres primitifs. Le système expert est formé de plusieurs modules, dont un modéleur 3D, un programme d'analyse des sollicitations et une base de données. Les résultats sont transmis à un pot-processeur puis à une chaîne CAM pour réalisation sur cellule flexible. Ce système, ambitieux car il couvre toutes les étapes de la conception et de la fabrication, ne traite cependant la problématique d'un réducteur que par la succession de traitements parallèles de trains d'engrenages. De plus, les règles sont principalement utilisées dans la partie fabrication et non la partie dimensionnement des trains d'engrenages.

1.7.3.5 Conclusion sur l'utilisation de l'I.A.

Les systèmes présentés permettent d'approcher, de façon qualitative, une solution réalisable mais on reste cependant loin de toute considération optimaliste. La conduite en parallèle de la détermination individuelle des différents composants fonctionnels conduit à une surenchère de règles visant à contrôler les interférences (comme par exemple le diamètre de bague intérieure de roulement et celui de son tourillon d'arbre). L'éclatement des composants fonctionnels complexes, dans le cas de SCLIP, oblige l'auteur à gérer des relations d'appartenance, de voisinage qui deviendront vite pénalisantes dans l'étude de mécanismes plus élaborés.

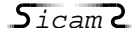
Les leçons que nous pouvons tirer de ces études sont les suivantes :

- les systèmes experts, outils de l'I.A., se distinguent des traitements algorithmiques par une grande souplesse ;

- la mise au point de tels systèmes est longue et nécessite un important travail de recensement des règles de conception associé à une modélisation rigoureuse des relations entre composants du système ;
- un système expert ne peut, à lui seul, assurer la conception intégrale d'un mécanisme. On doit lui associer des modules de calcul qui autoriseront une analyse qualitative et quantitative des solutions proposées.

Nous pensons donc que ce genre de système se prêteraient plus particulièrement à l'étude des mécanismes lors des étapes de synthèse de haut niveau, ou bien lors des phases de synthèse dimensionnelle de «solutions technologiques» sur des composants fonctionnels précis.

1.7.4 Les objectifs du nouveau SICAM.

Les travaux que nous présentons ici ont pour thème le développement d'une nouvelle version du logiciel SICAM que nous avons baptisé SICAM2 (Nous utiliserons le logo  par la suite). Son domaine d'intervention est celui de la tâche T4 de notre organisation opérationnelle (chapitre 1.6.2.4). Son architecture doit être ouverte afin d'autoriser l'extension du logiciel à d'autres composants fonctionnels que ceux déjà proposés. Cela se traduira par une volonté de rendre indépendantes les diverses parties du logiciel. Le logiciel se veut être un système *interactif* et d'*assistance*, ce qui implique une grande convivialité dans l'interface homme/machine et une grande souplesse dans le déroulement des opérations afin de minimiser les contraintes sur le concepteur dans la chronologie de ces opérations. Comme le suggèrent Waldron et Waldron, un outil de CAO intelligent devrait contenir des routines de calculs pour les composants familiers au concepteur [WW96]. Cette réflexion peut même être poussée plus loin en introduisant une logique d'optimisation appliquée à ces routines de calcul. Notre outil devra proposer des solutions optimisées pour chaque DTLM étudié. Nous préconisons également d'associer aux DTLM des modèles 3D ainsi que des modules d'aide à la sélection. La notion de bibliothèque de solutions technologiques pour une fonction donnée est séduisante. Elle a été retenue et mise en application pour les DTLM de quelques liaisons. Nous discutons dans le chapitre suivant de ces éléments et de leur exploitation dans notre système.

Cette nouvelle version est aussi l'occasion d'exploiter les progrès de l'informatique réalisés depuis les années quatre-vingts, notamment en matière de modeleur volumique. Il ne s'agit pas simplement d'ajouter une troisième dimension à la représentation du mécanisme, mais plus généralement de mener une réflexion sur les techniques mises en œuvre par le passé, dans un contexte donné, et ce qu'elles sont aujourd'hui. Nous désirons exploiter toutes les possibilités offertes par l'informatique actuelle. En premier lieu nous utilisons un outil de modélisation géométrique issu du domaine commercial : I-DEAS[®] de SDRC. Les intérêts sont :

- une mise à profit immédiate de la puissance d'un modeleur 3D ainsi que des méthodes de gestion de bibliothèques de modèles ;
- la production d'un modèle de mécanisme exploitable dans l'environnement du logiciel, et plus généralement dans tout environnement CAO (via des filtres d'exportation par exemple) ;
- la démonstration des capacités d'ouverture des logiciels de CAO sur les métiers de la conception, y compris en avant-projet, et son corollaire, i.e. l'intégration de modules spécifiques dans un système CAO ;

- la conformation d'un système de CAO à une méthodologie de travail telle que nous l'avons définie.

En second lieu, il nous paraît indispensable de proposer une interface homme/machine qui bénéficie des nouvelles représentations graphiques telles que le multi-fenêtrage, les boutons, etc. Par le biais de menus, de fenêtres de dialogues, d'interactivité dans le dialogue, le nouveau système doit être plus adapté aux méthodes de travail du concepteur. Cet aspect ergonomique du logiciel sera traité dans le chapitre 6. Enfin c'est aussi l'occasion de nous interroger sur la construction d'un logiciel de conception intégré, le choix d'architecture et de langages de programmation.

Cette nouvelle version de SICAM est une première contribution à l'élaboration d'un système intégré de conception. Dans le cadre d'une conception préliminaire, ce système doit, à terme, proposer une large bibliothèque de DTLM pour lesquels des méthodes d'optimisation connues et éprouvées seront disponibles. La principale difficulté réside dans le maintien de la cohérence du produit, notamment dans la gestion des interférences spatiales entre les différents DTLM. L'attrait d'un tel système est multiple. :

- il doit faciliter la conception nouvelle de mécanismes générateurs de fonction en proposant au concepteur de *construire* sa solution en assemblant des briques fonctionnelles. Le concepteur est déchargé des calculs puisque ceux-ci sont automatiquement gérés par le logiciel en fonction des DTLM choisis. Il peut alors se concentrer sur les choix d'architectures du mécanisme et doit pouvoir, à volonté, tester l'influence du choix des DTLM sur l'aspect global du mécanisme. Le système doit garantir la validité totale de la solution.
- Lors d'une modification d'architecture existante (reconception), le concepteur doit pouvoir faire évoluer progressivement sa solution en intégrant de nouveaux DTLM en lieu et place des anciens. Il mesure ainsi concrètement l'impact de ces modifications sur le mécanisme.

En définitive, *Sicam 2* doit faciliter le travail du concepteur dans la tâche qui semble actuellement la plus sensible, c'est-à-dire la concrétisation des liaisons mécaniques par des dispositifs technologiques. Ce passage du concept à la matière entraîne bien souvent une remise en cause des choix antérieurs. Le découpage en sous-problèmes ainsi que l'optimisation doivent pouvoir améliorer la recherche d'une solution cohérente, à condition que ce découpage soit judicieusement réalisé.

Chapitre 2

Concrétisation des liaisons mécaniques : les DTLM.

Résumé

LES DTLM sont les éléments de base de notre système *Sicam2*. Ils concrétisent les liaisons mécaniques mises en place lors des phases amont du processus de synthèse. Nous détaillons la composition d'un DTLM et illustrons notre propos par quelques exemples. Nous étudions ensuite ces éléments selon un point de vue fonctionnel en détaillant leur rôle dans le mécanisme et la structure objet associée à notre base de DTLM. Nous poursuivons sur la méthodologie de mise en œuvre de ces DTLM en posant les problématiques du paramétrage et du dimensionnement optimal (illustrées par un exemple). Nous concluons ce chapitre sur quelques remarques, après communication de la liste des DTLM disponibles à ce jour dans le système *Sicam2*.

Contenu du chapitre

2.1	Composition d'un DTLM.	37
2.1.1	Exemple de composition d'un DTLM.	37
2.1.2	Importance du paramétrage des DTLM.	38
2.1.3	Le DTLM : feature de haut niveau.	39
2.1.4	Intégration du DTLM dans un environnement CAO.	42
2.1.4.1	Définition des points caractéristiques du DTLM.	43
2.1.4.2	Positionnement du modèle 3D.	46
2.1.4.3	Exploitation du modèle 3D.	47
2.2	Aspect fonctionnel du DTLM.	47
2.2.1	Approche orientée objet.	48
2.2.2	Dimensionnement et optimisation d'un DTLM.	54
2.3	Création des DTLM.	58

2.3.1	Exemple de modélisation d'une famille de DTLM : la liaison complète par clavette.	58
2.3.1.1	Présentation du problème.	58
2.3.1.2	Formulation du problème de dimensionnement optimal de la liaison.	59
2.3.1.3	Résolution du problème par la méthode de Johnson.	70
2.3.1.4	Recherche par énumération.	74
2.3.2	Choix des méthodes d'optimisation.	75
2.3.3	Ensemble des DTLM développés.	77
2.4	Conclusions.	79

2.1 Composition d'un DTLM.

A l'issue du chapitre précédent, nous connaissons le niveau d'activité du logiciel *Sicam2* dans notre découpage opérationnel de la phase de synthèse. Il s'agit de la tâche T4 (sec. 1.6.2.4) durant laquelle le concepteur choisit et pré-dimensionne les solutions technologiques afin de concrétiser les liaisons qui ont été définies durant les tâches précédentes. Pour cela, il doit avoir à sa disposition les entités fonctionnelles qu'il pourra associer à chacune de ces liaisons. Les « *dispositifs technologiques assurant une liaison mécanique* » (ou DTLM) sont une extension de ce que J. Guillot a nommé « *solutions technologiques* » et dont il donne la définition suivante [Gui87] : « la réalisation technologique d'une liaison est constituée de l'ensemble des parties fonctionnelles (surfaces de contact, volumes de matière) appartenant aux différentes pièces et nécessaires pour assurer la transmission des efforts entre deux pièces principales ». Le concept de DTLM va plus loin dans la définition du composant. En intégrant le terme de « liaison mécanique » dans son nom, le DTLM précise exactement le type de fonction qu'il remplit. Les DTLM constituent des objets à part entière auxquels nous appliquons une méthodologie de traitement orientée objet (classe, méthode. . .). Cela se traduit par une structure de données ainsi qu'une structure logicielle fondamentalement différentes de ce que présentait SICAM. Les liaisons les plus courantes se répartissent suivant trois types :

- Les liaisons mécaniques de base. Elles sont définies par la théorie des mécanismes et constituent la chaîne cinématique du mécanisme.
- Les liaisons complètes (démontables ou non). Celles-ci ne participent pas à la cinématique du mécanisme puisqu'elles n'introduisent pas de degré de liberté. Elles sont destinées à marquer des discontinuités dans les matériaux et les formes des pièces, pour de multiples raisons : montage, changement de matériaux, adaptation d'éléments standards. . .
- Les liaisons d'étanchéité. Considérées comme surabondantes, du point de vue de l'étude strictement mécanique du système, ces liaisons ne transmettent pas de puissance et servent à délimiter des espaces pour lesquels les règles environnementales diffèrent (lubrification, atmosphère, isolation phonique. . .). Ces liaisons ne sont pas (encore) représentées dans la nouvelle version de SICAM.

Le DTLM est la matière d'œuvre principale de la tâche T4. Il participe à la matérialisation des concepts introduits précédemment en représentant les entités physiques impliquées dans le mécanisme. A ce titre, on doit le considérer comme le lien entre le domaine fonctionnel (où se situent les objectifs de la conception) et le domaine physique (où se situent les solutions) [LO98]. Ces deux mondes sont fondamentalement dissociés, mais le propre de la conception est de créer des liens entre eux. C'est une tâche qui peut être assimilée à l'« *embodiment design* » de [PB96]. La représentation du DTLM nous apporte de précieuses informations sur sa réalisation et sa fonction. On ne peut pas considérer le DTLM comme un composant mécanique traditionnel puisqu'il est constitué de plusieurs pièces et volumes de matière dont la plupart ne seront finalisés que lors de la tâche suivante du processus de synthèse. Nous devons plutôt le voir comme un sous-ensemble du mécanisme qui, à lui seul, assure une fonction de liaison mécanique.

2.1.1 Exemple de composition d'un DTLM.

Traisons le cas d'une liaison rotule comme exemple. L'élément généralement considéré comme l'élément central du DTLM est l'élément roulant (ou glissant) qui autorise les trois rotations. Si

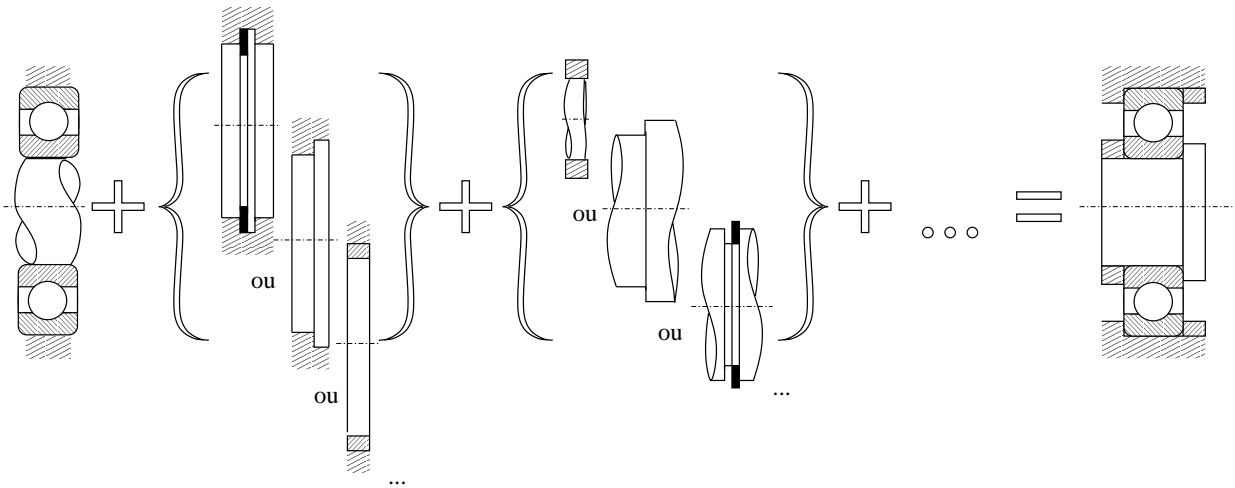


FIG. 2.1 – Assemblage d'un DTLM.

le concepteur opte pour un roulement à billes, nous y adjoignons un tronçon d'arbre qui lui sert de support ainsi qu'une partie de moyeu. Par la suite nous devons y ajouter des éléments visant à supprimer la translation axiale restante. Plusieurs dispositifs sont envisageables : anneau élastique, entretoise, épaulement... A ce stade de la définition du DTLM, le concepteur ne dispose pas des dimensions du roulement. Il existe néanmoins des relations implicites entre les différents éléments du DTLM qui vont nous permettre de le créer. Il s'agit, entre autres, des relations liant les diamètres. Prenons l'exemple de la figure 2.1. Diverses solutions, symbolisées par les ensembles d'éléments entre accolades, sont à la disposition du concepteur pour matérialiser les arrêts axiaux sur la bague intérieure. S'il opte pour un épaulement, on sait alors que la face d'appui de l'épaulement doit venir en contact avec la face latérale de la bague intérieure, que le petit diamètre de l'épaulement doit être identique au diamètre du tronçon d'arbre supportant le roulement. Il subsiste alors des éléments non déterminés tels que la largeur d'épaulement ou le diamètre de l'épaulement. Ce dernier est fortement lié aux dimensions de la bague intérieure du roulement. Dès que le roulement aura été choisi, ce diamètre pourra être fixé conformément aux recommandations du constructeur du roulement. Quand à la largeur de l'épaulement, elle sera déterminée par un calcul de résistance dès que les dimensions du roulement seront connues.

2.1.2 Importance du paramétrage des DTLM.

Si nous envisageons maintenant le remplacement de l'épaulement par un écrou, nous devons mettre en place une liaison fileté sur l'arbre dont les filets doivent obligatoirement dépasser l'aplomb de la face latérale du roulement afin d'assurer le contact entre la rondelle de l'écrou et le roulement. Il est important de noter que l'interface entre le tronçon d'arbre du roulement et le tronçon d'arbre de l'arrêt axial est dépendant du type d'arrêt choisi. Il est donc difficile de prévoir une frontière commune avant un quelconque dimensionnement, et dans l'optique d'un stockage de DTLM «en kit», on devrait augmenter le nombre de variables de paramétrage de la géométrie. La figure 2.2 visualise ce que devrait être le paramétrage d'un DTLM à base de features simples. Comme nous le voyons, bon nombre de règles sont implicites dans la construction du DTLM. A l'inverse de pratiques visant à extraire du composant mécanique un maximum de features géométriques simples [Qam98], nous considérons que l'intégration de tous ces éléments dans le DTLM le *soulage d'un grand nombre de règles implicites* de construction.

L'introduction dans le DTLM d'éléments de géométrie qui ne forment pas une pièce entière

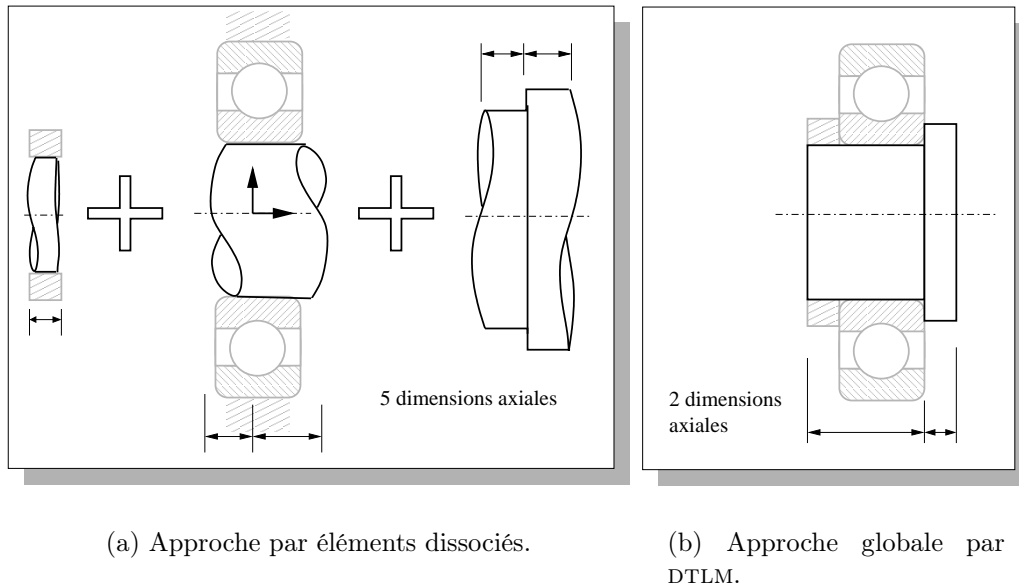


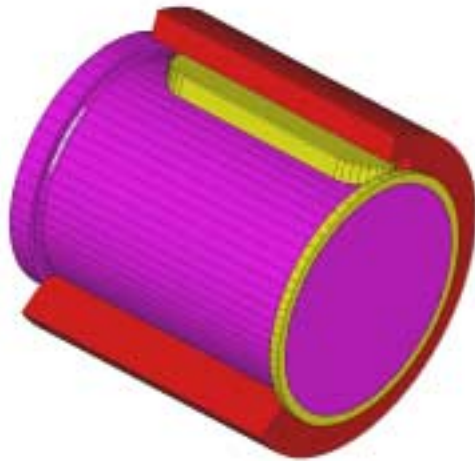
FIG. 2.2 – Différences de paramétrage.

mais une partie de celle-ci (arbre, moyeu, entretoise) peut dérouter le concepteur. Il est alors important de faciliter une reconnaissance de ces groupements de matière. Une des solutions que nous avons mise en place est un codage par la couleur. Ainsi, tous les tronçons d'arbre sont de couleur identique, ce qui permet aisément de les distinguer en une seule et même entité, en l'occurrence l'«arbre». Cette méthode peut cependant présenter un défaut : le concepteur ne doit pas confondre l'entité fonctionnelle «arbre» avec la pièce «arbre». Le concept de pièce impose une unité de matière, ce qui est une vision restrictive et pénalisante de l'entité. L'exemple du carter est éloquent. Imaginons que les règles de montage, issues de la structuration du carter de la tâche T3 (sec. 1.6.2.3), impliquent un carter en deux parties dont la séparation doit être située quelque part entre les deux paliers. Les moyeux seront alors dissociés physiquement, puisqu'appartenant à deux pièces différentes, mais liés fonctionnellement en une entité «carter» dont l'une des fonctions sera de fournir un support commun aux paliers. Cette division des pièces mécaniques est discutée dans [PB96]. Les DTLM fournissent donc une plus grande liberté de création au concepteur puisque ce dernier n'a pas à se soucier des formes qui n'interviennent pas directement dans la tâche dont il est responsable. La figure 2.3 donne un aperçu de quelques DTLM réalisant diverses liaisons (complètes, rotules et linéaires annulaires). On remarquera particulièrement les éléments (a) et (b) qui mettent en œuvre différentes technologies pour réaliser une fonction de liaison identique.

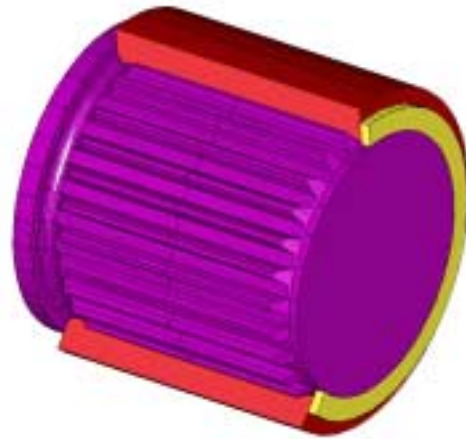
2.1.3 Le DTLM : feature de haut niveau.

Les logiciels de CAO proposent de nouveaux éléments de construction, les features, élaborés à partir de surfaces ou volumes géométriques, que le concepteur utilise dans la construction de son mécanisme. Bien souvent, ces features, dont il n'existe pas de définition universelle, ne sont utilisés que dans la création des pièces du mécanisme. L'origine historique de ces features est le monde de la fabrication industrielle. Ce sont alors des features d'usinage¹. Les travaux concernant les features sont une des clés de la recherche en CAO des années à venir. Deux grands

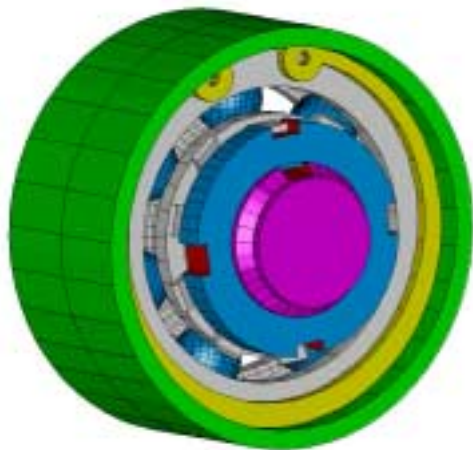
1. Ou "machining features".



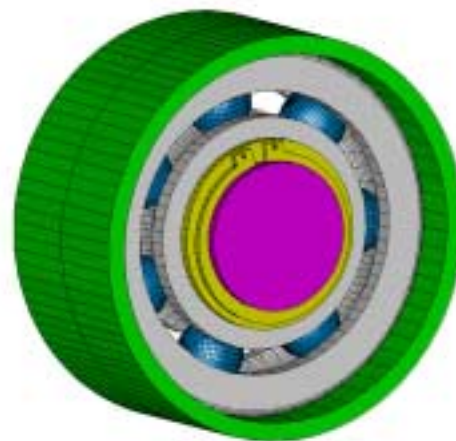
(a) Liaison complète par clavette.



(b) Liaison complète par cannelures.



(c) Liaison rotule par roulement à billes radial.



(d) Liaison linéaire annulaire par roulement à billes radial.

FIG. 2.3 – Exemples de DTLM réalisant différents types de liaisons.

axes se développent actuellement [GR01], complémentaires, qui sont la modélisation à base de features et la reconnaissance de features². Les features pourraient être amenés à sortir de ce strict cadre de la FAO et devenir un élément incontournable de la conception.

Les features sont généralement constitués de couches d'informations qui se superposent à une représentation solide 3D (BRep, CSG,...). Cet ajout de *sens* à l'objet permet de le manipuler dans un contexte de conception. Il est fondamental pour le concepteur que le feature symbolise, sans aucune ambiguïté, le rôle pour lequel il a été choisi. Il peut cependant entraîner des problèmes d'intégrité lorsque, par combinaison de features, les géométries et les intentions fonctionnelles interfèrent entre elles [CH00]. Les auteurs reprennent et développent le concept d'intentions du concepteur orienté features³ en trois points : intentions fonctionnelles théoriques, relationnelles et morphologiques. Les intentions fonctionnelles théoriques matérialisent les comportements mécaniques, les flux d'énergies, et plus généralement tous les concepts physiques susceptibles d'être mis en œuvre dans le système. Elles se traduisent par des formes de pièces en relations étroites avec le comportement attendu. Les intentions fonctionnelles relationnelles traduisent les relations géométriques et spatiales entre les différents features. Il peut s'agir de positionnements relatifs, d'orientations qui sont, ou non, dépendants des applications. Enfin les intentions morphologiques différencient les membres au sein d'une même famille de features. Toutes ces informations témoignent de la multitude de données que peut contenir un feature. Il en vient même des problèmes d'interprétation voire de reconnaissance de features par les corps de métiers dans le cadre d'ingénierie concourante [JG00]. Le fabricant ne reconnaît pas les mêmes features que le concepteur parce que l'utilisation qu'il en fera est différente. L'un des objectifs visés dans [Tse99] est justement, par une approche modulaire, de pouvoir simplifier les mécanismes de reconnaissance de features simples (de fabrication) dans les features complexes (de conception). Ces travaux partent du constat que le travail du concepteur est facilité s'il utilise des features de haut niveau sans se soucier de leur contenu. Le lien entre les concepteurs et les fabricants n'est pas encore parfait, mais les features peuvent constituer un élément de réponse à ce problème dual.

Contrairement à [Ama98], nous pensons que le feature est apte à colporter des *intentions de conception*, et cela indépendamment du degré de définition géométrique. Nous nous démarquons du contexte actuel qui restreint la conception par features à la seule élaboration de pièces, et nous pensons que les DTLM sont des features de *haut niveau* qui sont associés à la fonction «liaison mécanique». Le terme «haut niveau» mérite une explication : il s'agit en fait du niveau d'intégration du feature. Il regroupe toutes les connaissances évoluées sous-tendues par le concept de liaison mécanique, et une géométrie complexe puisque basée sur un assemblage de volumes de matière (il est délicat de mentionner ici le mot *pièces* puisque certaines parties ne sont que des fragments de pièces).

Ce type de feature se démarque des features « traditionnels » par sa complexité. L'amélioration des conceptions basées-feature est concernée au premier chef. Ce problème soulevé par de nombreux travaux dont [CH95] se traduit par une modification de la topologie des pièces à la suite des modifications de features. Ces inconvénients sont principalement dus au manque d'explicitation des intentions du concepteur lors de l'introduction de features de bas niveau. Ils sont exacerbés par la fonctionnalité plurivalente que peuvent présenter des éléments géométriques trop simples. Les risques sont limités lorsqu'on utilise des features de plus haut niveau, car présentant moins d'ambiguïté dans l'interprétation.

2. Respectivement “feature-based design” et “feature recognition”.

3. Traduction (heureuse?) de “Feature-based designer’s intents”, soit l'intention du concepteur manipulant des features.

L'utilisation de features simples peut servir de base à des assemblages plus complexes. C'est alors un moyen de diminuer le nombre de features disponibles pour le concepteur et de ce fait, de simplifier la gestion des bases de données. Mais une telle affirmation peut sembler optimiste, compte tenu du nombre de combinaisons que représente l'assemblage des nombreuses formes de base. De plus, on constate régulièrement que de nombreuses combinaisons sont irréalisables, ou ne présentent pas d'intérêt particulier, ce qui diminue sensiblement le nombre de combinaisons nécessaires. Des outils permettant de concevoir des features de haut niveau sont envisageables, mais ils sont pour l'instant hors de notre propos.

Les DTLM que nous présentons doivent être considérés comme des features-utilisateur⁴. C. Hoffmann reconnaît aux features-utilisateur des caractéristiques intéressantes [HJ98]. L'agrégation, en composants complexes, de features simples augmente l'indépendance fonctionnelle de ces features entre-eux. Malheureusement, les logiciels commerciaux ne proposent pas encore de mécanismes sophistiqués de construction et de manipulation de features-utilisateur. Ils sont généralement basés sur une conception paramétrée des géométries qui, par le biais de mécanisme d'historique de construction, permettent d'instancier des variantes de classes prédéfinies. La méthode est fragile et conduit parfois à des aberrations.

La vision qu'a le concepteur des DTLM est une vision de spécialiste. Il emploie ces features pour leur fonction de liaison. Il ne se préoccupe pas des features de plus bas niveau compris dans le DTLM formé par les surfaces fonctionnelles. L'emploi de ces surfaces fonctionnelles en CAO a été traité dans de nombreux travaux et a souvent mené à une complexité importante, notamment dans sa représentation (souvent à base de graphes) et dans sa prise en compte pour la réalisation des liaisons [Con96]. Cette aptitude à masquer une partie de l'information est donc intéressante : elle simplifie la vision instantanée du DTLM tout en autorisant une lecture pluridisciplinaire (les features d'usinage sont bien présents dans le DTLM!). En situant le DTLM à un niveau supérieur de compréhension et de définition, nous proposons un ensemble cohérent d'éléments mécaniques qui satisfait entièrement à la fonction de liaison assignée. Sa représentation graphique fait apparaître des composants mécaniques ou des formes qui suggèrent au concepteur la fonction qu'il remplit. Le DTLM intègre des composants standards tels que des roulements, des clavettes, ou bien des formes paramétrées, souvent normalisées, comme des cannelures, des lamages, etc. qui sont autant d'indices d'identification pour le spécialiste.

2.1.4 Intégration du DTLM dans un environnement CAO.

La visualisation du DTLM par le concepteur est une opération incontournable dans le processus de conception. C'est l'élément clé du raisonnement du concepteur puisqu'elle permet un retour d'information quasi instantané vers le concepteur. C'est seulement lorsque le DTLM, préalablement choisi et dimensionné, est intégré au mécanisme que le concepteur peut alors juger de la pertinence de son choix. L'introduction du DTLM dans son environnement autorise la comparaison directe avec les DTLM voisins et une appréciation globale du mécanisme. Le concepteur peut constater *de visu* les interférences axiales (le chevauchement) entre DTLM, dans le cas où ils n'auraient pas été suffisamment contraints, et juger de la réalisation des choix de montage. Il est difficile, pour les concepteurs peu expérimentés, de pressentir un sens de montage avant d'avoir des informations dimensionnelles sur les composants. Il est donc nécessaire de pouvoir vérifier, image à l'appui, que les ordres de grandeur des dimensions des divers DTLM valident les sens de montage que le concepteur souhaitait imposer. A l'inverse, si une solution technologique ne convient pas, le concepteur a tout loisir de changer de solutions technologiques avant

4. "User-defined features" est le terme anglo-saxon consacré.

de remettre en cause son ordre de montage, donc sa structure de mécanisme. A titre d'exemple, nous pourrions citer le cas où le concepteur, ayant prévu un montage de tous les composants sur l'arbre par la droite, constate que le DTLM central, réalisant une liaison pivot, est tant sollicité qu'il impose une dimension d'arbre rendant impossible le montage des composants à sa gauche. La gestion automatique de ces contraintes de montage est abordée au chapitre 5.

2.1.4.1 Définition des points caractéristiques du DTLM.

Durant son travail, le concepteur sélectionnera des DTLM qui viendront alors s'intégrer au mécanisme en cours d'élaboration. Dans le système *Sicam2*, l'instanciation d'un DTLM se traduira parallèlement par la mise en place de son modèle géométrique dans le modèle 3D global du mécanisme. Sans entrer dans le détail de la réalisation, ceci faisant l'objet du chapitre 6, nous pouvons succinctement présenter la structure de dialogue de *Sicam2*. Celle-ci se décompose en deux interfaces principales concernant le dialogue concepteur/logiciel : l'une est consacrée à la saisie des données du cahier des charges par des menus graphiques tandis que l'autre réalise la visualisation 3D du mécanisme en cours d'élaboration par un logiciel de CAO commercial. Elles sont représentées dans la figure 2.4 en configuration d'utilisation : l'interface graphique du logiciel de CAO, constituant le fond d'écran, est surmontée d'une fenêtre de saisie de données.

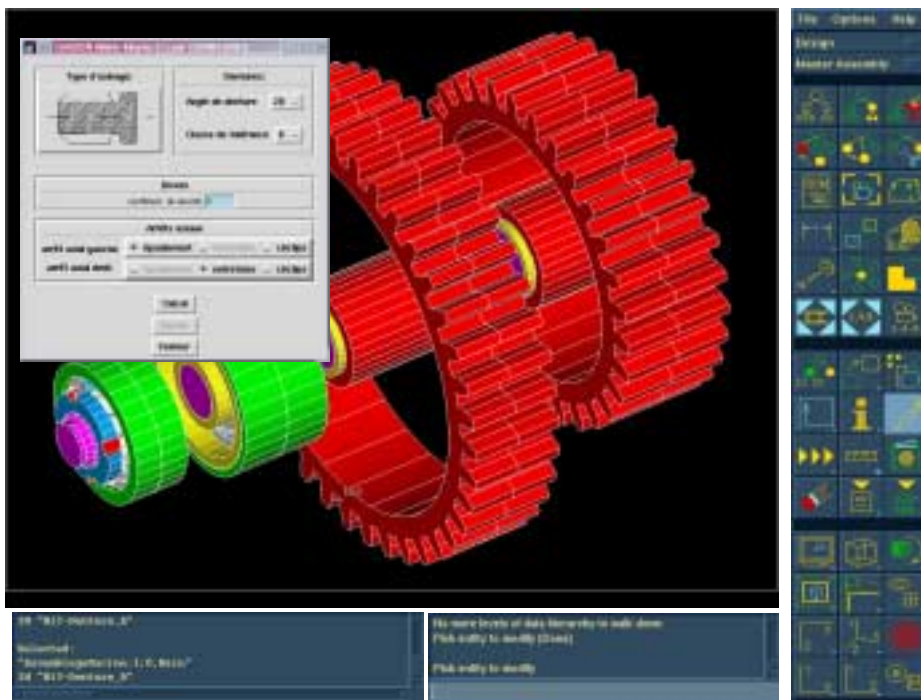


FIG. 2.4 – Interface graphique menus/CAO de *Sicam2*.

Définition du centre géométrique de la liaison.

Nous avons vu, au chapitre précédent, que le concepteur aura été amené à positionner les liaisons de son mécanisme dans les tâches antérieures à la tâche T4. Il aura notamment défini des points, que nous nommons « *centres géométriques des liaisons* », qui correspondent à des points de l'espace tridimensionnel autour desquels doivent se construire les liaisons. Ces centres géométriques constituent donc un des éléments clés du *cahier des charges global* de la tâche T4.

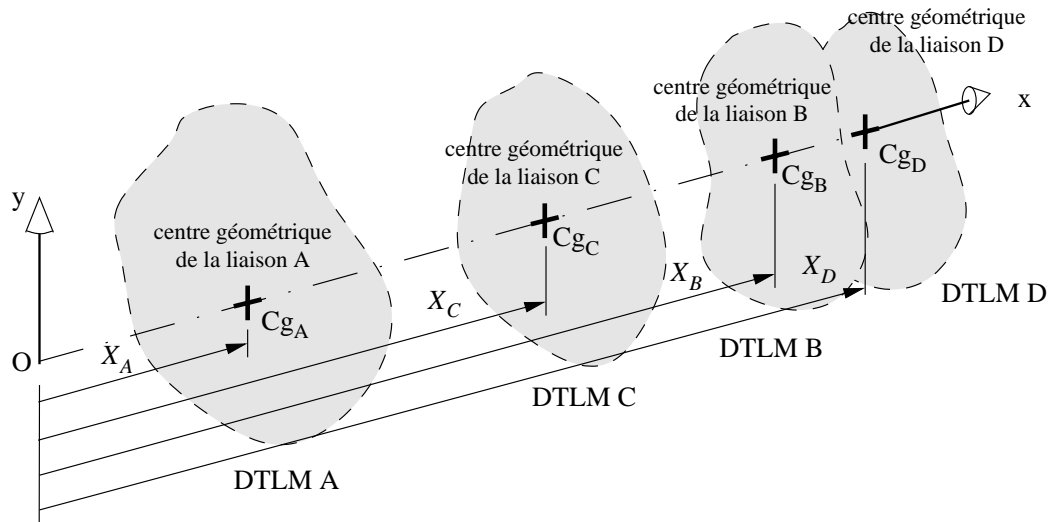


FIG. 2.5 – Placement des DTLM selon le cahier des charges principal.

C'est dans l'espace situé autour d'un centre géométrique que viendra prendre place le DTLM associé (cf. figure 2.5).

Définition du point de référence du modèle géométrique 3D.

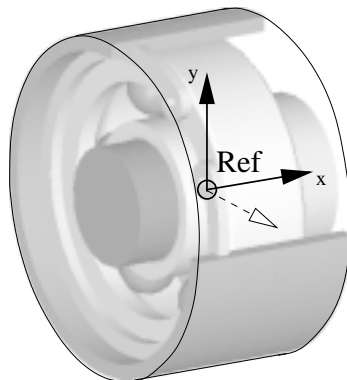


FIG. 2.6 – Point de référence et trièdre associé d'un modèle géométrique 3D.

Le DTLM est un ensemble de divers éléments mécaniques. Il est logique de le réaliser comme un assemblage, au sens du logiciel de CAO, i.e. comme un ensemble de volumes de matière liés entre eux par un réseau de contraintes géométriques et dimensionnelles. Afin de manipuler les modèles 3D, nous avons besoin d'un repère spatial propre au DTLM. Cette référence est concrétisée par un trièdre directeur (Ref, x , y , z) dont le centre est appelé point de référence du DTLM (Ref). Ce point pourra par la suite être affecté d'un indice du nom du DTLM auquel il est attaché : par exemple, Ref_A pour le point de référence du DTLM A. Ce trièdre servira aussi à orienter et positionner le modèle du DTLM lorsqu'il faudra placer le DTLM dans le modèle global 3D du mécanisme, nommé « super-assemblage » puisqu'il est un assemblage de plusieurs assemblages. Les DTLM considérés dans notre travail étant construits avec des pièces de révolution, seuls deux des trois axes seront effectivement exploités (cf. figure 2.6). De ce fait, nous pourrions adopter, pour la suite de notre exposé, une représentation bidimensionnelle conforme à la figure 2.7.

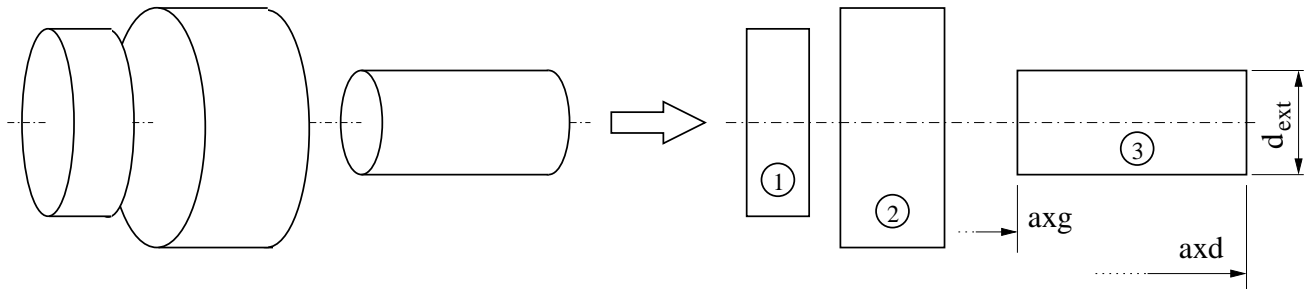


FIG. 2.7 – Représentation bidimensionnelle d'un problème tridimensionnel.

Nous considérons que le point de référence est placé au centre de l'élément *principal* (qui se trouve également être l'élément *central*). En effet, le centre de l'élément principal correspond généralement au milieu du DTLM complet. Dans le cas d'une liaison complète par clavette par exemple, ce point se trouvera sur l'axe de l'arbre, à hauteur du milieu de la clavette. Seuls les arrêts axiaux peuvent introduire une dissymétrie du DTLM. Nous avons cependant constaté qu'elle reste toujours faible devant les dimensions axiales du DTLM.

Définition du centre d'effort du DTLM.

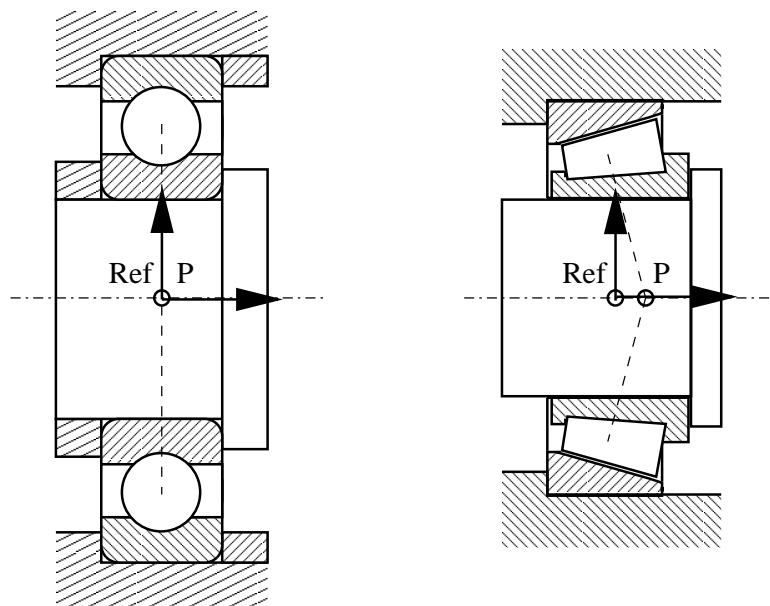


FIG. 2.8 – Point de référence et centre d'effort.

Les calculs de statique sont réalisés à partir d'éléments de réduction du torseur des efforts de la liaison. Ce torseur est attaché à un point précis que nous nommons *centre d'effort* et repérons par un point⁵ P. Pour la plupart des DTLM, nous ne rencontrons pas de difficultés en considérant que ce point est placé au centre de l'élément *principal*, et donc confondu avec le point de référence. Cette hypothèse est valable pour les éléments roulants à contact radial (billes, rouleaux cylindriques), mais les liaisons pivots réalisées par des roulements à contact oblique introduisent un décalage dû à l'excentration du foyer de poussée (caractérisé par l'angle α appelé *angle de contact*). Nous ne pouvons pas prévoir de valeur, même approximative, pour

5. Ce point est également appelé « centre de Poussée ».

cet angle puisqu'il dépend du roulement employé et varie dans de fortes proportions (de 10° à 30° [SKF01]). Ce point n'est donc connu qu'après détermination du type de roulement employé, ou plus généralement après dimensionnement du DTLM. La figure 2.8 illustre le décalage entre le point Ref —point de référence du DTLM— et le point P —centre d'effort du roulement— dans le cas d'une liaison par contact oblique.

2.1.4.2 Positionnement du modèle 3D.

L'intégration du modèle géométrique 3D du DTLM dans le super-assemblage pose le problème suivant. Lorsque le concepteur fixe la position des centres géométriques des liaisons aux niveaux supérieurs de définition, il n'a encore aucune connaissance des DTLM qu'il emploiera. Ces points sont les seuls éléments tangibles et disponibles pour un calcul de statique au début de la tâche T4. Ils servent donc de base au calcul des sollicitations, et donc de dimensionnement des DTLM.

Une fois les DTLM dimensionnés, il nous reste à intégrer leurs modèles géométriques dans le modèle 3D global. Deux solutions s'offrent à nous :

- la première consiste à faire coïncider le point de référence (Ref) du modèle géométrique avec le centre géométrique (Cg) de la liaison. On a ainsi l'assurance que le modèle géométrique du DTLM sera centré sur le centre géométrique de la liaison. On respecte donc l'intention du concepteur qui a exprimé la volonté de construire le DTLM autour de ce centre géométrique de liaison.
- La seconde solution consiste à faire coïncider le centre d'effort (P) du modèle géométrique avec le centre géométrique (Cg) de la liaison. Cette méthode présente l'avantage de respecter les hypothèses faites sur la transmission d'effort en s'assurant que le point utilisé pour les calculs de statique est réellement le centre d'effort.

Chaque méthode est cependant préjudiciable, dans une certaine mesure, à la qualité du travail de conception. La figure 2.9 représente les deux options que nous venons d'évoquer. Si nous positionnons le modèle 3D du DTLM en plaçant son point de référence (Ref) sur le centre géométrique de liaison (Cg), nous obtenons la position représentée en trait fort. Le modèle géométrique est bien centré (à la dissymétrie près) sur le centre géométrique de liaison, mais nous introduisons un décalage entre le point que nous *voudrions* utilisé pour les calculs de statique (Cg) et le point *réellement* concerné par la transmission d'effort (P). Si nous positionnons le modèle 3D du DTLM en plaçant son centre d'efforts (P) sur le centre géométrique de liaison (Cg) nous obtenons la position représentée en pointillés. Les calculs de statique sont alors valides, mais le positionnement du DTLM diffère manifestement de ce que pouvait espérer le concepteur. Nous assistons à un *décalage inopiné* du DTLM par rapport à la position spécifiée dans le cahier des charges global.

Nous avons opté pour le positionnement basé sur le point de référence, c'est-à-dire que le point de référence du DTLM (Ref) sera placé sur le centre géométrique de la liaison (Cg). Cette méthode de placement convient généralement bien pour les différents types de DTLM puisque seuls les DTLM impliquant des roulements à contact oblique introduisent un décalage significatif du modèle géométrique 3D. Dans ce dernier cas, nous serons contraint de *mener conjointement* le calcul de statique et la détermination des DTLM. Nous verrons plus loin que le choix de ce positionnement simplifie la tâche du concepteur dans la définition des volumes limites du DTLM.

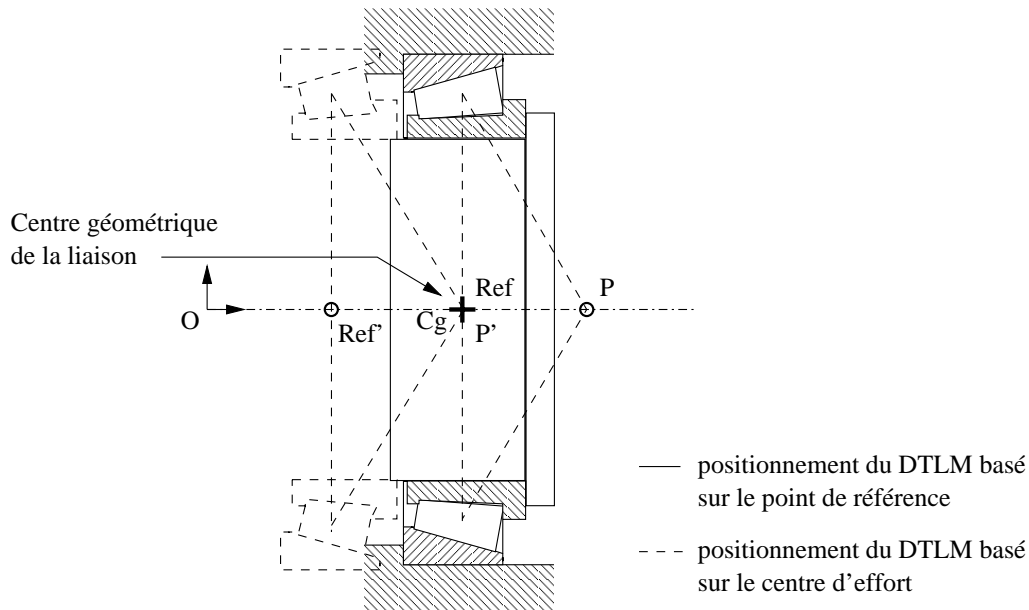


FIG. 2.9 – Positionnements possibles d'un DTLM par rapport au centre géométrique de la liaison.

2.1.4.3 Exploitation du modèle 3D.

Dès l'instant où tous les DTLM intégrés au mécanisme auront pris leurs caractéristiques définitives, le concepteur pourra procéder à un éclatement des assemblages de CAO afin de récupérer toutes les parties d'une même pièce (volumes de matière) et poursuivre la dernière tâche de synthèse. La construction des pièces pourra alors s'effectuer par additions (opérations booléennes) des volumes qui les composent. Les modèles tridimensionnels seront disponibles pour une analyse informatique (éléments finis, par exemple) lors de la recombinaison du système.

Dans la maquette informatique actuelle de *Sicam 2*, les DTLM sont stockés dans une bibliothèque du logiciel de CAO employé pour la visualisation (I-DEAS®) sous forme d'assemblages de pièces paramétrées. Leur implémentation informatique est détaillée dans le chapitre 6.

L'introduction des modèles 3D paramétrés dans le système étudié nécessite des dimensions qui sont déterminées par les modules de calcul. La section suivante traite de la problématique du dimensionnement optimal des DTLM et présente la structure de données orientée objet qui leur est associée.

2.2 Aspect fonctionnel du DTLM.

Le DTLM assure une fonction de liaison. Il doit donc remplir cette fonction d'un point de vue géométrique et analytique. De nombreux travaux ont tenté de modéliser le produit et ses composants en tissant des liens entre la fonction et la matière. C'est l'objet des méthodes d'analyse fonctionnelle telles que SADT, FAST, BOND-GRAPH... Nous avons retenu une approche orientée objet, que nous développons ci-après, afin de représenter la structure de données liée à l'exploitation des DTLM. Nous y mettons en évidence la classification et les liens existants entre les objets de la structure en prenant pour exemple la liaison pivot. Enfin nous détaillons les calculs et la logique d'optimisation qui forment certaines des méthodes associées à chaque objet DTLM.

2.2.1 Approche orientée objet.

Tout système mettant en jeu des éléments mécaniques appartenant à une base de données nécessite une structure rigoureuse. Ces données sont généralement archivées sous forme de bibliothèques de composants mécaniques standards. Ces bibliothèques, souvent l'œuvre de fabricants de composants ou d'acteurs du domaine de la CAO (concepteurs de logiciel, intégrateurs), ne brillent pas par leur diversité puisque la plupart d'entre-elles sont consacrées aux éléments de visserie [Mar95]. Les composants mécaniques (les pièces) sont souvent classés par familles sur un modèle hiérarchique. L'utilisation de telles bibliothèques se justifie dans des cas de conception variés comme :

- la conception routinière pour laquelle un gros effort est mené dans le but d'automatiser l'utilisation de ces bases de données ;
- l'étude de variantes pour laquelle les concepts sont bien identifiés et que l'on tente de rationaliser ;
- la conception créative, moins concernée, qui utilise parfois ces données comme base de travail.

Ces bibliothèques sont parfois extensibles par l'utilisateur. Quelques principes de base reviennent régulièrement dans leur architecture : approche orientée objet, multi-représentation, modèles fonctionnels, géométries paramétrées, normes d'échanges de données... La conception d'une telle base doit donc répondre à des besoins comme :

- l'extensibilité de la base de données à de nouveaux éléments ;
- la lisibilité de la base par la mise en évidence de catégories d'informations ;
- l'automatisation de l'accès à la base de données par des logiciels ;
- la rationalisation de la gestion des informations pour éviter, par exemple, la redondance de certaines données ;
- la robustesse et la pérennité des modèles de représentation employés.

Cette structure doit donc s'appuyer sur une hiérarchisation des informations. Dans le cas des DTLM, la fonction *liaison mécanique* s'impose comme un critère logique de classement. Il serait alors souhaitable que notre structure de données soit caractérisée par :

- une lisibilité accrue de la base par le concepteur, notamment en mettant en lumière les relations entre les différents DTLM ;
- la capacité à modéliser les caractéristiques communes ou divergentes des DTLM.

Pour toutes ces raisons, nous avons retenu une *approche orientée objet* (nous entendons par « approche » l'ensemble des techniques orientées objet, ce qui va bien au delà de la simple programmation objet puisque comprenant aussi l'analyse et la conception). L'approche objet [AF96] [Fer91] [CY92] fournit un cadre de travail particulièrement bien adapté à notre problématique. Comme il est précisé dans [DF95], un objet se définit avant tout par les services qu'il offre. De ce

fait, la structure comprendra aussi bien les données que les opérations utiles à chaque objet (dimensionnement, visualisation. . .). Les caractéristiques principales reconnues de la représentation objet sont les suivantes :

Abstraction. C'est un mécanisme permettant de spécifier une entité en ne retenant que ses *caractéristiques essentielles*, indépendamment de la représentation interne. Le concepteur s'appuie sur l'identification de *similarité* entre objets, et la décision de se concentrer sur ces similarités en faisant abstraction des différences. De fait, l'abstraction procure des frontières conceptuelles rigoureuses par rapport à l'observateur.

Encapsulation. Elle est connexe à l'abstraction puisqu'il s'agit d'une méthode de *masquage des informations* qui occulte l'aspect interne de l'objet. Elle dissocie l'interface contractuelle des algorithmes et des données internes. Ainsi, les objets en relation ne se soucient pas de leur implémentation interne réciproque, diminuant ainsi les phénomènes d'interdépendance. L'encapsulation gagne en efficacité, dans un contexte de langages à objet, bien qu'elle n'y soit pas spécifique.

Polymorphisme. Principe selon lequel une valeur ou fonction peut appartenir à plusieurs types d'objet à la fois. Pratiquement, cela signifie qu'une même opération pourra s'appliquer sur différentes classes en ayant des comportements différents. Cette capacité d'un objet à être vu de différentes manières entraîne un typage dit faible.

Modularité. Cette propriété permet la décomposition d'un système d'objets en unités sémantiques fonctionnelles (les modules). Cette division fournit des entités de forte cohésion interne et de faible couplage externe. C'est une conséquence des capacités d'abstraction et d'encapsulation.

Héritage. Il traduit l'existence d'une *relation hiérarchique* de filiation entre les objets qui autorise la réutilisation par un *partage* des attributs et/ou des méthodes.

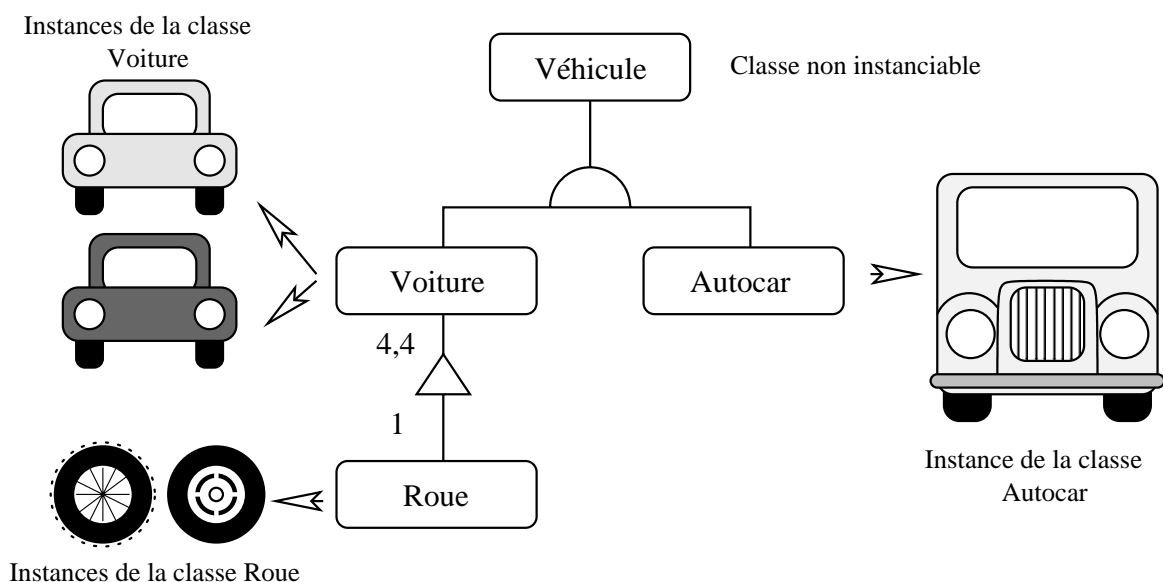


FIG. 2.10 – *Formalisme objet selon [Boo94].*

Toutes ces caractéristiques recouvrent une utilité dans notre modèle de représentation. Les objets sont en réalité des instances de classes, la classe étant considérée comme une matrice qui donne

une structure à ses objets ; ceux-ci se distinguent ensuite par la valeur de leurs attributs. Sur la figure 2.10, on remarque que la classe «voiture», qui est un *enfant* de la super-classe «véhicule», au même titre que la classe «autocar», peut être instanciée en objets «voiture». Ceux-ci ont en commun un ou plusieurs attributs, comme la «couleur», qui peuvent prendre différentes valeurs : par exemple, l'attribut «couleur» peut prendre les valeurs «foncé», «clair»... Les voitures et autocars sont des enfants de la classe «véhicule». Ils héritent donc des attributs et des méthodes des *parents*. Par exemple, nous pouvons envisager une méthode qui donnerait le nombre de roues. Cette méthode est alors capitalisée dans la classe parent puisque tous les véhicules ont des roues. Cette méthode est ensuite léguée à chaque classe de véhicule qui individuellement peuvent donner leur nombre de roues.

L'approche objet permet une construction par affinage car elle est basée sur une analyse descendante [Fer91]. Elle se prête donc admirablement à la définition par niveaux de mécanismes, et dans notre exemple, elle a conduit à une structure arborescente particulièrement intéressante. Il existe des méthodes de développement orientées objet qui fournissent un cadre structuré aux concepteurs de systèmes. La méthode OMT⁶ [Rum95] se décompose en quatre phases :

1. Analyse : elle vise à produire un modèle du monde réel à partir des spécifications du client. Le modèle précise les objectifs de l'application sans évoquer la moindre technique de réalisation. Il doit être compréhensible et discutable par les spécialistes du domaine de l'application.
2. Conception du système : le travail propre aux informaticiens commence ici. Le concepteur du système fait des choix concernant les performances du système, son architecture et établit une stratégie d'action.
3. Conception des objets : en reprenant le modèle établi durant l'analyse, le concepteur définit les structures de données et les algorithmes nécessaires à chaque classe en précisant les détails d'implémentation. Cette phase approfondit l'analyse abstraite du monde réel en prenant en compte les premiers choix techniques du concepteur système.
4. Implémentation : les classes d'objets créées précédemment sont traduites en code informatique. En principe, la phase de rédaction du programme ne doit pas présenter de difficultés particulières car toutes les décisions importantes ont été prises en amont. Le travail d'écriture du programme demande essentiellement de la rigueur afin de garantir une bonne traçabilité de l'ensemble du processus de conception. Le code n'en sera que plus compréhensible et évolutif.

De nombreuses méthodes orientées objet adoptent trois points de vue système : objet, dynamique et fonctionnel. La différence entre ces méthodes réside principalement dans la hiérarchisation de ces points de vue qui influencent la conception du système. Alors que certaines méthodes orientées objet mettent l'accent sur le découpage fonctionnel du système étudié (SA/SD⁷ par exemple), la méthode OMT privilégie le modèle objet. Le débat légitimiste est affaire de spécialistes et ne présente que peu d'intérêt pour le choix d'une méthode.

Nous détaillons ci-dessous, pour le cas de la liaison pivot, l'arborescence obtenue en appliquant le formalisme objet de G. Booch [Boo94]. En partant de la liaison pivot, nous pouvons proposer plusieurs architectures cinématiques telles que représentées sur la figure 2.11. La liaison pivot constitue l'objet parent tandis que les objets de niveaux inférieurs constituent l'ensemble

6. «Object Modeling Technique» où technique de modélisation des objet.

7. Structured Analysis / Structured Design

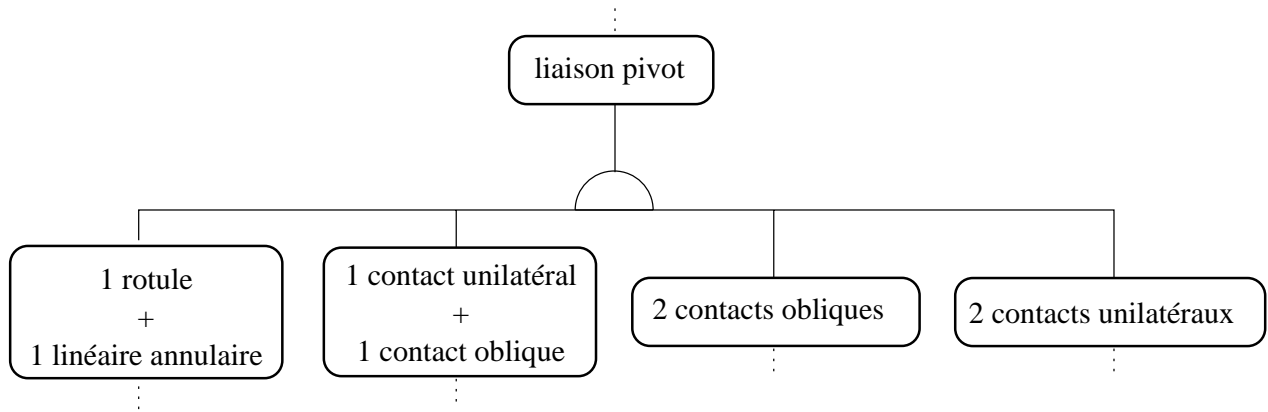


FIG. 2.11 – Décomposition d'une liaison pivot.

des enfants. Le symbole \oplus qui marque une notion d'héritage, encore appelée *généralisation-spécification* (ou gen-spec), indique que les attributs⁸ et les méthodes⁹ de l'objet parent sont transmises aux objets enfants. Littéralement, nous pouvons traduire cette relation d'héritage par le syntagme «...est un(e)...», ce qui dans le cas de la liaison pivot entraîne : «le composant 1 rotule + 1 linéaire est une liaison pivot». Lorsque le concepteur met en place une liaison pivot, il dispose des données telles que le couple transmis ou la vitesse de rotation. En choisissant une des architectures cinématiques proposées, on procède à une définition plus précise de cette liaison pivot. Le nouvel objet doit donc être considéré comme une spécialisation de son parent. Il hérite des attributs tels que la vitesse de rotation, auxquels on ajoute les torseurs d'efforts statiques pour chaque liaison.

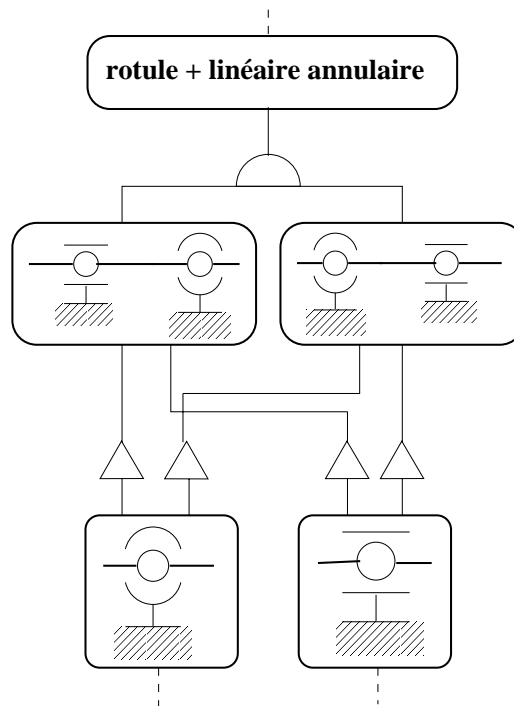


FIG. 2.12 – Architecture interne d'une liaison pivot.

8. Informations contenues dans les instances d'objet.

9. Opérations internes à l'objet.

L'étape suivante (figure 2.12) de la définition nous permet de préciser l'architecture interne de l'objet : positionnement des liaisons par exemple. L'objet est alors à même de calculer les torseurs au droit de chaque liaison, puis de tracer les diagrammes d'efforts, à partir des nouveaux attributs et de ceux hérités des niveaux supérieurs. La figure 2.12 nous montre qu'une redistribution est possible grâce à l'éclatement de l'objet « rotule + linéaire annulaire » en deux objets : l'objet « rotule » et l'objet « linéaire annulaire ». Le symbole $\hat{\Delta}$ introduit une nouvelle relation dans notre formalisme, appelée *composé-composant*¹⁰, qui indique que l'objet enfant est un composant de l'objet parent. On emploiera le syntagme « ...est une partie de... » pour exprimer que « la rotule est une partie de la liaison rotule + linéaire annulaire ». Notons encore qu'un même objet peut partager ses services entre plusieurs objets parents. Cette notion d'éclatement est importante car elle démontre la capacité de traitement individuel de chaque composant. Dans le cas d'une liaison pivot réalisée par une rotule et une linéaire annulaire, ces dernières peuvent être calculées et dimensionnées séparément, reflétant ainsi leur indépendance. La figure 2.13 montre

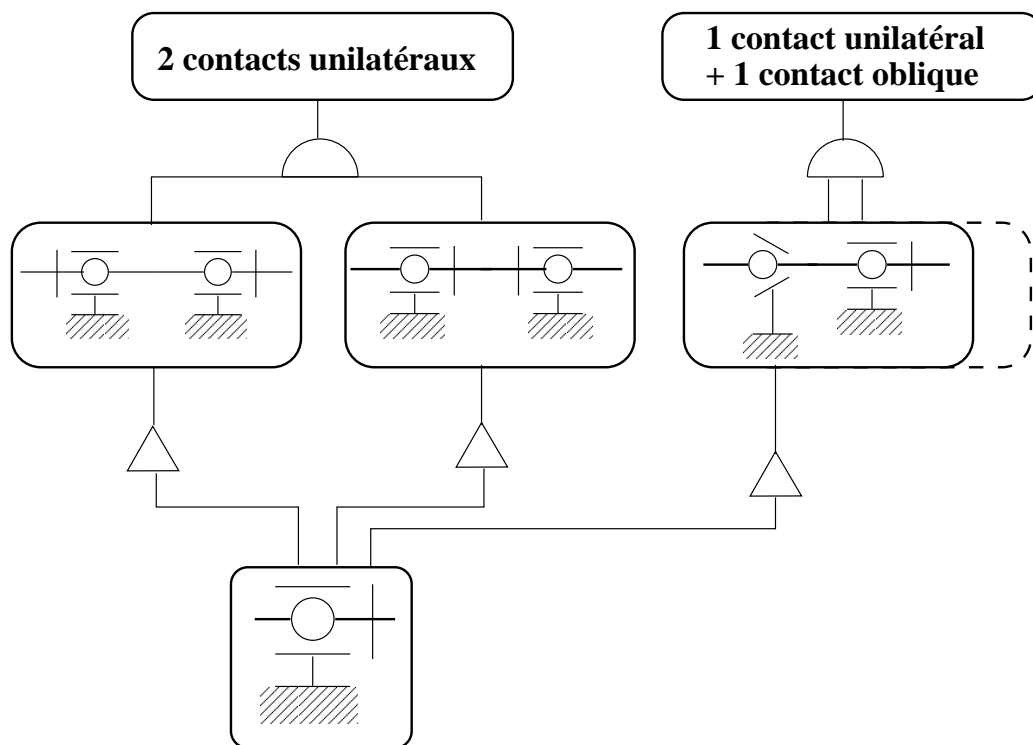


FIG. 2.13 – Rotule unilatérale fournissant ses services.

l'objet « rotule unilatérale » fournissant ses services à divers objets parents. Cet éclatement n'est malheureusement pas toujours réalisable. En effet, la condition d'indépendance entre les composants de la liaison n'est pas toujours satisfaite. Dans le cas d'une liaison pivot réalisée par des éléments roulants à contact oblique (figure 2.14) on ne peut séparer les deux éléments car la méthode de calcul de ce type de dispositif est itérative et requiert une connaissance simultanée des attributs de chacun d'eux.

Nous accédons au dernier niveau de notre arborescence. Il s'agit des DTLM qui viennent concrétiser les liaisons précédemment définies. Le concepteur construit sa solution à partir de la bibliothèque de ces composants. Plusieurs DTLM peuvent remplir la même fonction, comme on peut le constater sur la figure 2.15(a) avec la liaison rotule. Ils peuvent être interchangeés entre

10. "Whole-part" en anglais.

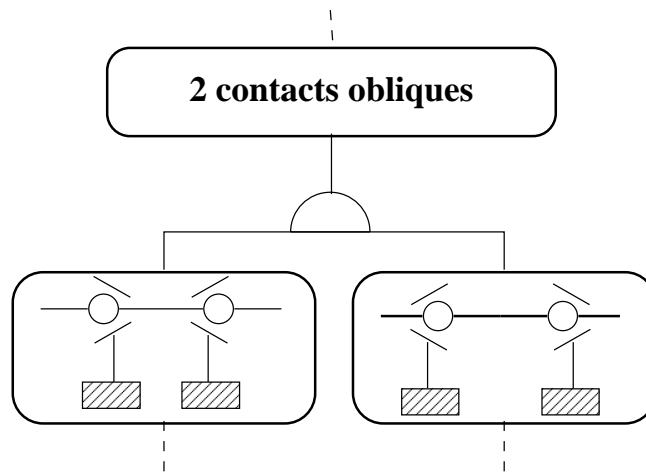
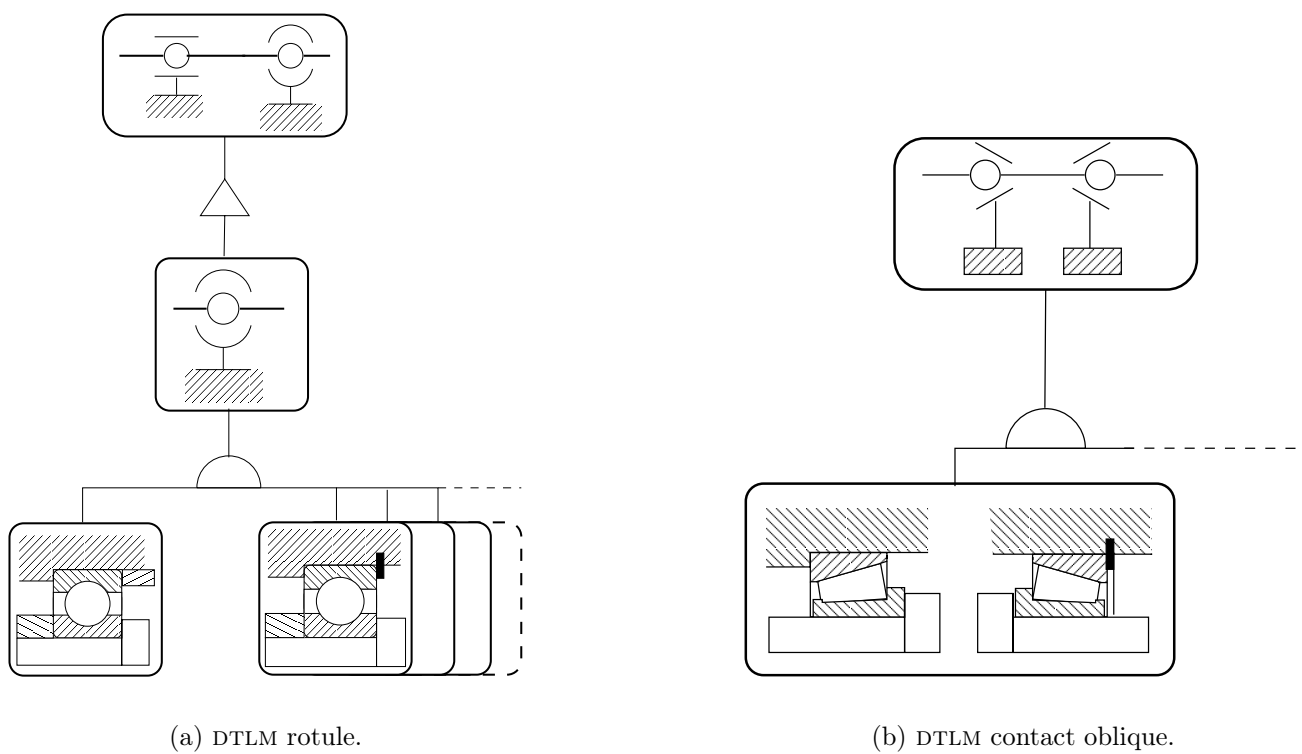


FIG. 2.14 – Exemple d'objets non divisibles.



(a) DTLM rotule.

(b) DTLM contact oblique.

FIG. 2.15 – Exemples de DTLM disponibles.

eux sans remettre en cause les niveaux supérieurs. Seuls les attributs propres aux DTLM seront modifiés, et non les informations héritées des niveaux supérieurs. Par exemple, si le concepteur décide de substituer le DTLM comportant un anneau élastique par un DTLM utilisant une entretoise, il ne remet pas en cause les calculs de statique qui ont été effectués au niveau supérieur, en l'occurrence celui de l'objet « rotule ». Par contre, il introduit de nouvelles méthodes de dimensionnement liées aux technologies employées : plus d'accident de forme provoqué par l'usinage de la gorge de l'anneau élastique, mais un calcul de matage pour l'appui de l'entretoise. Les DTLM sont donc tous regroupés en une famille d'objets et, dans le cas de la liaison rotule, forment une spécialisation de l'objet parent. Nous parlerons dans ce cas de DTLM *indépendants*. La liaison pivot par contact oblique ne présente pas la même structure (figure 2.15(b)). Puisque la liaison ne peut être décomposée, on fait directement appel aux composants de l'objet parent, c'est-à-dire les DTLM. La logique de calcul est respectée, et le dimensionnement de cette liaison est conditionné par le choix des DTLM. Nous parlerons alors de DTLM *dépendants*.

Quelques remarques sur cette structure peuvent être formulées. Tout d'abord, la structure est évolutive. Elle peut intégrer à tout moment de nouveaux objets à n'importe quel niveau. Sa représentation sous forme d'arborescence souligne parfaitement les liens hiérarchiques entre les différents objets qui, rappelons-le, matérialisent les niveaux de définition progressive de la fonction au DTLM. Elle dévoile aussi les complexités liées aux procédures de calcul en développant une arborescence dont les branches ne sont pas forcément semblables. Considérons l'exemple de la figure 2.15(b). Dans le cas des DTLM « oblique », nous avons vu que le calcul de statique ne pouvait être mené qu'une fois les DTLM déterminés. A l'inverse, le calcul de statique peut être mené dès la connaissance du découpage en « une rotule + une linéaire annulaire ». C'est une manifestation du principe de polymorphisme appliqué à la méthode « calcul de statique ». Chaque relation gen-spec correspond toujours à une étape de choix pour le concepteur, tandis que les relations composé-composant impliquent l'ajout de nouveaux attributs. La notion d'encapsulation est très présente dans la structure proposée. Les données sont visibles au niveau où elles sont nécessaires et transitent par les mécanismes d'héritage jusqu'au plus bas de l'arborescence. Elles sont ainsi conservées lorsqu'on procède à une substitution d'objets qui ne les contiennent pas directement. Les méthodes sont également concernées par l'encapsulation. Prenons le cas d'une substitution d'objets DTLM. Les règles de calcul, en fatigue, à la statique, etc. sont propres à chaque DTLM et sont activées de façon transparente par le système informatique qui utilise les DTLM. Ainsi, le système informatique n'a aucune connaissance du contenu de chaque DTLM et se contente d'invoquer les méthodes de dimensionnement, d'affichage. Cette façon de traiter les objets sans présumer de leur contenu garantit une souplesse considérable du système. Nous pensons enfin que la structure objet que nous présentons pourrait s'insérer dans une structure plus large d'objets métiers [AC98]. Le couplage des DTLM avec le logiciel de CAO I-DEAS[®] profite pleinement de cette approche objet. Nous présentons ci-après ce qui forme, sans conteste, la principale méthode liée aux objets DTLM. Il s'agit du calcul de dimensionnement des DTLM.

2.2.2 Dimensionnement et optimisation d'un DTLM.

Lorsque le concepteur choisit un DTLM pour réaliser une liaison mécanique, il doit faire face à un problème d'intégration de composants dans un système partiellement défini. Ce manque d'information oblige à fixer des limites au dimensionnement du DTLM, que nous traduisons en contraintes. Celles-ci sont de plusieurs natures :

Les contraintes de fonctionnement. Nous aurions pu les appeler *contraintes mécaniques*.

Elles sont formées à partir de l'ensemble des conditions de service du DTLM (durée de

vie, résistance mécanique, rendement énergétique...). Les critères sont généralement issus du CdCF. Il n'est pas rare que certains critères tels que la durée de vie ne soient pas homogènes sur l'ensemble du mécanisme. On peut préférer une maintenance plus fréquente sur certains composants pour de multiples raisons (gain de masse, de coût, de sécurité). Les DTLM seront donc dimensionnés selon des règles propres à chacun d'eux, qui peuvent varier en complexité : des modèles simples pour les composants d'importance moindre ou pour un prédimensionnement sommaire, à des modèles plus évolués dans le cas de composants dits « stratégiques » nécessitant un prédimensionnement plus précis. Toutes les combinaisons de complexité de modèles sont possibles au sein d'un même mécanisme.

Les contraintes d'encombrement axial et radial. Le DTLM doit s'intégrer dans le mécanisme sans provoquer d'interférences avec les autres constituants. Nous avons choisi d'utiliser la notion de *limites d'encombrement* pour contrôler l'espace dans lequel le DTLM devra se loger. Ces limites sont explicites, c'est-à-dire fixées par le concepteur — pour se réserver des volumes à d'autres fins par exemple — ou implicites, et dans ce cas imposées par les autres composants du système. Pour le cas des mécanismes que nous étudions (cf. section 1.3), les contraintes d'encombrement se limitent à des frontières axiales et radiales. Elles prennent la forme d'un diamètre maximum admissible (ldM) et de deux limites axiales

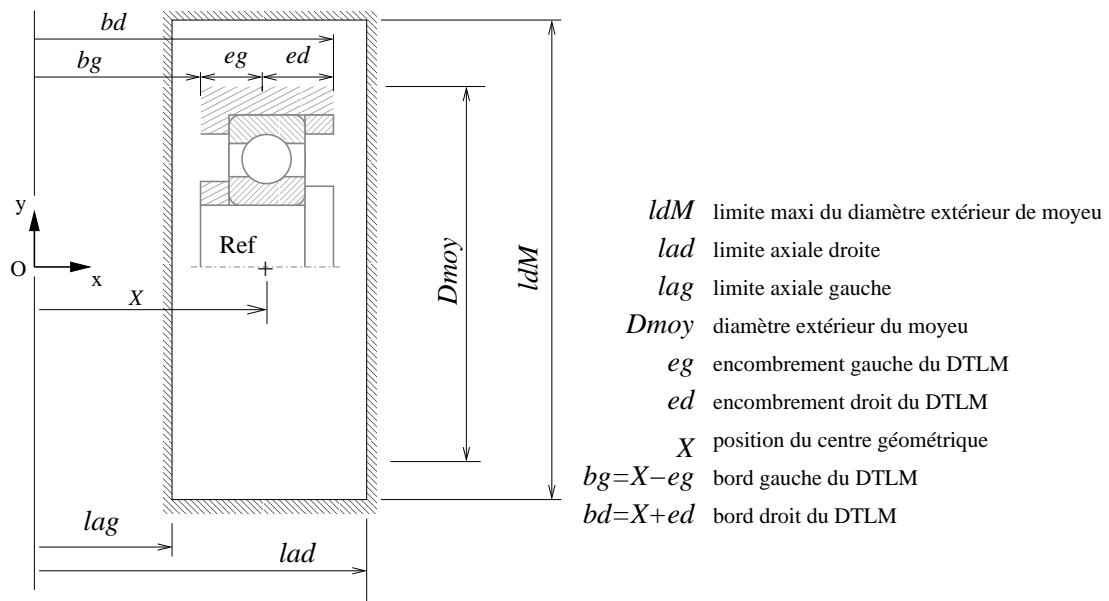


FIG. 2.16 – Contraintes d'encombrement sur un DTLM.

(lag à gauche et lad à droite). Elles sont symbolisées par un rectangle (section du cylindre formée par les limites) sur la figure 2.16. Les contraintes s'expriment de la manière générale suivante :

$$\left\| \begin{array}{l} bg \geq lag \\ bd \leq lad \\ Dmoy \leq ldM \end{array} \right.$$

Les contraintes sur les diamètres d'arbre. Le choix des DTLM fait suite à une étude de montage du mécanisme. Celle-ci impose un ordre de montage qui se traduit, pour les tronçons d'arbre, par un étagement des diamètres. Nous verrons que, dans le processus de synthèse découlée, le concepteur doit pouvoir imposer des valeurs minimum et maximum

de diamètres de tronçon d'arbre afin d'assurer le montage des divers composants mécaniques sur l'arbre, selon l'ordre et le sens voulus lors de l'étude de montage. La figure 2.17 met en place deux DTLM que nous nommons A et B. Il est important de remarquer que le choix des arrêts axiaux doit être compatible avec le sens de montage envisagé. Supposons

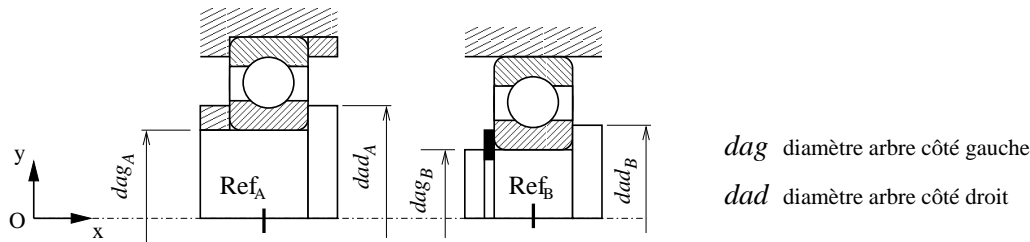


FIG. 2.17 – Montage de deux DTLM.

que le sens de montage imposé soit un montage par la gauche, c'est-à-dire que le roulement du DTLM B devra être monté par la gauche avant le roulement du DTLM A. Nous en déduisons la contrainte « brute » suivante :

$$\| \quad dad_A \leq dag_B$$

Notons que si nous avons considéré le problème symétrique, la contrainte aurait été inversée (en prenant soin de choisir des arrêts axiaux compatibles avec le sens de montage) :

$$\| \quad dad_A \geq dag_B$$

Pour chaque DTLM, nous devons donc nous doter de limites qui permettent de contrôler les plages de variations possibles pour les diamètres d'arbre. Cela facilite l'expression des contraintes traduisant les possibilités de montage. Pour une plus grande souplesse, et de par la réversibilité des contraintes, nous imposons un diamètre maximal et minimal de chaque côté de l'arbre. La figure 2.18 représente ces limites sur un exemple de DTLM. Nous

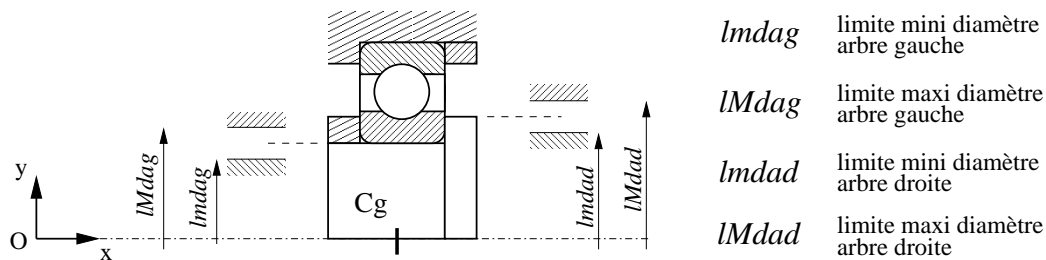


FIG. 2.18 – Contraintes de montage d'un DTLM.

pouvons donc exprimer, d'une façon plus générale pour tout DTLM i , les contraintes de montage :

$$\left\| \begin{array}{l} dag_i \leq lMdag_i \\ dag_i \geq lmdag_i \\ dad_i \leq lMdad_i \\ dad_i \geq lmdad_i \end{array} \right. \quad (2.1)$$

Plusieurs stratégies sont envisageables pour fixer les valeurs limites (propagation automatique des valeurs des diamètres d'arbre voisins, dichotomie sur les intervalles de violation...). La mise en place de ses diverses stratégies est facilitée par l'approche objet qui

autorise le dialogue entre les DTLM. Nous étudions cet aspect plus en profondeur dans le chapitre consacré à la synthèse coordonnée. Si nous reprenons l'exemple de la figure 2.17, l'algorithme de coordination pourra gérer les conditions de montage en imposant, par exemple :

$$\left\| \begin{array}{l} lMdad_A = \alpha \\ lmdad_A = 0 \end{array} \right. \quad \text{et} \quad \left\| \begin{array}{l} lMdag_B = \infty \\ lmdag_B = \alpha \end{array} \right.$$

ou α sera établi dynamiquement à partir de valeurs calculées localement (avant introduction des contraintes de couplage) pour dad_A et dag_B .

Le problème d'optimisation. Toutes ces contraintes forment un ensemble d'inéquations. Dès lors qu'on y ajoute un critère d'optimisation, exprimé par une fonction objectif, on dispose d'un problème d'optimisation contraint en bonne et due forme que l'on peut exprimer par : « Déterminer les n composantes du vecteur X des variables de conception du DTLM qui minimisent la fonction objectif $f(X)$ sous un ensemble de contraintes inégalités ». La forme mathématique est :

$$\begin{array}{l} \min_{X \in \Omega} f(X) \text{ avec } \Omega \subset R^n \\ \text{sous les contraintes } g_{j=1, \dots, m} \leq 0 \end{array}$$

Il est nécessaire de préciser ici que nous avons choisi de formuler un problème d'optimisation en raisonnant individuellement pour chaque DTLM, et bien sûr de créer la procédure de résolution associée. Cette approche nous a conduit à créer un ensemble de problèmes d'optimisation « localisés ». Notons que l'optimisation globale d'un système formé de plusieurs DTLM ne sera pas abordée par le biais de la formulation d'un problème d'optimisation global, mais par le biais d'une coordination des procédures d'optimisation locales. Plusieurs raisons nous ont conduit à ce choix :

- la première raison est d'ordre technique. Le nombre des systèmes qui peuvent être créés par association de DTLM est très élevé. Il est vite apparu qu'il nous serait impossible de prédéfinir et de coder tous les problèmes d'optimisation globaux envisageables. Par ailleurs, il semblait difficile de construire et de résoudre un problème d'optimisation global en associant dynamiquement des « morceaux de modèles » relatifs aux DTLM retenus par le concepteur.
- La deuxième raison est liée à la diversité des problèmes locaux rencontrés et des méthodes de résolution associées. Certains DTLM engendrent des problèmes en variables mixtes (continues et discrètes) alors que d'autres ne présentent que des variables discrètes.
- Troisièmement, les résultats des différentes optimisations locales se sont révélées une source intéressante d'informations pour le concepteur. En effet, en confrontant ces résultats à ceux fournis après coordination, le concepteur peut mesurer l'effet des contraintes de couplage.
- Enfin, l'approche sous forme de plusieurs sous-problèmes localisés permet d'introduire des objectifs différents au niveau de ces sous-problèmes.

Les différentes fonctions objectifs $f(x)$ que nous avons retenues jusqu'ici (au niveau de chaque DTLM) sont :

- l'encombrement radial : $f(x) = eg + ed$

- le diamètre d’arbre : $f(x) = \max(dag, dad)$
- le volume du tronçon d’arbre : $f(x)$ dépend des arrêts axiaux

Nous n’avons pas proposé l’optimisation de la masse (difficile à évaluer pour des composants aussi hétérogènes que les DTLM et finalement assez proche du volume) ni du coût (très difficile à évaluer ponctuellement en phase d’avant-projet). Le concepteur peut donc tester tout type de solutions extrêmes en s’orientant vers un gain d’espace axial, radial, ou une compacité accrue.

Dans le paragraphe suivant, nous illustrons le problème de dimensionnement optimal que nous venons d’évoquer en prenant pour exemple la création d’un DTLM réalisant une liaison complète.

2.3 Création des DTLM.

2.3.1 Exemple de modélisation d’une famille de DTLM : la liaison complète par clavette.

Parmi les familles de DTLM que comporte *Sicam 2*, figure la liaison arbre/moyeu assurée par clavette. Cette liaison, familière au concepteur en mécanique, illustre bien les difficultés rencontrées lorsqu’on cherche à dimensionner une liaison de manière optimale. C’est aussi l’occasion de discuter des techniques d’optimisation envisageables à ce niveau de la conception.

2.3.1.1 Présentation du problème.

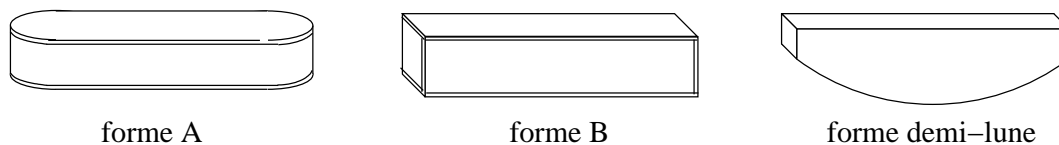


FIG. 2.19 – Différentes géométries de clavettes.

La liaison complète par clavette est une liaison d’entraînement par obstacle démontable. Le principe de ce type d’entraînement consiste en l’interposition, entre les deux pièces à coupler, d’un troisième corps qui les unit cinématiquement en rotation. Cette troisième pièce peut prendre le nom de clavette, coin, goupille, dépendant de la géométrie du corps et de sa position. La clavette est généralement parallélépipédique et nécessite pour son montage des aménagements de formes (usinages) dans l’arbre et le moyeu. Plusieurs géométries de clavettes existent (figure 2.19). On distingue assez facilement les surfaces fonctionnelles dans la liaison par clavette, notamment ses flancs. Pour constituer la bibliothèque de DTLM de *Sicam 2*, nous n’avons retenu que les liaisons impliquant des clavettes de formes A et B. Un recensement des combinaisons d’arrêts axiaux nécessaires à l’immobilisation du moyeu a donné les résultats de la figure 2.20. Hormis la combinaison épaulement–épaulement, toutes les solutions techniques sont réalisables, mais certaines présentent moins d’intérêt que d’autres, d’où notre classement subjectif. On dénombre ainsi vingt-deux combinaisons (symétrie oblige !) d’arrêts axiaux pour chaque type de clavette, soit un total de quarante-quatre DTLM à concevoir et paramétrer. Nous présentons dans le paragraphe suivant la formulation du problème d’optimisation de l’un de ces DTLM.

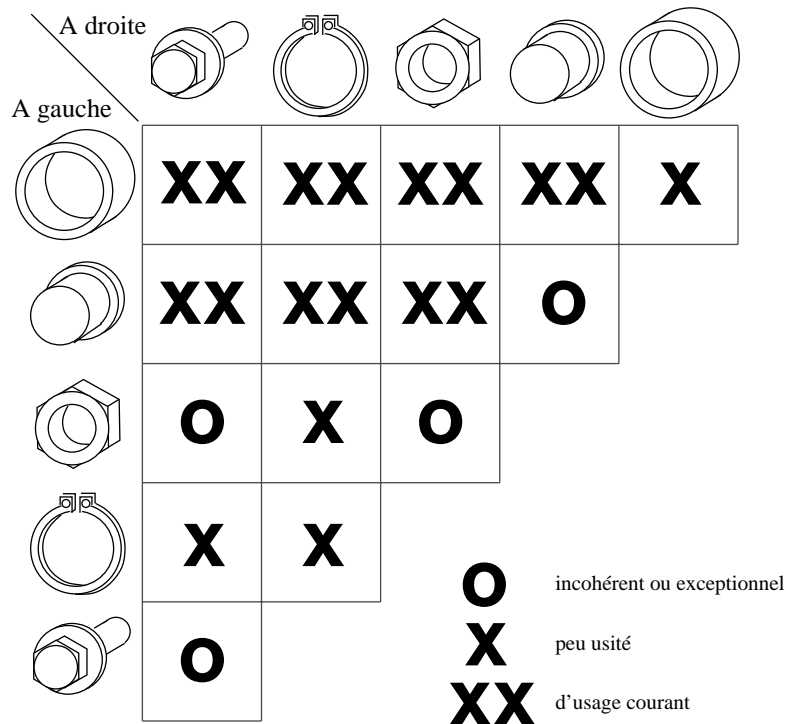


FIG. 2.20 – Combinaisons d'arrêts axiaux.

2.3.1.2 Formulation du problème de dimensionnement optimal de la liaison.

Le problème de dimensionnement optimal de la liaison peut s'exprimer de la façon suivante :

« Quelles valeurs optimales doit-on attribuer aux paramètres dimensionnels du DTLM pour satisfaire les conditions de fonctionnement de la liaison complète, en respect d'un critère d'optimisation fixé par le concepteur ? »

Nous allons traduire cette question en un problème d'optimisation contraint. Nous commençons par énumérer toutes les variables dimensionnelles qui interviendront dans la formulation mathématique du problème en notant que le choix des matériaux ne constitue pas une dimension supplémentaire du problème. Nous dressons ensuite la liste des contraintes du problème en utilisant la classification présentée ci-avant (fonctionnement, encombrement et montage).

Les variables.

Nous avons discuté de la composition d'un DTLM au paragraphe 2.1.1. Le DTLM sur lequel nous appuyons notre démonstration réalise une liaison complète démontable par clavetage libre (clavette parallèle de forme A), entretoise et épaulement (cf. figure 2.21). Ce type de clavetage peut ne pas être adapté aux sollicitations rencontrées sur les mécanismes conçus avec *Sicam* lorsque les puissances à transmettre sont grandes. Le concepteur a cependant la totale liberté de tester ce DTLM au même titre que les autres, plus robustes, tels que les assemblages par cannelures. La figure 2.22 représente un tronçon d'arbre appartenant à une liaison clavetée dotée de ses aménagements d'arrêts axiaux. Les figures 2.23 et 2.24 complètent le paramétrage en introduisant respectivement les dimensions liées à la clavette et au moyeu.

Cherchant ici à obtenir automatiquement le dimensionnement du DTLM, nous devons mettre en place un processus apte à produire toutes les dimensions qui définissent complètement le

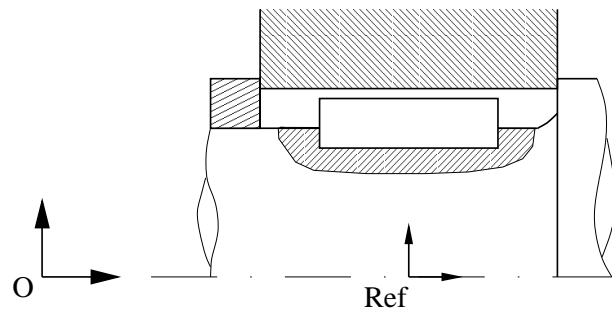


FIG. 2.21 – DTLM *liaison complète par clavette, entretoise et épaulement.*

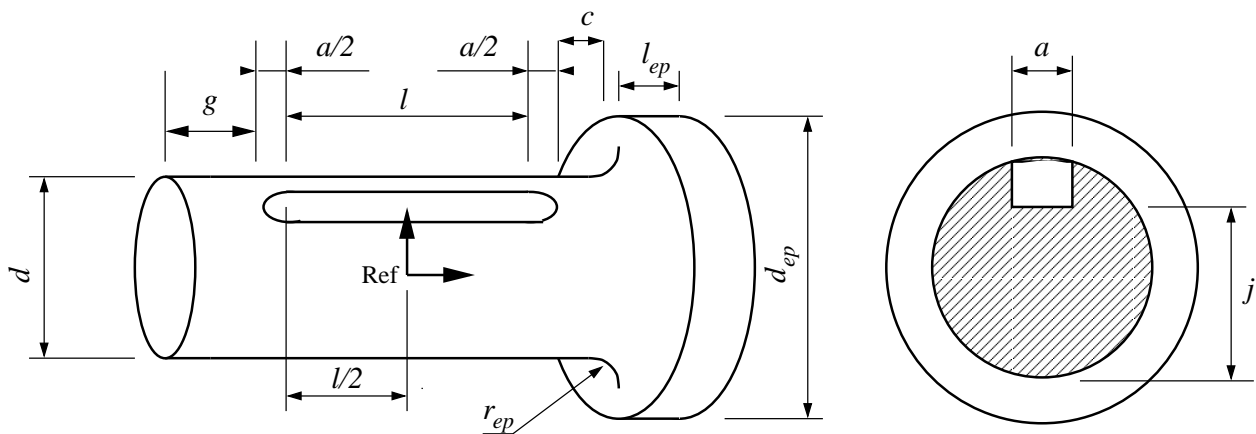


FIG. 2.22 – *Exemple de paramétrage de l'arbre du DTLM.*

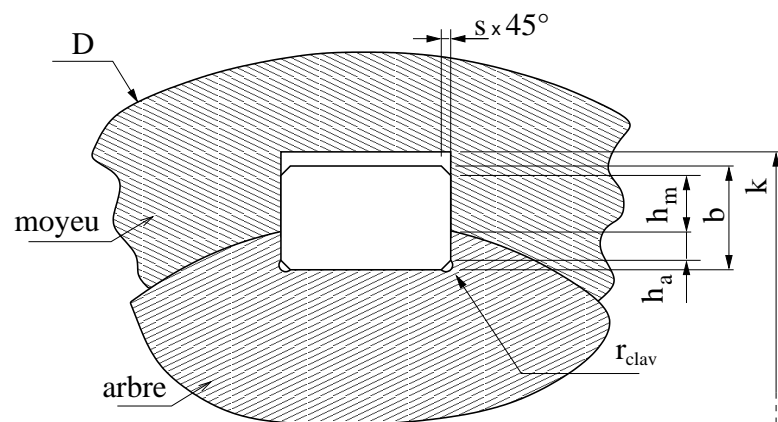


FIG. 2.23 – *Détail du paramétrage de la clavette.*

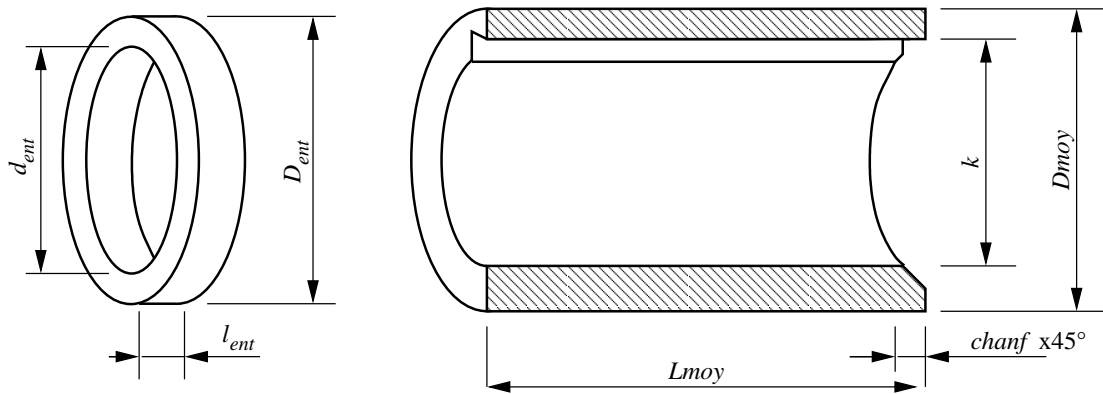


FIG. 2.24 – Paramétrage de l'entretoise et du moyeu.

DTLM. Il est clair que plus le nombre de caractéristiques qui sont considérées comme indépendantes (les « variables ») est grand, plus le processus nécessaire pour définir la solution est complexe. Dans le cas qui nous intéresse, comme dans le cas de tout autre DTLM, il serait aberrant de considérer que tous les paramètres dimensionnels (ceux qui apparaissent sur les figures 2.22 à 2.24) sont des variables à part entière.

D'une part, les DTLM mettent en œuvre des composants normalisés dont les dimensions découlent en général d'une seule dimension de la pièce qui les supporte. Dans le cas de la clavette, la plupart des dimensions (j , k , r , a , b ...) dépendent directement du diamètre d de l'arbre. Elles proviennent des normes [AFN87b] et [AFN87c] relatives aux assemblages clavetés (cf. figure 2.25). Seule la longueur l de la clavette reste une caractéristique qui peut varier entre deux limites (qui dépendent de d). Il en va de même pour d'autres composants standards tels que les anneaux élastiques dont les dimensions issues des catalogues de fournisseur dépendent directement du diamètre d'arbre.

ARBRES*		CLAVETAGES PAR CLAVETTES USUELLES															
		ARBRE-MOYEU			CLAVETTES USUELLES						CLAVETTES FIXÉES PAR VIS						
d	à	j	k	r	a	b	longueurs		min.	s	max.	c	d ₁	vis C ou vis CHC	d ₂	e	f
6	8	d - 1,2	d + 1	0,08	2	2	6	20	0,16	0,25							
8	10	d - 1,8	d + 1,4	0,16	3	3	6	36	0,16	0,25							
10	12	d - 2,5	d + 1,8	max.	4	4	8	45	0,16	0,25							
12	17	d - 3	d + 2,3	0,16	5	5	10	56	0,25	0,4							
17	22	d - 3,5	d + 2,8	0,25	6	6	14	70	0,25	0,4							
22	30	d - 4	d + 3,3	max.	8	7	18	90	0,25	0,4	6,5	M3	M3-8	3,4	3,5	6,5	
30	38	d - 5	d + 3,3	0,25	10	8	22	110	0,4	0,6	8	M4	M4-10	4,5	4,5	9	
38	44	d - 5	d + 3,3	0,25	12	8	28	140	0,4	0,6	10	M5	M5-10	5,5	5,5	10,5	
44	50	d - 5,5	d + 3,8	min.	14	9	36	180	0,4	0,6	12	M6	M6-10	6,6	6,5	11,5	
50	58	d - 6	d + 4,3	0,4	16	10	45	180	0,4	0,6	12	M6	M6-10	6,6	6,5	10,5	
58	65	d - 7	d + 4,4	max.	18	11	50	200	0,4	0,6	16	M8	M8-12	9	8,5	14,5	
65	75	d - 7,5	d + 4,9	0,4	20	12	56	220	0,6	0,8	16	M8	M8-12	9	8,5	13,5	
75	85	d - 9	d + 5,4	min.	22	14	63	250	0,6	0,8	20	M10	M10-12	11	10,5	14,5	
85	95	d - 9	d + 5,4	0,6	25	14	70	280	0,6	0,8	20	M10	M10-12	11	10,5	14,5	
95	110	d - 10	d + 5,4	max.	28	16	80	300	0,6	0,8	20	M10	M10-16	11	10,5	16,5	

FIG. 2.25 – Dimensions normalisées des assemblages clavetés (extraits de la norme NF E 22-175 et NF E 22-177).

D'autre part, il existe pour les éléments restants (ceux qui ne sont pas normalisés) des lois d'évolution qui dictent les dimensions secondaires à partir de quelques dimensions principales. Quelques travaux ont traité du problème des géométries et des accidents de forme influant

sur le fonctionnement de la liaison clavetée ([FE82], [FE83], [FA89] et [Bor86]). Ils concluent unanimement sur l'importance du choix de la géométrie employée. Bien sûr, ces lois ne sont pas toujours clairement formalisées mais existent implicitement dans la pratique. On peut parler grossièrement de facteur d'échelle. Dans le cas que nous traitons ici, les dimensions qui définissent l'épaulement de l'arbre répondent à ce type de phénomène. Évidemment, c'est le diamètre d de l'arbre qui joue un rôle essentiel sur l'évolution de ces dimensions. Afin de diminuer autant que possible le nombre des variables du problème, nous avons cherché ici à définir en fonction de d les trois dimensions de l'épaulement. Pour ce faire, nous avons introduit des équations qui permettent un calcul direct :

- Rayon de raccordement :

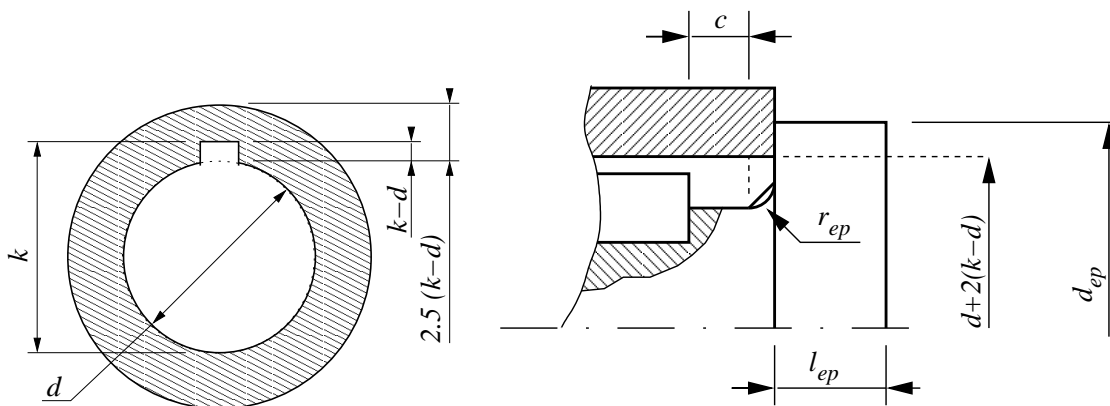
$$r_{ep} = \frac{k - d}{2} \quad (2.2)$$

où k dépend de d (voir fig. 2.25). Le choix d'une telle valeur de rayon produit des facteurs de concentration de contraintes dont la valeur, par expérience, excède rarement 2.

- Diamètre de l'épaulement :

$$d_{ep} = \sqrt{\frac{4.N.secu}{\pi.p_{adm}} + [d + 2(k - d)]^2} \quad (2.3)$$

où N est la composante axiale de la résultante du torseur d'efforts, $secu$ est un coefficient de sécurité et p_{adm} la pression admissible par l'épaulement de l'arbre. Le résultat correspond à un diamètre minimal d'épaulement égal au diamètre d'arbre additionné de deux fois la hauteur de rainure de clavette du moyeu (cas où N est nul). Dans le cas d'un effort axial non nul, on augmente la surface de contact en respectant une condition de non matage sur les portées.



(a) Définition de l'épaisseur de moyeu.

(b) Définition du diamètre d'épaulement.

FIG. 2.26 – Définition des relations dimensionnelles.

- Longueur de l'épaulement : elle est prise égale à la hauteur de rainure du moyeu.

$$l_{ep} = k - d \quad (2.4)$$

Il est possible de procéder de la même manière pour les dimensions restantes de l'arbre et du moyeu. Ainsi :

- Rainure de clavette :

$$c = d - j \quad (2.5)$$

avec j dépendant de d (voir fig. 2.25). Cette valeur définit le volume de matière encadrant, de part et d'autre, la rainure de clavette. Cette épaisseur a pour but d'éloigner d'éventuels accidents de forme de la zone déjà sous l'influence de la rainure de clavette. Elle est donc définie proportionnellement à la profondeur de cette rainure qui, elle-même, dépend du diamètre de l'arbre.

- Diamètre extérieur du moyeu :

$$D_{moy} = 5k - 4d \quad (2.6)$$

qui dépend directement de la dimension du fond de sa rainure de clavette (donc du diamètre d'arbre). Nous avons défini une épaisseur minimale de matière du moyeu équivalente à 2,5 fois la hauteur de rainure du moyeu comme représenté sur la figure 2.26(a). Notons que le chanfrein à 45° *chanf* du moyeu adopte bien entendu la valeur du rayon de raccordement de l'épaulement r_{ep} définie ci-dessus.

$$chanf = r_{ep}$$

- Largeur et hauteur d'entretoise :

$$\begin{aligned} d_{ent} &= d \\ l_{ent} &= d - j \\ D_{ent} &= \sqrt{\frac{4 \cdot N \cdot secu}{\pi \cdot p_{adm}} + (d + 4)^2} \end{aligned} \quad (2.7)$$

le diamètre intérieur d'entretoise adopte naturellement la valeur du diamètre d'arbre tandis que son diamètre extérieur est dimensionné au matage pour les mêmes considérations que l'épaulement (en respectant une épaisseur minimale de 2 mm liée à sa réalisation).

- Longueur après clavette :

$$\begin{aligned} g &= c + l_{ent} \\ &= 2d - 2j \end{aligned}$$

qui permet d'adopter une longueur de tronçon d'arbre compatible avec la longueur de l'entretoise l_{ent} et tenant compte du volume « tampon » de matière en extrémité de rainure de clavette.

Le paramétrage dimensionnel suscite quelques commentaires. Il convient de noter qu'un paramétrage de DTLM doit être le plus simple possible et refléter une fonctionnalité propre à chaque dimension, ceci afin de limiter le nombre d'équations de liaison entre les paramètres dimensionnels.

La figure 2.27 représente deux paramétrages différents d'une même pièce (le tronçon d'arbre du DTLM défini ci-avant). On constate que le paramétrage représenté en (a) reprend intégralement les dimensions que le concepteur a l'habitude de manipuler avec ce type d'élément (dimensions « catalogue », épaisseurs de matière. . .). En revanche, le paramétrage (b) obligera le concepteur

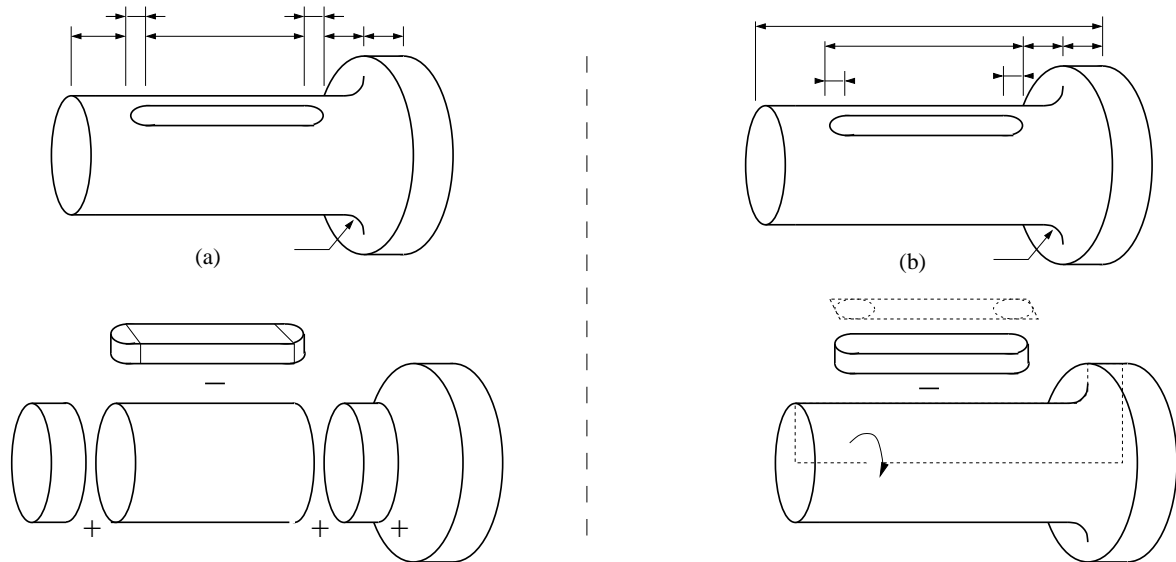


FIG. 2.27 – Exemples de paramétrages sur l'arbre de la liaison clavetée.

à une gymnastique arithmétique pour retrouver des variables ayant *un sens mécanique* (chaînes de cotes, donc équations supplémentaires). Considérons l'exemple de la longueur de clavette. Une analyse fonctionnelle de la clavette de forme A montre rapidement que la transmission du couple s'effectue par le contact entre les flancs de la clavette et les gorges de l'arbre et du moyeu. Les flancs de la clavette sont définis par une surface, elle-même définie par une longueur et une hauteur. Le paramétrage proposé en (a) associe cette longueur « utile » de la clavette, c'est-à-dire celle qui sera utilisée dans les calculs de résistance, et les longueurs d'arrondis en bout de clavette. Le paramétrage (b) caractérise la rainure de clavette par sa longueur totale, à laquelle nous devons, par une opération mathématique, retrancher les longueurs d'arrondis pour obtenir la longueur « utile ». Cet exemple simpliste démontre qu'un paramétrage fonctionnel apporte un gain de temps dans le traitement des données en sus d'une meilleure compréhension du problème physique. Le second aspect du problème de paramétrage concerne la construction des modèles CAO associés aux DTLM. Aux paramétrages (a) et (b) sont associés ce qui pourraient être deux méthodes de construction du modèle 3D correspondant. La première fait appel à un assemblage massif de volumes primaires en s'appuyant sur les paramètres dimensionnels indépendants. À l'inverse, la seconde méthode concentre un maximum d'informations dans un filaire¹¹ utilisé pour générer le tronçon d'arbre complet. Seule la rainure de clavette est ensuite « ajoutée » au modèle par opération booléenne (soustraction).

Une étude rigoureuse du paramétrage des DTLM est donc un gage de simplification des étapes ultérieures de modélisation et de résolution du problème de dimensionnement optimal, notamment par la réduction du nombre d'équations intermédiaires. Notons toutefois que le paramétrage est un paramétrage « local » qui ne concerne que des parties de pièces. Il est donc impossible, à ce stade de la synthèse, de mettre en place une cotation fonctionnelle qui devrait logiquement s'appuyer sur l'ensemble de la pièce considérée.

Finalement, seules deux variables ont été introduites dans la modélisation du DTLM considéré : il s'agit de d (diamètre de l'arbre) et l (longueur de la clavette). Il est évident qu'elles représentent les deux dimensions les plus significatives de ce DTLM. Il faut rappeler, à ce stade de l'étude, que notre travail se situe dans une phase d'avant-projet où l'optimisation n'est utilisée que pour

11. Ensemble de segments et courbes définissant la section d'une pièce obtenue par révolution ou extrusion.

guider le dimensionnement dans une direction d'ensemble souhaitée par le concepteur. Le but n'est pas de faire un travail extrêmement précis où chaque détail serait minutieusement étudié afin de minimiser autant que possible la fonction objectif.

Nous allons maintenant exprimer les diverses contraintes issues des conditions limites que nous imposons au DTLM et à la liaison.

Les contraintes de fonctionnement.

Nous avons choisi d'appliquer des règles de dimensionnement simples et éprouvées afin d'illustrer la démarche de mise en équation du problème. Cette vision est réductrice d'un point de vue physique, mais suffisamment complète pour appréhender les difficultés inhérentes à une démarche de calcul de prédimensionnement. Nous avons donc décidé de calculer la torsion de l'arbre et le matage des portées (même si l'une de ces deux sollicitations est souvent prépondérante). Ces règles de conception reflètent fidèlement les contraintes caractéristiques de ce genre de liaison. Nous avons retenu dans un premier temps les sollicitations suivantes :

- Torsion de l'arbre. La sollicitation de cisaillement due à la torsion de l'arbre est :

$$\tau = \frac{16 C}{\pi \cdot d^3} \quad \text{avec } C = \text{couple transmis} \quad (2.8)$$

La valeur maximale admissible de la sollicitation de torsion dans l'arbre est :

$$\tau_{adm} = \frac{R_{pg}}{K_{ts} \cdot secu} \quad \text{avec} \quad \left\{ \begin{array}{l} \bullet R_{pg} \text{ résistance pratique au glissement} \\ \bullet K_{ts} \text{ coefficient de concentration de contraintes} \\ \text{en torsion selon [Pil97] lié à la rainure de clavette} \\ \bullet secu \text{ coefficient de sécurité} \end{array} \right. \quad (2.9)$$

Nous déduisons des expressions 2.8 et 2.9 la formulation de la contrainte de torsion dans l'arbre :

$$\tau \leq \tau_{adm} \quad \text{ou encore} \quad \tau - \tau_{adm} \leq 0$$

qui peut être mise sous la forme standard suivante :

$$g_1 = 1 - \frac{d}{\sqrt[3]{\frac{16 C \cdot K_{ts} \cdot secu}{\pi \cdot R_{pg}}}} \quad (2.10)$$

- Matage. L'expression générale du matage sur les flancs de la clavette est :

$$p_{matage} = \frac{2 C \cdot secu}{l \cdot d' \cdot h} \quad \text{avec} \quad \left\{ \begin{array}{l} \bullet l \text{ longueur utile de clavette} \\ \bullet h \text{ hauteur de flanc} \\ \bullet d' \text{ diamètre d'application du couple} \end{array} \right. \quad (2.11)$$

Le matage peut intervenir sur la surface de contact clavette/moyeu comme sur la surface de contact clavette/arbre. De plus, l'endommagement peut être constaté sur la clavette, l'arbre ou le moyeu ; sa localisation dépend du choix des matériaux. Ce n'est donc pas une mais quatre équations de contraintes qui découleront de l'expression du matage par la combinaison des hauteurs de flancs arbre/clavette (notée h_a) et moyeu/clavette (notée h_m) et des pressions admissibles sur l'arbre, le moyeu et la clavette (p_{adm_arbre} , $p_{adm_clavette}$ et p_{adm_moyeu}). L'expression des hauteurs de flancs est la suivante :

$$h_a = \frac{d}{2} \left(1 + \cos \left(a \sin \left(\frac{a}{d} \right) \right) \right) - j - s \quad \text{et} \quad h_m = b - h_a - 2s \quad (2.12)$$

où s est la dimension du chanfrein de la clavette.

L'effort générant le matage découle du couple transmis et dépend donc directement de son diamètre d'application. Nous choisissons d'affecter à d' les valeurs des expressions simplifiées suivantes afin de tenir compte de la différence d'effort appliqué sur les flancs, côté moyeu ou côté arbre.

$$d' = \begin{cases} d - h_a & \text{côté arbre} \\ d + h_m & \text{côté moyeu} \end{cases}$$

Nous pouvons donc exprimer les contraintes dues au matage en écrivant la condition de résistance pour chaque combinaison :

$$\left\| \begin{array}{l} \frac{2 C.secu}{l.(d-h_a).h_a} \leq p_{adm_arbre} \text{ matage du flanc de l'arbre} \\ \frac{2 C.secu}{l.(d-h_a).h_a} \leq p_{adm_clavette} \text{ matage du flanc de clavette coté arbre} \\ \frac{2 C.secu}{l.(d+h_m).h_m} \leq p_{adm_clavette} \text{ matage du flanc de clavette coté moyeu} \\ \frac{2 C.secu}{l.(d+h_m).h_m} \leq p_{adm_moyeu} \text{ matage du flanc de moyeu} \end{array} \right.$$

Nous pouvons tenter d'éliminer des contraintes de plusieurs façons. La première consiste à repérer le matériau le plus « faible » en regard de chaque contact (arbre/clavette et clavette/moyeu). Cela nous permet d'éliminer deux contraintes.

Une autre solution consiste à étudier et prévoir qualitativement les sollicitations sur les deux types de flancs de la clavette. Le matériau étant identique, le flanc présentant la plus grande pression de contact sera le flanc à dimensionner au matage. Cette méthode peut nous permettre d'éliminer une des deux contraintes liées à la clavette. Nous avons donc comparé l'expression des pressions de matage sur les flancs :

$$\frac{2 C.secu}{l.(d-h_a).h_a} < ? > \frac{2 C.secu}{l.(d+h_m).h_m}$$

ce qui revient à étudier le quotient

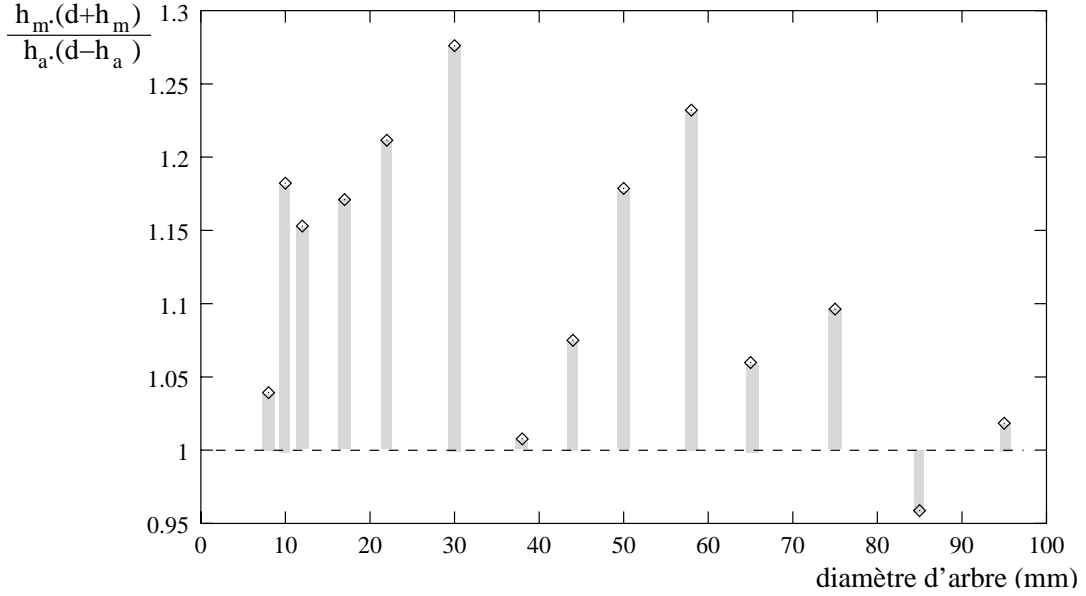
$$\frac{(d+h_m).h_m}{(d-h_a).h_a}$$

Nous avons donc exprimé les hauteurs de flancs pour toutes les sections de clavette, en nous plaçant toujours dans le cas le plus défavorable, i.e. clavette la plus grande autorisée sur le diamètre le plus faible de la plage de diamètres considérée. L'étude du quotient sur une partie du domaine de diamètre d'arbre a fourni les résultats représentés sur la figure 2.28. Nous constatons qu'il nous est impossible d'anticiper la détermination du flanc de clavette le plus sollicité puisqu'il ne peut être établi de hiérarchie sur l'ensemble du domaine étudié. Malgré la prépondérance du phénomène de matage sur le contact arbre/clavette, il survient une inversion de la tendance pour quelques valeurs de diamètre d'arbre.

Pour des raisons de commodité dans l'écriture du problème d'optimisation, nous avons conservé les quatre contraintes dont les expressions sont données ci-dessous, tout en gardant à l'esprit qu'il est peu probable que plusieurs d'entre elles soient simultanément actives :

$$g_2 = 1 - \frac{l.(d-h_a).h_a.p_{adm_arbre}}{2 C.secu} \quad (2.13)$$

$$g_3 = 1 - \frac{l.(d-h_a).h_a.p_{adm_clav}}{2 C.secu} \quad (2.14)$$

FIG. 2.28 – *Quotient de sollicitation au matage des flancs de clavette.*

$$g_4 = 1 - \frac{l \cdot (d + h_m) \cdot h_m \cdot p_{adm_clav}}{2 C \cdot secu} \quad (2.15)$$

$$g_5 = 1 - \frac{l \cdot (d + h_m) \cdot h_m \cdot p_{adm_moy}}{2 C \cdot secu} \quad (2.16)$$

- Longueur clavetée. La norme définit une longueur maximale de la clavette. Cette longueur dépend de la section de la clavette. Nous avons choisi d'utiliser la règle empirique suivante qui fixe la longueur de clavette maximale à une fois et demi le diamètre d'arbre :

$$l_{max} = 1,5 d \quad (2.17)$$

La condition $l \leq l_{max}$ génère donc la contrainte suivante :

$$g_6 = \frac{l}{1,5 d} - 1 \quad (2.18)$$

Selon le même principe, nous fixons une longueur minimale à la clavette qui garantira sa fabricabilité ainsi qu'un aspect esthétique (clavette de base rectangulaire). Nous choisissons un rapport minimal entre la largeur et la longueur de clavette de 1,5 :

$$l_{min} = 1,5 a$$

D'où la contrainte ($l \geq l_{min}$)

$$g_7 = 1 - \frac{l}{1,5 a} \quad (2.19)$$

- Sollicitations dans la section épaulée de l'arbre. Elles sont déterminées à partir des moments de flexion et de torsion dans la section considérée selon le critère de Von Mises :

$$\sigma_{epaul} = \sqrt{(k_{tf} \cdot \sigma_f)^2 + 3(k_{tt} \cdot \tau)^2} \quad \text{avec} \quad \sigma_f = \frac{32 M f}{\pi d^3} \quad \text{et} \quad \tau = \frac{16 M t}{\pi d^3}$$

Mt et Mf sont respectivement les moments de torsion et de flexion dans la section considérée. Ils sont déterminés à partir des résultats du calcul de statique en réalisant une interpolation des résultats obtenus pour certaines sections caractéristiques de l'arbre. Les coefficients k_{tf} et k_{tt} (dont nous n'avons pas détaillé l'expression) sont relatifs à un accident de forme sur un arbre plein provoqué par la réalisation d'un épaulement. En exprimant l'inégalité $\sigma_{epaul} \leq \sigma_{adm}$ on obtient la contrainte :

$$g_8 = \frac{\sqrt{\left(\frac{32 Mf k_{tf}}{\pi d^3}\right)^2 + 3\left(\frac{16 Mt k_{tt}}{\pi d^3}\right)^2}}{\sigma_{adm}} - 1 \quad (2.20)$$

Les contraintes d'encombrement.

Le concepteur a fixé des limites axiales (droite et gauche) ainsi qu'une limite radiale à l'encombrement du DTLM.

- Encombrement diamétral. Le diamètre extérieur du moyeu admet une valeur limite ldM :

$$g_9 = \frac{D_{moy}}{ldM} - 1 \quad (2.21)$$

- Les limites axiales droite et gauche sont exprimées dans le cahier des charges global à partir du repère global (cf. figure 2.16). Nous avons précisé auparavant que le placement du modèle 3D du DTLM devait s'effectuer en superposant le point de référence (situé au milieu de la clavette dans notre exemple) au centre géométrique de liaison (cf. 2.1.4.2). Les contraintes tirées du respect des limites axiales s'expriment donc de la façon suivante :

$$g_{10} = \frac{\frac{l+a}{2} + 2d - 2j}{X - lag} - 1 \quad (2.22)$$

et

$$\begin{aligned} g_{11} &= \frac{\frac{l+a}{2} + c + r_{ep} + l_{ep}}{lad - X} - 1 \\ &= \frac{l + a + 3k - d - 2j}{2(lad - X)} - 1 \end{aligned} \quad (2.23)$$

en rappelant que X représente l'abscisse du point de référence dans le repère global.

Les contraintes de montage.

Ces contraintes ont été formulées par les équations 2.1, il nous reste donc à les adapter aux variables de notre problème :

$$g_{12} = \frac{d}{lMdag} - 1 \quad (2.24)$$

$$g_{13} = 1 - \frac{d}{lmdag} \quad (2.25)$$

$$g_{14} = \frac{d_{ep}}{lMdad} - 1 \quad (2.26)$$

$$g_{15} = 1 - \frac{d_{ep}}{lmdad} \quad (2.27)$$

Formulation complète du problème.

A partir de toutes ces contraintes fonctionnelles, nous pouvons former le jeu de contraintes de notre problème d'optimisation en adoptant la formulation du §2.2.2 :

Fonction objectif :

$$\min_{(d,l) \in R^n} f(d, l)$$

Contraintes :

$$g_1 = 1 - \frac{d}{\sqrt[3]{\frac{16 C \cdot K_{ts} \cdot secu}{\pi \cdot R_{pg}}}} \leq 0 \quad \text{d'après 2.10}$$

$$g_2 = 1 - \frac{l \cdot (d - h_a) \cdot h_a \cdot p_{adm_arbre}}{2 C \cdot secu} \leq 0 \quad \text{d'après 2.13}$$

$$g_3 = 1 - \frac{l \cdot (d - h_a) \cdot h_a \cdot p_{adm_clav}}{2 C \cdot secu} \leq 0 \quad \text{d'après 2.14}$$

$$g_4 = 1 - \frac{l \cdot (d + h_m) \cdot h_m \cdot p_{adm_clav}}{2 C \cdot secu} \leq 0 \quad \text{d'après 2.15}$$

$$g_5 = 1 - \frac{l \cdot (d + h_m) \cdot h_m \cdot p_{adm_moy}}{2 C \cdot secu} \leq 0 \quad \text{d'après 2.16}$$

$$g_6 = \frac{l}{1,5 d} - 1 \leq 0 \quad \text{d'après 2.18}$$

$$g_7 = 1 - \frac{l}{1,5 a} \leq 0 \quad \text{d'après 2.19}$$

$$g_8 = \frac{\sqrt{\left(\frac{32 M f k_{tf}}{\pi d^3}\right)^2 + 3\left(\frac{16 M t k_{tt}}{\pi d^3}\right)^2}}{\sigma_{adm}} - 1 \leq 0 \quad \text{d'après 2.20}$$

$$g_9 = \frac{D_{moy}}{D_{max}} - 1 \leq 0 \quad \text{d'après 2.21}$$

$$g_{10} = \frac{\frac{l+a}{2} + c + l_{ent}}{X - lag} - 1 \leq 0 \quad \text{d'après 2.22}$$

$$g_{11} = \frac{l + a + 3k - d - 2j}{2(lad - X)} - 1 \leq 0 \quad \text{d'après 2.23}$$

$$g_{12} = \frac{d}{lMdag} - 1 \leq 0 \quad \text{d'après 2.24}$$

$$g_{13} = 1 - \frac{d}{lmdag} \leq 0 \quad \text{d'après 2.25}$$

$$g_{14} = \frac{d_{ep}}{lMdad} - 1 \leq 0 \quad \text{d'après 2.26}$$

$$g_{15} = 1 - \frac{d_{ep}}{lmdad} \leq 0 \quad \text{d'après 2.27}$$

Dans la présentation de la méthode de résolution que nous proposons ci-après, nous nous limitons volontairement aux contraintes g_1 , g_2 , g_6 , g_7 , g_{10} , g_{12} et g_{13} pour ne pas surcharger les figures. Un trop grand nombre de courbes rendrait les diagrammes illisibles. Bien sûr, la totalité des contraintes a été considérée dans l'algorithme de calcul de *Sicam*.

2.3.1.3 Résolution du problème par la méthode de Johnson.

La réduction du nombre de variables nous a permis de limiter à deux le nombre de variables de conception. En conséquence, nous pouvons adopter une représentation graphique du problème d'optimisation. Les variables de conception sont d (diamètre de l'arbre dans la section rainurée) et l (longueur utile de la clavette). Notons que certains paramètres dépendent de façon discrète du diamètre d (dimensions de la section de clavette qui prennent une valeur constante par plage de diamètres d'arbre). Ainsi certaines fonctions subissent des discontinuités au passage d'une plage de diamètres à la suivante. La figure 2.29 représente une de ces plages pour lesquelles les dimensions des clavettes prennent une valeur constante, et par conséquent, engendrent des contraintes continues sur ce domaine. Au sein d'une plage, les paramètres étant constants, seules

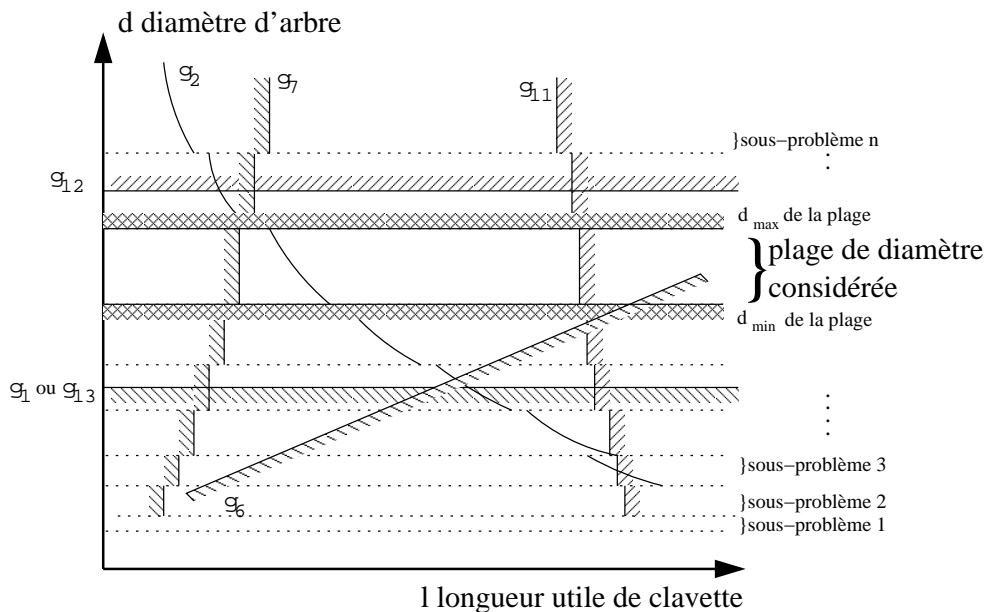


FIG. 2.29 – *Discretisation du domaine des solutions.*

subsistent les variables de conception d et l qui sont continues sur ce domaine. Nous pouvons alors formuler, sur cette plage de diamètres, un problème d'optimisation comportant des contraintes continues. Cette opération revient à découper le problème d'optimisation initial en une série de problèmes identiques — la formulation ne change pas, seules les valeurs des paramètres sont modifiées — définis dans des domaines contigus (cf. figure 2.29). Nous avons donc choisi de diviser l'espace des solutions en plages de diamètres représentées sur la figure 2.29 par un « d_{max} de la plage » et un « d_{min} de la plage ». Grâce à cet artifice, nous avons à résoudre autant de problèmes que de plages de diamètres existantes (n sous-problèmes). Les diamètres extrêmes de ces plages deviennent des contraintes à part entière qui s'ajoutent à celles déjà existantes. Ce découpage peut entraîner une désactivation de certaines contraintes qui n'ont pas d'existence dans la plage considérée (c-à-d dont la fonction ne traverse pas la plage ainsi définie). C'est notamment le cas des contraintes portant sur les diamètres maximum et minimum qui peuvent être supplantés par les valeurs limites des plages de diamètres. Nous évitons ainsi l'étude de certaines plages

de diamètres pour lesquelles une contrainte est systématiquement violée. Ce découpage nous ramène à la représentation simplifiée des sous-problèmes représentée sur la figure 2.30. Nous

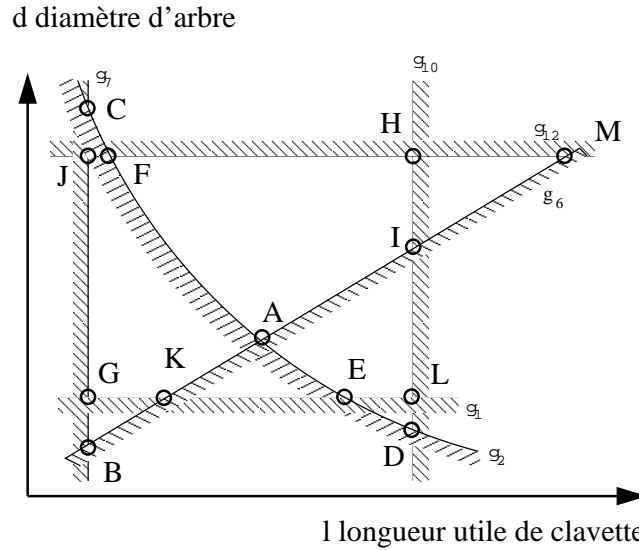


FIG. 2.30 – Représentation graphique du domaine des solutions.

appliquons la méthode de Johnson pour identifier les points représentant les couples susceptibles d'être la solution du problème ([Joh78] et [Joh80]). Ceux-ci sont repérés sur le *diagramme de variation* (figure 2.30) par des lettres. A chaque intersection de contraintes correspond un point potentiellement solution. Parmi ces points, nous ne conservons que ceux susceptibles de fournir une solution optimale selon les critères que nous avons retenus. Pour chaque *configuration du problème*, c'est-à-dire les positions relatives des contraintes, nous avons préalablement déterminé quel était le point solution. L'algorithme balaye ces quelques points pour lesquels il calcule la valeur de l'objectif, en déduit le point solution au sens du critère retenu, d'où le couple (d, l) optimal. Par exemple, le point A correspond à une solution où les contraintes g_2 et g_6 sont simultanément actives. En combinant donc les équations 2.13 et 2.15 nous obtenons les variables de conception :

$$l = \sqrt{\frac{3.C.secu}{h_a \cdot p_{adm_arbre}}} \quad \text{et} \quad d = \frac{2}{3} \sqrt{\frac{3.C.secu}{h_a \cdot p_{adm_arbre}}} \quad (2.28)$$

Les valeurs des couples solutions correspondant aux autres points sont calculés de la même façon.

Considérons deux des critères d'optimisation que nous avons présenté (cf. §2.2.2), à savoir l'optimisation en diamètre et en longueur. L'expression de l'objectif

$$f(x) = Dmoy \quad \text{soit} \quad f(x) = 5k - 4d \quad \text{d'après 2.6}$$

révèle la seule influence de la variable d sur ce critère (on rappelle que k est un paramètre discret de d). Minimiser $Dmoy$ revient donc à minimiser d .

Exprimons maintenant le critère de longueur minimale :

$$f(x) = e_g + e_d \quad \text{avec} \quad \begin{cases} e_g = \frac{l+a}{2} + c + l_{ent} \\ e_d = \frac{l+a}{2} + c + r_{ep} + l_{ep} \end{cases}$$

soit d'après les équations 2.7, 2.2, 2.4 et 2.5

$$f(x) = l + a + 2(d - j) + (d - j) + \frac{(k - d)}{2} + (k - d)$$

or nous savons que les termes $(d - j)$, $(k - d)$ et le paramètre a dépendent de façon discrète de d . Ceci signifie que pour une plage de diamètres donnée, ces valeurs sont constantes et ne dépendent plus des variations de d . En conséquence, la fonction objectif devient une fonction de l'unique variable l , et minimiser $(e_g + e_d)$ revient alors à minimiser l (pour chaque plage de diamètres).

Nous pouvons donc représenter les points solutions sur un diagramme dont les axes portent les variables l et d . Ces points seront recherchés dans le sens de l'axe d décroissant et l décroissant, respectivement pour la minimisation de D_{moy} (encombrement radial) et $e_g + e_d$ (encombrement axial).

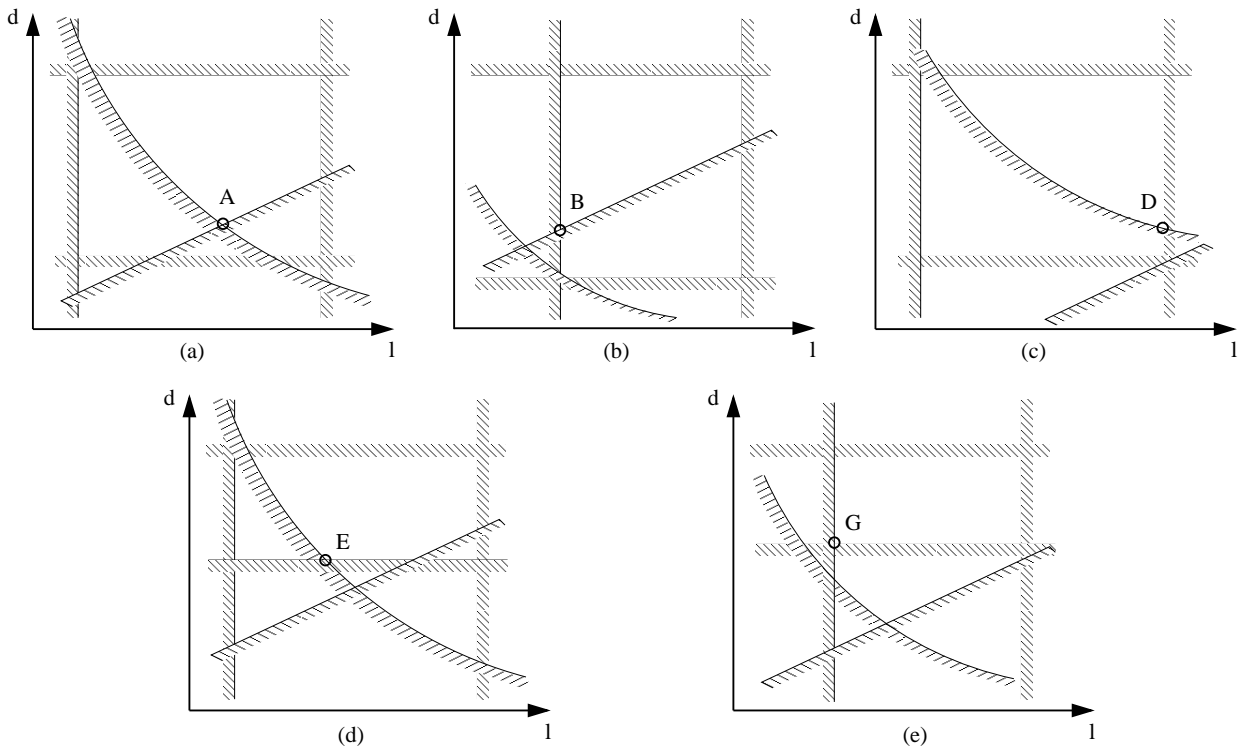


FIG. 2.31 – Couples solutions pour une minimisation du diamètre.

La figure 2.31 représente les configurations possibles du problème et les points solutions associés pour une minimisation du diamètre de l'arbre. Les points A, B, D, E et G sont donc examinés par l'algorithme quand ce critère est retenu. La figure 2.32 représente les configurations et points solutions pour une minimisation de la longueur de clavetage. On constate qu'ainsi, certains points ne sont jamais retenus (H à M). Dans certaines configurations nous obtenons non pas un, mais une infinité¹² de points solutions. Dans la configuration de la figure 2.32.c (cas d'une minimisation de la longueur) par exemple, le segment (JG) représentant la contrainte g_7 constitue un ensemble de points solutions (cette contrainte correspond à la longueur minimale de la clavette exprimée en fonction de sa largeur). Nous avons donc choisi d'appliquer un second critère d'optimisation pour départager les candidats de cet ensemble de points. Nous sélectionnons le point qui donne la longueur la plus faible *et* le diamètre le plus faible (On démontre facilement qu'il s'agit du point G). Ce cas de figure, où le concepteur doit faire face à une multitude de solutions équivalentes, est strictement dépendant de la nature des contraintes et de la fonction objectif. Ce point G est donc aussi une solution optimale dans le cas d'une optimisation

12. En réalité, nous obtenons un nombre fini de solution dont la grandeur est proportionnelle à la précision recherchée du paramètre de conception.

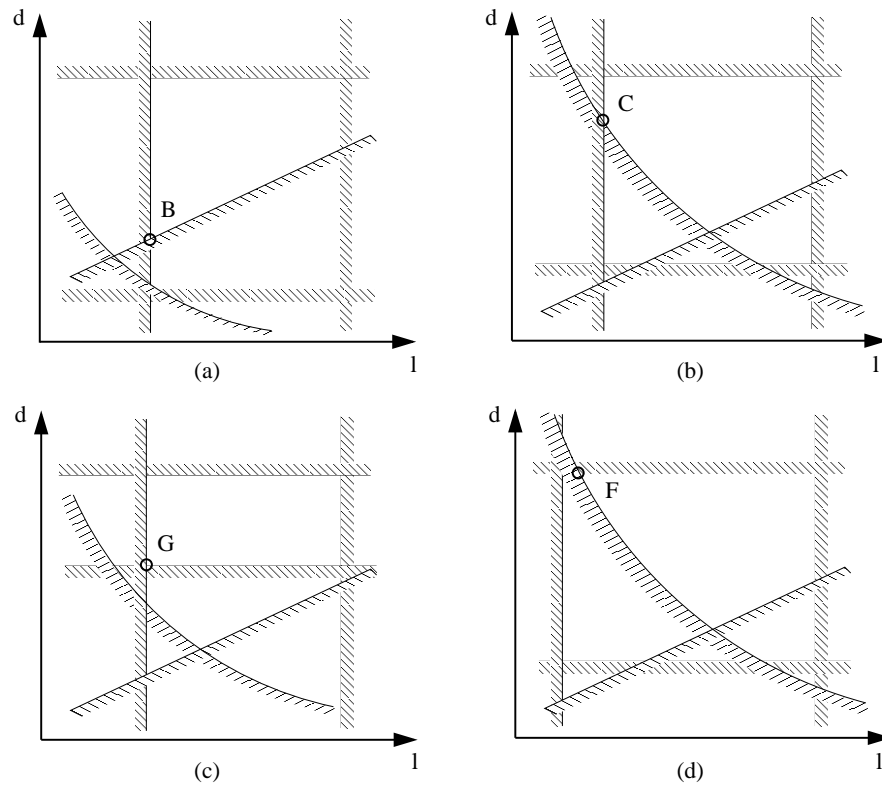


FIG. 2.32 – Couples solutions pour une minimisation de la longueur de clavetage.

du diamètre, et préférable à toute autre solution contenue sur la frontière constituée dans ce cas par la contrainte g_1 (cf. figure 2.31.e). Ainsi, certains points fournissent un couple solution optimal pour plusieurs critères d'optimisation.

Une fois les valeurs des variables de conception calculées, nous les reportons dans les relations complémentaires pour en déduire les valeurs des paramètres dimensionnels. Nous pouvons alors procéder à la mise à jour des dimensions du modèle 3D.

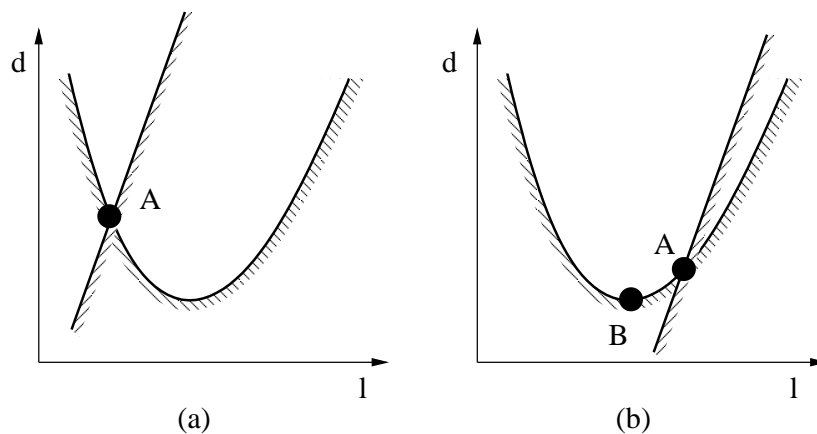


FIG. 2.33 – Cas d'une fonction contrainte non monotone.

Il est nécessaire de préciser le cas où il est fait usage de fonctions non monotones. Elles obligent à une étude supplémentaire de leur variation. Si nous examinons la figure 2.33, nous constatons que le point A défini par l'intersection des deux contraintes représentées, ne constitue une solution à la minimalisation de la variable d que dans le cas (a). A l'inverse, le cas (b)

donne une autre solution, le point B. Nous devons donc considérer que nous avons là deux configurations différentes, introduites par la non-monotonie d'une des contraintes. L'algorithme doit bien entendu tenir compte de cette éventualité lorsque des fonctions non-montones sont mises en jeu.

La méthode de Johnson présente plusieurs avantages. La représentation graphique permet de détecter des problèmes inconsistants où le domaine d'existence de solutions est un ensemble vide. Le concepteur peut alors, à partir de l'analyse du tracé, en déduire des actions correctives à mener sur les valeurs limites pour « ouvrir » ce domaine. Elle permet également de déterminer les contraintes influentes sur l'obtention d'une solution. La programmation algorithmique assure, si une solution existe, de toujours trouver le point solution associé à une configuration donnée. Le plus gros du travail, à savoir l'élaboration de l'algorithme, n'est effectué qu'une fois. Son exécution fait appel à des calculs simples et des tests de comparaison, ce qui laisse envisager son introduction dans un processus itératif.

2.3.1.4 Recherche par énumération.

Désireux de valider notre méthode d'optimisation, nous avons utilisé une méthode de recherche par discrétisation et énumération. Notre objectif était de vérifier que la méthode algorithmique, présentée dans le paragraphe précédent, avait produit de bons résultats, mais aussi qu'elle était encore compétitive, bien que datant des années soixante-dix. En effet, le but clairement avoué de ce genre de méthode était de diminuer intelligemment le nombre de calculs nécessaires à la recherche de solutions à une époque où le calcul informatique était coûteux. Cela se justifiait si, malgré un gros travail de préparation, on pouvait ensuite élaborer un algorithme qui, lors de tâches répétitives comme le dimensionnement de composants usuels, pouvait être aisément mis en œuvre.

Nous avons choisi de discrétiser la variable d en utilisant un pas suffisamment petit pour approcher une solution optimale. La valeur du millimètre a paru acceptable d'un point de vue mécanique, la taille de la plage de solution restant raisonnable (trois cents valeurs à tester pour un diamètre d'arbre variant de 1 à 300 mm). Nous avons quand même mis en place certains mécanismes de limitation afin de ne pas tester inutilement toutes les valeurs. La première manœuvre vise à appliquer les limites sur le diamètre (d_{min} et d_{max}) puis éventuellement à opérer une propagation de contraintes à partir des autres valeurs limites de contraintes. Les résultats sont éloquentes puisque les calculs fournissent une réponse en un temps comparable (moins d'une seconde) à celui de la méthode algorithmique. Les résultats obtenus sont identiques (au degré de précision dimensionnelle près) à ceux obtenus par la méthode précédente. Cela confirme le bon fonctionnement de l'algorithme de Johnson (en termes de fiabilité dans la recherche d'une solution optimale) ainsi que sa performance (en temps d'exécution).

Une analyse rapide de ces résultats s'impose. Fondamentalement, la méthode énumérative requiert plus de calculs que l'algorithme établi précédemment. On doit cependant noter que l'éternel point faible de l'informatique est le transfert de données vers la mémoire de masse, et non le calcul. Ceci signifie que les tâches les plus lentes sont l'ouverture, la fermeture de fichiers ainsi que l'affichage des résultats. En ce sens, et toutes proportions gardées, une augmentation sensible du nombre de calculs n'entraîne pas forcément un allongement sensible du temps de résolution. Nous distinguons le *temps de calcul* du *temps de résolution* car sur le type de calcul que nous effectuons, le temps de calcul est souvent négligeable devant la durée totale de lancement du programme, car infinitésimal. Bien entendu, une multiplication des lancements de programme rendrait notre propos inopportun. Mais ces séquences de calculs sont majoritairement suivies d'une analyse critique du concepteur qui nécessitent beaucoup plus de temps. Enfin, gardons

à l'esprit l'ordre de grandeur de ces résultats —moins d'une seconde— qui dans le cadre d'un processus engageant un raisonnement humain, réduit significativement son importance. Dans le cadre d'un processus entièrement automatisé, la question mériterait d'être reposée. La *phase de synthèse coordonnée* sera l'occasion de revenir sur ce point.

2.3.2 Choix des méthodes d'optimisation.

De nombreuses liaisons mécaniques peuvent donner lieu à un problème d'optimisation simple en paramétrant judicieusement la géométrie du DTLM. Cela peut être facilité par l'introduction de règles empiriques, ou issues d'études, dans l'association des variables. De la finesse des modèles employés dépend la pérennité de la solution proposée. Ainsi, si les modèles sont trop grossiers, ou les coefficients de sécurité adoptés trop importants, on court le risque de s'éloigner de la solution optimale. Une approche par objets tels que les DTLM permet d'utiliser, en fonction du degré de fidélité recherché, des modèles ayant des définitions géométriques et comportementales plus ou moins fines. L'abstraction et l'encapsulation facilitent la substitution d'objets de modélisations plus ou moins fines, sans modifier la structure de données existante.

Nous aurions pu tester d'autres méthodes de recherche de solutions optimales. Les méthodes stochastiques, actuellement en vogue, semblent parfaitement adaptées aux types de problèmes que nous rencontrons (variables mixtes). Au rang de celles-ci nous pouvons citer les méthodes évolutionnistes (algorithmes génétiques, . . .) ou purement aléatoires (Monte-Carlo). Nous pensons cependant qu'il n'est pas opportun d'utiliser de telles méthodes dans le cadre du dimensionnement des DTLM pour les raisons suivantes :

- Le nombre de combinaisons engendré par la faible taille des problèmes ne justifie que rarement l'utilisation de méthodes stochastiques. L'emploi de règles visant à diminuer la combinatoire s'avère souvent plus efficace sur une méthode de recherche directe que l'exploration aléatoire complète du domaine des solutions. Bien souvent, pour les problèmes de taille faible, la méthode stochastique s'apparente, de facto, à une énumération complète avec l'inconvénient de pouvoir passer plusieurs fois en revue les mêmes solutions !
- Les méthodes stochastiques sont souvent d'une *mise en œuvre complexe*. Elles sont sensibles au point (ou population) de départ et au codage des données du problème. Le réglage est souvent long et peut réclamer des modifications si les paramètres viennent à évoluer (nombre de variables, codage. . .).
- La gestion des contraintes est rudimentaire. Elle recourt à des méthodes de pénalités dont le choix et la mise au point restent délicats.
- Les temps de calcul des méthodes stochastiques sont souvent supérieurs à ceux des méthodes traditionnelles. Ceci est dû au grand nombre d'évaluations effectuées par l'algorithme.
- L'introduction de méthodes stochastiques dans un processus de traitement automatique n'est pas aisé. Ces dernières nécessitent souvent des phases de réglage de paramètres, ainsi que de codage des données qui alourdissent l'ensemble de la procédure.
- Plus fondamentalement, les problèmes que nous rencontrons font appel à des modélisations relativement simples qui sont issues de comportements physiques avérés. Il est donc aisé pour le concepteur de deviner la prépondérance de certains paramètres et leur tendance à évoluer dans le bon sens. Le processus aléatoire *ne profite pas de cette connaissance experte*, ce qui constitue une perte de temps non négligeable. Enfin, le processus de recherche

stochastique bénéficie par définition d'une *lisibilité faible* pour le concepteur. L'historique de recherche est souvent difficile à exploiter voire inexistant. Dans tous les cas, la recherche d'une solution ne s'appuie ni ne met en évidence aucune «règle» que le concepteur pourrait mettre ultérieurement à profit (par exemple, tel paramètre ou contrainte est souvent prépondérant dans tel cas d'optimisation).

Pour toutes ces raisons, nous pensons que ces méthodes sont à réserver à des problèmes plus complexes pour lesquels peu d'heuristiques existent.

Les méthodes classiques de gradient peuvent être utilisées dans des cas où le problème ne fait apparaître que des variables continues et des fonctions dérivables sur leur domaine d'existence. C'est notamment le cas pour la liaison par frettage où les paramètres de conception —longueur, diamètre de frettage et éventuellement ajustement— répondent à ces conditions.

La structure de données associées au traitement des DTLM autorise l'utilisation de méthodes d'optimisation propres à chacun d'eux. Il nous est ainsi plus facile de sélectionner la méthode qui s'adaptera le mieux à la nature du problème rencontré (nombre et nature des variables et des contraintes).

La méthode algorithmique de Johnson est intéressante, comme nous l'avons montré dans le cas d'étude présenté ci-avant. Il existe pourtant des limitations à son utilisation. Un point critique est la dimension du problème d'optimisation : le nombre de composantes du vecteur des variables de conception et le nombre de fonctions contraintes. Si la réduction du nombre de variables s'avère difficile et que, de ce fait, ce nombre est supérieur à deux, il est alors nécessaire de tracer des diagrammes à n dimensions, n représentant le nombre de variables de conception, ce qui n'est pas aisé. Par ailleurs, un trop grand nombre de contraintes entraîne une surcharge graphique des diagrammes de variation et par conséquent une mauvaise lisibilité.

Lorsque nous devons traiter des problèmes faisant intervenir des variables discrètes ou entières, la méthode algorithmique que nous avons illustrée par l'exemple de la liaison complète, doit être réaménagée. Plusieurs solutions sont envisageables : l'une d'elle consiste à fixer les valeurs des variables discrètes une par une. On doit alors lancer l'algorithme autant de fois qu'il existe de combinaisons de valeurs discrètes. Une autre méthode consiste à rendre ces variables continues, soit en découpant le domaine d'étude comme nous l'avons fait pour la liaison clavetée, soit en approximant leurs valeurs à l'aide de polynômes de degré plus ou moins élevé, au prix d'une certaine erreur. Une fois les points solutions déterminés dans le domaine continu, il nous reste encore à rechercher les points solutions voisins ayant des valeurs discrètes de ces variables, afin d'obtenir la solution.

L'utilisation d'une méthode d'énumération n'est pas sans intérêt lorsqu'on aborde la création de DTLM ne présentant que peu (ou pas) de variables continues comme la liaison par cannelure (module de denture et nombre de dents) ou la liaison par roulement (composants catalogue). Les progrès de l'informatique permettent actuellement de parcourir très rapidement une base de données à la recherche du composant fournissant la solution optimale en s'affranchissant des contraintes liées aux méthodes d'optimisation traditionnelles. Son traitement algorithmique simple présente également l'avantage d'une maintenance aisée du code : l'introduction ou la suppression de contraintes liées, par exemple, à une évolution du degré de précision de la modélisation n'entraîne qu'un minimum de modifications sur le code informatique. La méthode énumérative possède aussi l'avantage de pouvoir fournir *l'ensemble* des solutions réalisables et non la seule solution optimale. Cette méthode ne doit donc pas être négligée, et mérite d'être considérée dès lors que d'autres atteignent leurs limites.

2.3.3 Ensemble des DTLM développés.

Nous n'avons présenté dans le paragraphe précédent que le développement d'un type de DTLM (liaison clavetée forme A / entretoise / épaulement) appartenant à une famille (arbre/moyeu). La liste de tous les DTLM déjà développés et disponibles dans *Sicam2* est donnée dans le tableau 2.1. La figure 2.34 illustre la structure de notre base de DTLM en reprenant le formalisme objet présenté en §2.2.1. Les pointillés symbolisent l'ouverture de la base à de nouvelles familles de

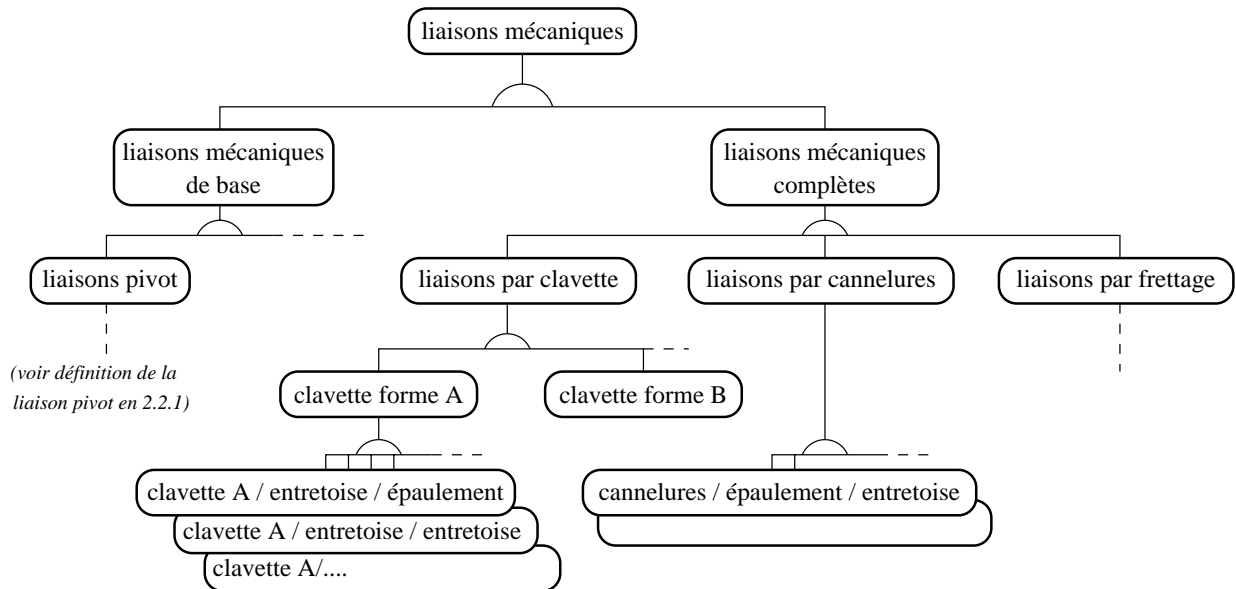


FIG. 2.34 – Structure objet des DTLM.

DTLM, voire à de nouveaux DTLM parmi les familles déjà présentes. Pour chaque catégorie¹³ nous trouvons une liste de lettres correspondant aux combinaisons d'arrêts axiaux de chaque DTLM (deux sur l'arbre, deux sur l'alésage), en formule abrégée. Les multiplicateurs entre parenthèses indiquent le nombre de combinaisons identiques d'arrêts axiaux par application d'une symétrie droite/gauche sur l'arbre et l'alésage. Ce n'est donc pas moins de *deux cent vingt-trois* DTLM qui sont mis à disposition du concepteur dans le système *Sicam2*.

Chaque famille de DTLM possède des caractéristiques propres qui ont rendu leur développement unique. Les DTLM à base de roulement ont nécessité une approche directe. En effet, le roulement est un composant standard du type « catalogue » dont les dimensions s'échelonnent selon des pas de variation fluctuants. L'utilisation de plusieurs séries du même type de roulement ne facilite pas une éventuelle linéarisation de leurs dimensions comme nous pouvons le constater sur la figure 2.35. La méthode énumérative reste donc la meilleure solution dans le cadre d'une recherche sur une base de données de taille modeste (en l'occurrence cent vingt références).

Les DTLM à base de cannelures présentent la même problématique que les liaisons à base de clavette, c'est-à-dire un problème en variables mixtes. Une partie des variables de conception est de nature discrète (diamètre et module ou nombre de dents) tandis que l'autre est continue (longueur de cannelures). Les méthodologies envisageables sont identiques à celles de la liaison par clavette. Nous avons mis en application un algorithme qui parcourt les séries de cannelures par module.

13. Ces catégories ne correspondent pas exactement au découpage de la structure de données. Elles ont pour fonction unique de simplifier la lecture du tableau.

Complète			
Clavette	forme A_ep_ci (x2)	forme A_en_ci (x2)	forme A_ep_en (x2)
	forme A_en_en		
	forme B_ep_ci (x2)	forme B_en_ci (x2)	forme B_ep_en (x2)
	forme B_en_en		
Frettage	ri_ri	ri_ep (x2)	
Cannelures	ci_en (x2)	ci_ep (x2)	en_ep (x2)
Pivot			
Radial à billes	ep_ec_ci_en (x4)	en_ec_ep_ci (x4)	ri_ri_ep_en (x4)
	ri_ri_ep_ci (x2)	ri_ri_en_en	ri_ri_en_ci (x2)
	ep_en_ri_ri (x2)	ep_en_en_ep (x4)	ep_en_ci_ep (x4)
	ep_ec_ri_ri (x2)	ep_ec_ep_ci (x4)	ep_ec_en_ep (x4)
	ep_ec_en_en (x2)	ep_ci_ri_ri (x2)	ep_ci_en_ep (x4)
	ep_ci_en_en (x2)	ep_ci_ci_ep (x4)	ep_ci_ci_en (4x)
	en_en_ri_ri	en_en_ep_en (x2)	en_en_ci_ep (x2)
	en_ec_ri_ri (x2)	en_ec_en_ep (x4)	en_ec_en_en (x2)
	en_ec_ri_ci (x2)	en_ci_ri_ri (x2)	ep_en_ci_en (x4)
	en_en_ci_en (x2)	ci_en_en_ep (x4)	ci_en_en_en (x2)
ci_en_ci_ep (x4)	ci_en_ci_en (x4)		
Radial à rouleaux	ep_ec_ci_en (x4)	en_en_ep_en (x2)	ep_en_ci_en (x4)
	en_ec_ep_ci (x4)	en_en_ci_ep (x2)	en_ec_en_ci (x4)
	ep_en_en_ep (x4)	en_ec_en_ep (x4)	en_ec_en_en (x2)
	ep_en_ci_ep (x4)	ci_en_ci_en (x4)	ep_ci_en_ep (x4)
	ep_ec_ep_ci (x4)	ci_en_ci_ep (x4)	ep_ci_en_en (x2)
	ep_ec_en_ep (x4)	ci_en_en_en (x2)	ep_ci_ci_ep (x4)
	ep_ec_en_en (x2)	ci_en_en_ep (x4)	
	ep_ci_ci_en (x4)	en_en_ci_en (x2)	
Oblique à billes	ri_en_ep_ri (x2)	ri_ep_ep_ri (x2)	ri_ci_en_ri (x2)
	ri_en_ci_ri (x2)	ri_ec_ep_ri (x2)	ri_en_en_ri (x2)
	ri_ec_en_ri (x2)	ri_ep_en_ri (x2)	
Oblique à rouleaux coniques	ri_en_ep_ri (x2)		
	ri_en_ci_ri (x2)	ri_ep_ep_ri (x2)	ri_ci_en_ri (x2)
	ri_ec_en_ri (x2)		
Paliers lisses	ri_ri_ri_ri		

abréviations utilisées : rien, circlip(anneau élastique), epaulement, ecrou et entretoise

TAB. 2.1 – Liste des DTLM.

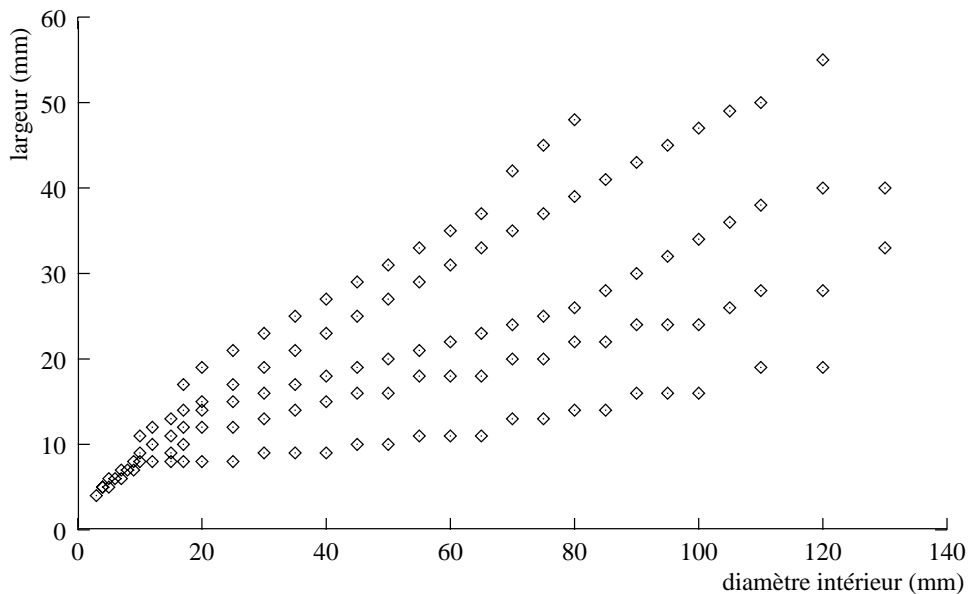


FIG. 2.35 – *Étagement des dimensions de roulement à billes radiales.*

Enfin la liaison par frettage est un problème qui ne présente que des variables continues (diamètre d'arbre et valeur de serrage) qui peut se traiter par des méthodes de gradients classiques. Seule la formulation du problème est un peu complexe, notamment à cause de la présence d'équations du troisième degré.

2.4 Conclusions.

Le DTLM doit être considéré comme un feature CAO de haut niveau. Il est composé de surfaces et volumes fonctionnels et forme une unité élémentaire de conception associée à la tâche T4 de notre organisation opérationnelle. Il assure une fonction de liaison mécanique et possède une méthode de dimensionnement optimale propre à chaque individu. Cette modularité est rendue possible grâce à l'adoption d'une approche orientée objet pour la modélisation de notre système. Les DTLM sont interchangeables entre eux au sein d'une même classe (les classes étant définies par les degrés de spécialisation de la fonction de liaison). Les méthodes de dimensionnement sont internes aux DTLM, et propres à chacun d'eux : cela permet d'utiliser les méthodes d'optimisation les mieux adaptées à la nature du problème.

Le DTLM possède un modèle 3D qui joue un rôle primordial dans l'évaluation du résultat par le concepteur. L'utilisation du dessin technique est infiniment plus riche qu'une représentation sémantique, par exemple, et renseigne le concepteur de façon bien plus intuitive que n'importe quelle autre modélisation. Le modèle 3D du système ainsi conçu est disponible pour les étapes ultérieures de la conception (synthèse avec la tâche T5 et analyse).

L'utilisation des DTLM procure de nombreux avantages. Le découpage qui mène à la définition de sous-ensembles mécaniques que sont les DTLM autorise l'emploi de méthodes d'optimisation diverses, adaptées au composant, qui ne seraient pas envisageables pour des systèmes de taille plus importante. Elle favorise aussi le dimensionnement de chaque liaison de manière indépendante, et autorise l'emploi d'objectifs d'optimisation hétérogènes.

Chapitre 3

Organisation générale de l’outil d’assistance.

Résumé

AFIN de valider le concept de DTLM, nous avons bâti une maquette logicielle d’outil d’assistance à la conception. Ce chapitre présente tout d’abord les objectifs de cet outil. Nous effectuons ensuite une présentation générale pour donner au lecteur une vue globale du fonctionnement du système. Enfin nous détaillons, à l’aide d’un exemple, la première étape d’une séquence de conception assistée par *Sicam2* qui est la phase de renseignement du cahier des charges global.

Contenu du chapitre

3.1	Objectifs de l’outil d’assistance.	82
3.1.1	Synthèse découplée.	82
3.1.2	Synthèse coordonnée.	84
3.2	Présentation générale de l’outil d’assistance.	84
3.2.1	Architecture opérationnelle de l’interface homme/machine.	84
3.2.2	Les interfaces graphiques : menus et I-DEAS®.	85
3.2.2.1	Dialogue par les menus.	86
3.2.2.2	Dialogue par I-DEAS®.	86
3.2.3	Menu principal.	86
3.3	Initialisation de la phase de conception : le cahier des charges global.	90

3.1 Objectifs de l'outil d'assistance.

L'outil d'assistance que nous avons élaboré a pour principal objectif la validation du concept de DTLM. En démontrant que l'utilisation de ce type de composant est non seulement réalisable, mais aussi adaptée au traitement informatique par le biais de la CAO, nous espérons ouvrir la voie vers une nouvelle méthode de synthèse des mécanismes.

Les DTLM sont les éléments de base d'une tâche de synthèse, elle-même sous-ensemble du processus, plus large, de synthèse des mécanismes que nous avons proposé. L'outil d'aide à la synthèse que nous présentons est une des briques logicielles de ce qui devrait être un système intégré de conception.

Notre outil d'assistance est couplé à un logiciel de CAO qu'il utilise comme outil de visualisation des modèles volumiques 3D. Nous tentons de démontrer qu'il est possible, à l'aide des logiciels de CAO commerciaux, de mettre en place une méthodologie de conception différente des approches traditionnellement proposées.

Pour décrire la fonction principale de *Sicam 2*, nous reprenons la définition de la tâche T4. Cette tâche a pour but de définir les DTLM des liaisons de guidage et des liaisons complètes pour les sous-ensembles cinématiquement équivalents. Le logiciel assiste donc le concepteur dans la définition des DTLM d'une même ligne d'arbres¹ en prenant en charge la capitalisation des données, le calcul des dimensions des DTLM et leur visualisation 3D. Les mécanismes pris en charge par *Sicam 2* sont, pour le moment, les lignes d'arbres présentant une liaison pivot décomposable en deux DTLM, repérés A et B, et deux liaisons complètes assurées par les DTLM repérés C et D. Le repérage alphabétique des DTLM au sein de chaque classe de liaison reflète la proximité de ceux-ci avec l'origine de l'axe : ainsi, le DTLM A (resp. C) sera toujours à gauche du DTLM B (resp. D).

Le principe de fonctionnement est le suivant. Le logiciel prend en charge le calcul de dimensionnement optimal de chaque DTLM. Le résultat est contenu dans un fichier propre à la liaison et peut, sur simple commande de l'opérateur, être modélisé en 3D grâce au logiciel de CAO. La figure 3.1 représente une modélisation 3D partielle d'un système étudié. Nous y distinguons un *assemblage racine*² que nous appelons aussi *super-assemblage*. Il est initialement constitué des dentures d'engrenage, dont toutes les caractéristiques sont connues (l'épaisseur sous denture a été déterminée selon les travaux d'A. Daidié [Dai96]) et qui ont été calculées au niveau supérieur. À cet assemblage seront ajoutés, au fur et à mesure de leur définition, les DTLM. Le résultat final est disponible sur le plan de travail du logiciel I-DEAS®. Il s'agit du modèle volumique 3D complet du système étudié (le *super-assemblage*) incorporant les dentures d'engrenages et les divers DTLM définis à l'aide de *Sicam 2*. Notre outil propose deux méthodes de synthèse qui pourront être assimilées à des *phases d'exploitation* différentes de notre outil. L'une est appelée *synthèse découplée* car elle ne gère pas les interactions entre DTLM dans le mécanisme étudié. L'autre est appelée *synthèse coordonnée* car elle gère, par un processus de coordination des opérations de dimensionnement, les interactions entre DTLM dans le but de fournir une solution cohérente. Ces deux méthodes sont présentées ci-dessous.

3.1.1 Synthèse découplée.

La synthèse découplée est un premier niveau d'assistance au concepteur. Elle a pour but d'aider le concepteur dans son dimensionnement des divers DTLM. Chaque DTLM est considéré

1. La ligne d'arbres, qui n'est qu'un sous-ensemble du produit conçu, constitue pour nous le « mécanisme » étudié. Nous nous y référerons donc dorénavant en tant que mécanisme.

2. Le terme *racine* désigne l'élément qui est à la base de l'arborescence de l'assemblage.

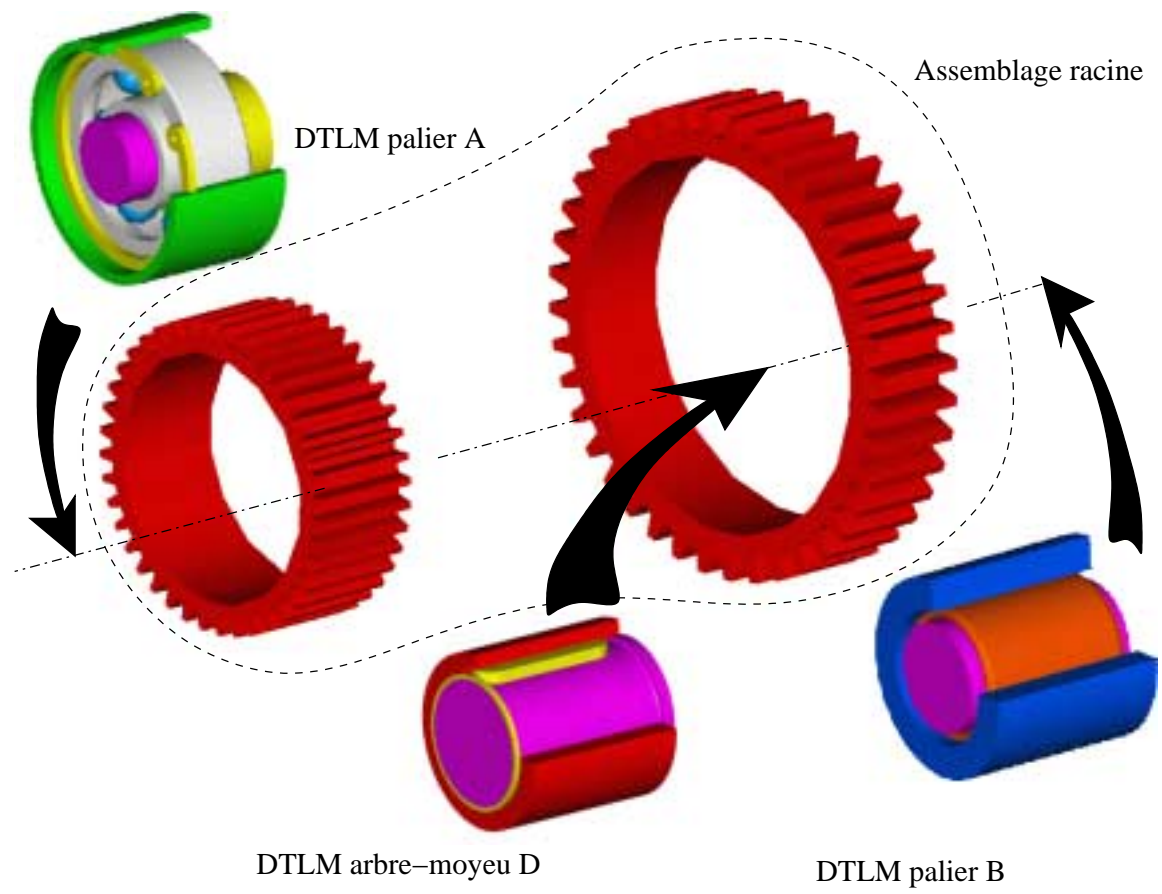


FIG. 3.1 – Exemple de composition d'un super-assemblage.

indépendamment ; il possède un objectif d'optimisation propre et le jeu de contraintes lié à chaque DTLM n'engage que des règles internes à celui-ci (résistance des matériaux, durée de vie, etc.). L'outil fournit une assistance pratique au concepteur en automatisant le processus de dimensionnement optimal des DTLM et en proposant éventuellement une modélisation 3D sous I-DEAS®. Il permet aussi au concepteur de tester différents choix concurrents et de prendre connaissance rapidement des ordres de grandeur des dimensions de chaque solution. Toutes les interactions et les conflits pouvant intervenir entre les différents DTLM ne sont pas pris en charge par l'outil. Le concepteur peut, *manuellement*, contraindre les DTLM pour guider le dimensionnement de chacun vers une solution globale qu'il veut cohérente. Il dispose pour cela de mécanismes l'autorisant à imposer des valeurs limites aux encombrements de chaque DTLM. Il devra cependant effectuer lui-même le choix des espaces alloués à chaque DTLM, ainsi que les priorités qu'il accorde à l'encombrement de chacun d'eux. Notre outil offre la possibilité de modifier à volonté les choix relatifs à chacun des DTLM et d'en visualiser, de façon quasi instantanée, la concrétisation. Il favorise ainsi la créativité du concepteur en simplifiant les démarches de modification et en soulageant le concepteur des tâches fastidieuses et répétitives de calcul.

3.1.2 Synthèse coordonnée.

La synthèse coordonnée fournit un second niveau d'assistance au concepteur. Elle est consécutive à la phase de synthèse découplée et ne remet pas en cause les choix précédemment effectués. Elle a pour but de générer, de façon autonome, une solution cohérente de mécanisme à partir des cahiers des charges fixés durant la synthèse découplée. Elle doit pour cela prendre en compte les contraintes de liaison entre les DTLM, à savoir les interférences géométriques axiales ainsi que les contraintes de montage. Il est fait appel à un processus de coordination des procédures de dimensionnement optimal des DTLM pour aboutir à une solution globale.

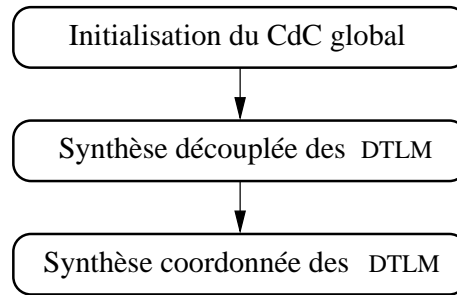
3.2 Présentation générale de l'outil d'assistance.

Ce paragraphe s'attache à expliciter la structure générale de *Sicam* 2. Dans la suite de cet exposé, nous confondrons les notions de concepteur (personne en charge d'une tâche de conception) et l'opérateur (la personne préposée à l'utilisation du logiciel) puisque, dans notre démarche, ils sont une seule et même entité.

3.2.1 Architecture opérationnelle de l'interface homme/machine.

Une certaine cohérence dans la synchronisation des actions est impérative. Nous distinguons trois phases dans la chronologie opérationnelle de *Sicam* 2 qui sont visibles sur la figure 3.2.

- La première phase est une *phase d'initialisation* du problème de conception. Elle consiste à renseigner le *cahier des charges global* (ou CdC global). L'opérateur fournit à l'outil les informations de conception issues des niveaux précédents de définition. Le cahier des charges global est ainsi dénommé car les informations qu'il contient concernent l'intégralité du mécanisme étudié et donc l'ensemble des DTLM. Ces attributs du mécanisme sont transférés aux DTLM par les méthodes d'héritage vues en §2.2.1.
- La seconde phase est la phase que nous avons nommée «synthèse découplée». Elle consiste à déterminer, pour chaque liaison, les dimensions optimales des DTLM envisagés par le

FIG. 3.2 – Chronologie opérationnelle de *Sicam2*.

concepteur. Ce dimensionnement est effectué à partir du CdC global, d'un cahier des charges local, encore appelé *cahier des charges liaison* (CdC liaison), et selon un critère d'optimisation qui peut être différent pour chacun des DTLM.

- La troisième et dernière phase est consacrée à la «synthèse coordonnée». Elle s'appuie sur les données des phases précédentes et tente de générer une solution globale cohérente en prenant en compte les contraintes de couplage entre DTLM.

Chaque phase fait appel à un certain nombre de fenêtres de l'interface graphique que nous détaillons ci-dessous.

3.2.2 Les interfaces graphiques : menus et I-DEAS®.

Les deux éléments de dialogue entre le logiciel et l'opérateur sont la structure de menus et le logiciel I-DEAS® (cf. figure 3.3). La structure de menus est gérée par le module principal qui invoque les divers sous-modules selon les besoins de l'opérateur. La figure 3.3 représente l'acti-

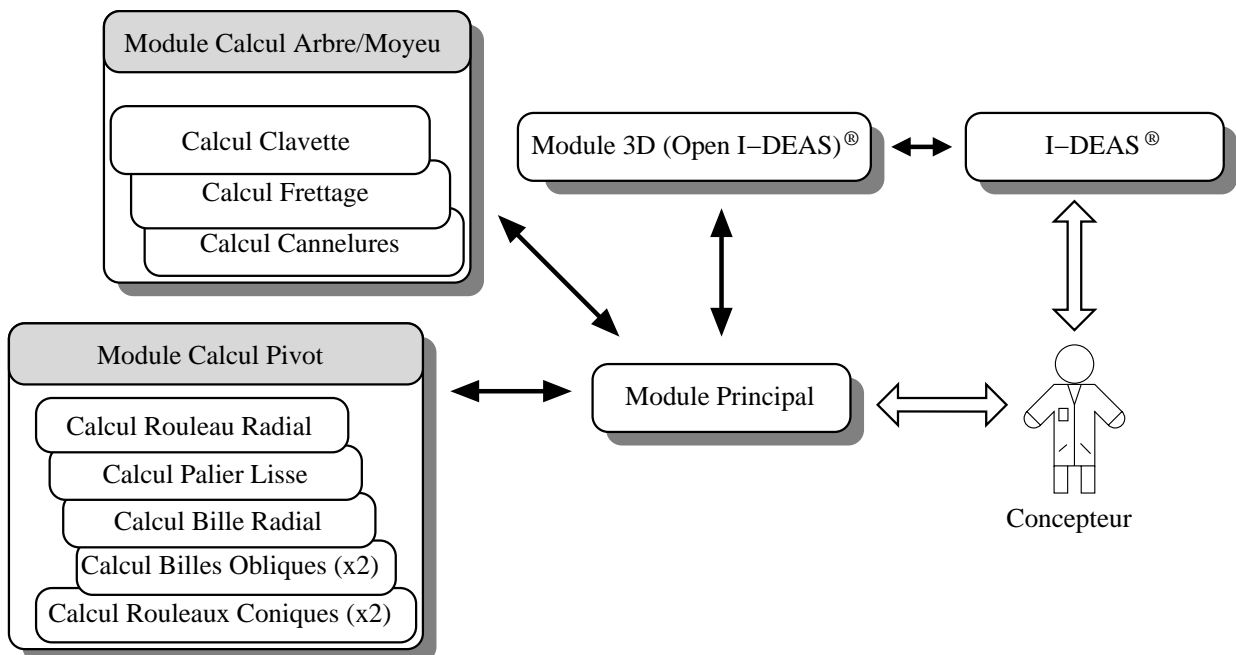


FIG. 3.3 – Actigramme du système.

gramme de *Sicam2*. Nous constatons que l'opérateur est au cœur du système de conception.

Les seuls éléments avec lesquels il communique sont le module principal et le logiciel de CAO I-DEAS®. Les flux de données sont différents selon que le concepteur dialogue avec la structure de menus ou avec I-DEAS®. Nous les détaillons ci-après.

3.2.2.1 Dialogue par les menus.

La figure 3.4 représente l'arborescence des menus de *Sicam* 2. Nous détaillerons progressivement leur contenu. Intéressons-nous pour le moment à sa structure. On remarque l'extrême simplicité de cette arborescence qui ne compte que trois niveaux de hiérarchie (une arborescence trop profonde est souvent nuisible à l'utilisation). La présentation en cascade simule la progression du concepteur dans l'arbre de définition du système. Le sommet de cette arborescence est constitué du *menu principal*. Il sert de point de départ vers les différentes étapes de la synthèse des DTLM. Les menus peuvent être regroupés selon les trois phases d'exploitation de *Sicam* 2 :

- le menu du CdC global appartenant à la phase d'initialisation,
- les menus des CdC liaison appartenant à la phase de synthèse découlée,
- le menu de coordination appartenant à la phase de synthèse coordonnée.

L'opérateur naviguera au travers de ces fenêtres pendant la séquence de conception assistée.

3.2.2.2 Dialogue par I-DEAS®.

Afin de visualiser les modèles 3D des DTLM et du mécanisme, nous avons utilisé un logiciel commercial, I-DEAS® de SDRC. L'interface graphique de ce logiciel est représentée par une copie d'écran sur la figure 3.5. Elle est d'un format relativement classique. Elle se compose de quatre fenêtres qui ont respectivement pour fonction : l'impression de messages textuels à destination de l'utilisateur, la saisie de données alphanumériques, la saisie de commandes par menus d'icônes et l'affichage graphique des objets modélisés (cette dernière occupant les trois-quarts de l'espace-écran). Nous avons délibérément choisi de ne pas modifier l'interface du logiciel pour des raisons d'ergonomie. En conséquence, rien ne différencie en apparence une version d'I-DEAS® classique d'une version couplée à *Sicam* 2.

Le logiciel I-DEAS® présente deux types de fonctionnement, selon qu'il est utilisé couplé ou non avec *Sicam* 2. Lorsque nous l'utilisons en couplage, il a pour fonction principale de visualiser les modèles 3D du mécanisme étudié. A l'issue d'une séquence de conception avec *Sicam* 2, l'opérateur a la possibilité, en utilisant les fonctionnalités d'I-DEAS®, de sauvegarder les modèles 3D ainsi créés. Il peut ensuite les rappeler et les réutiliser ultérieurement, indépendamment de notre outil d'assistance : c'est le second mode de fonctionnement. Cela présente un intérêt certain dans les étapes suivantes du processus de synthèse (tâche T5 de définition des pièces) ainsi que dans la phase d'analyse (éléments finis par exemple) où le concepteur profite des modèles créés à l'aide de *Sicam* 2.

3.2.3 Menu principal.

Au lancement du système *Sicam* 2, la fenêtre du menu principal apparaît (fig. 3.6). Détaillons-en la composition.

La fenêtre se décompose verticalement en quatre zones que nous décrivons de haut en bas.

- La première zone contient la barre de menu qui propose les actions concernant la gestion du projet (« Projet ») ou des options de visualisation autres que la visualisation 3D des

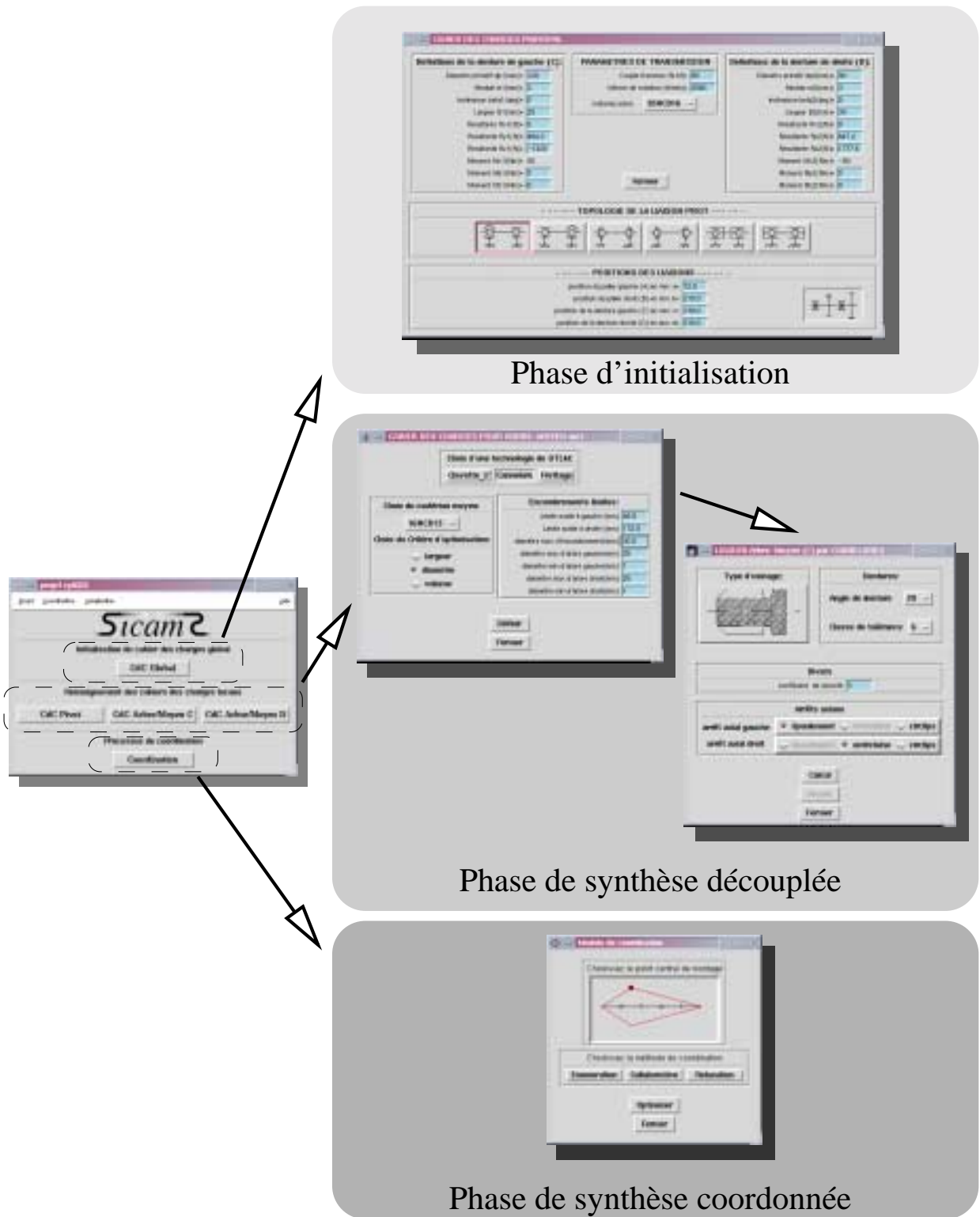


FIG. 3.4 – Arborescence des fenêtres.

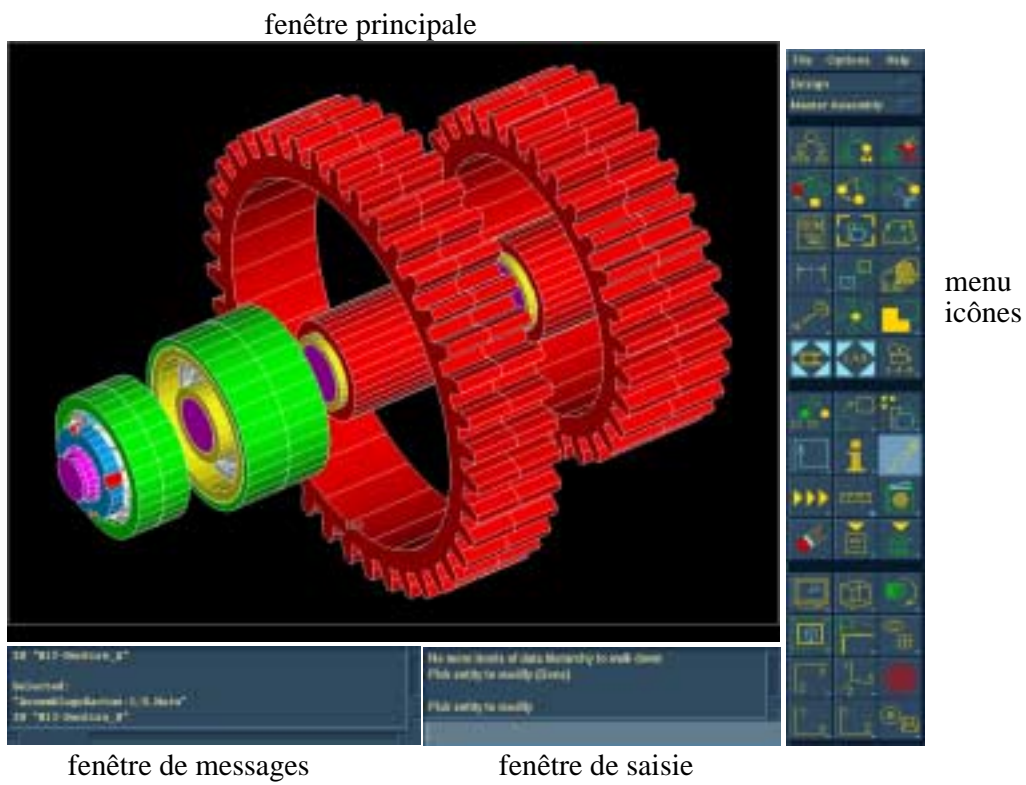


FIG. 3.5 – Interface graphique du logiciel I-DEAS®.

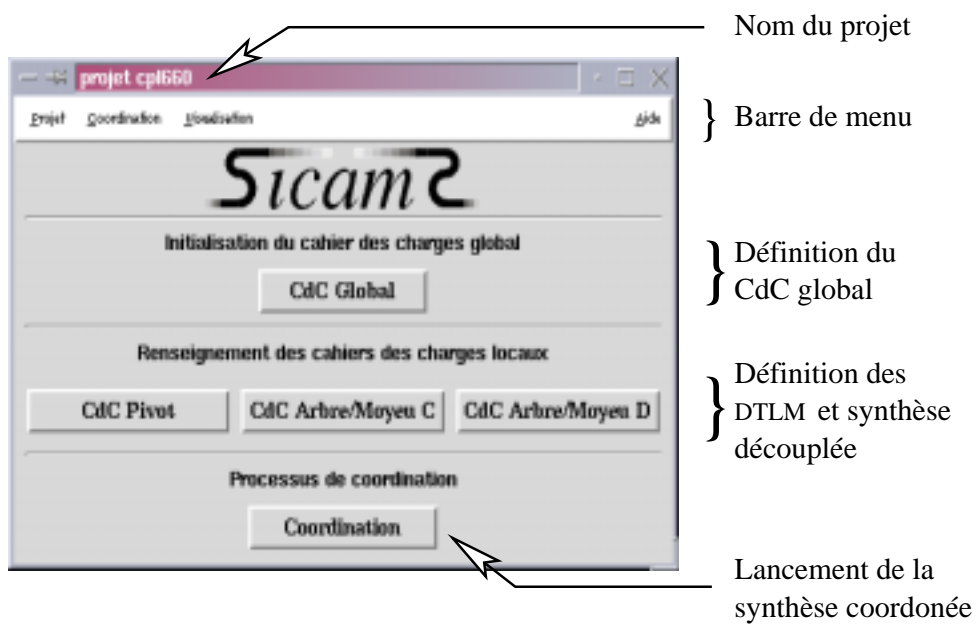


FIG. 3.6 – Menu principal.

DTLM. Les projets sont gérés par un mécanisme de fichiers dont les noms indiquent leur appartenance au projet ainsi que le type de données contenues. Les opérations sur les projets sont les opérations courantes *d'ouverture*, de *sauvegarde*, de *sauvegarde sous un nom différent*, de *création* d'un nouveau projet, de *destruction* ou de *purge* (celle-ci éliminant les fichiers inutiles du répertoire de projet). Ces opérations, concernant principalement la gestion des fichiers du projet, sont complétées par des actions sur les données du projet en cours qui sont la *restauration* des données du projet (dans l'état de la dernière sauvegarde), et la *réinitialisation* —retour aux valeurs par défaut— de ces mêmes données.

- La deuxième zone contient un bouton poussoir qui ouvre la fenêtre du cahier des charges global. Lorsque le concepteur démarre un nouveau projet, il doit obligatoirement passer par cette étape en premier lieu. Il renseigne le CdC global comme nous le verrons dans la section 3.3. Cette zone correspond aux actions de la phase d'initialisation.
- La troisième zone est consacrée au lancement des menus de définition des DTLM. Elle représente donc la phase de synthèse découplée. Elle comprend trois boutons, du même type que celui employé pour le CdC global, qui activent respectivement le menu de la liaison pivot (DTLM A et B), et ceux des deux liaisons complètes arbre/moyeu (DTLM C et D).
- La dernière zone propose un bouton de lancement de la procédure de coordination qui sera décrite au chapitre 5. Elle représente la phase de synthèse coordonnée qui est l'ultime niveau d'assistance de *Sicam2*.

Ce menu est toujours présent à l'écran. Il constitue la plaque tournante de *Sicam2* dont il est la base de l'arborescence opérationnelle. Le concepteur peut, à tout moment, activer un ou plusieurs sous-menus de définition des DTLM. Le bouton est alors désactivé pour indiquer qu'un sous-menu est déjà ouvert. Il ne redevient disponible qu'à la fermeture du sous-menu actif.

Dans une séquence type de conception à l'aide de *Sicam2*, le concepteur crée un nouveau projet (Projet⇒Nouveau) en donnant un nom dans la fenêtre prévue à cet effet (figure 3.7). Après vérification, le système valide le nouveau projet et initialise toutes les données du projet aux valeurs par défaut.

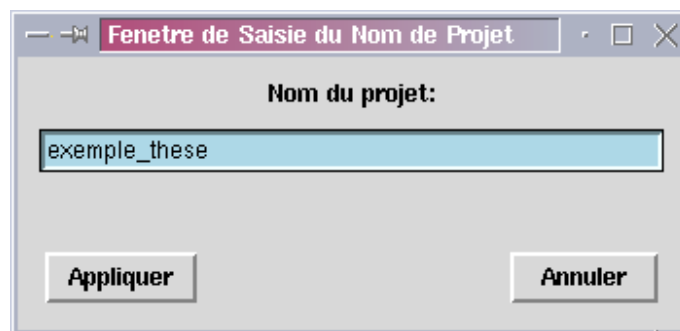


FIG. 3.7 – Fenêtre de saisie du nom de projet.

3.3 Initialisation de la phase de conception : le cahier des charges global.

La première étape de la phase de synthèse des DTLM consiste à remplir le cahier des charges global (bouton **CdC Global** du menu principal). Cette étape est nécessaire à la collecte des informations dérivées des niveaux supérieurs de définition. Dans l'organisation opérationnelle que nous avons définie en §1.6.2, nous avons situé le niveau d'action de *Sicam* à la tâche T4. Il est donc impératif de récupérer les dispositions conceptuelles issues des niveaux supérieurs. Dans le cadre d'un système intégré de conception, ces informations transitent naturellement par des mécanismes internes. Or, notre outil ne constitue qu'un élément de ce système qui, pour l'heure, ne gère pas les niveaux de définition supérieurs. Nous sommes donc contraints de simuler ce passage d'informations en mettant en place un menu de renseignement du CdC global.

La fenêtre du cahier des charges global (fig. 3.8) est divisée en plusieurs cellules. Les cellules

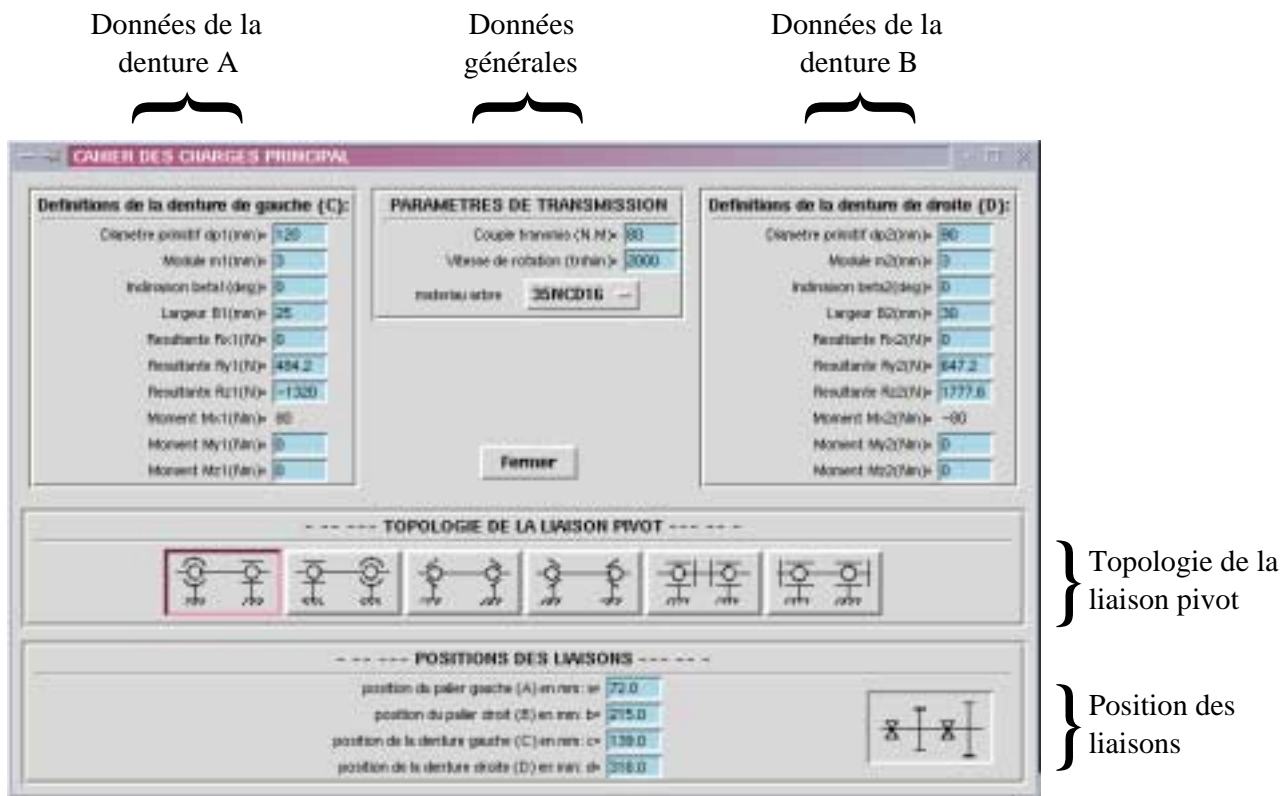


FIG. 3.8 – Cahier des charges global.

contenant les informations relatives aux dentures d'engrenage (associées aux liaisons complètes C et D) sont situées en haut à gauche et à droite de la fenêtre. Elles rassemblent les caractéristiques géométriques des dentures et les torseurs d'efforts transmis par chaque liaison.

En haut, au centre, on trouve les informations, dites générales, qui concernent la ligne d'arbres, soit l'ensemble des DTLM du sous-système étudié. Nous y trouvons les paramètres de transmission (vitesse de rotation et couple transmis) et du matériau de l'arbre que nous considérons comme monobloc. Certaines données telles que le moment de torsion sur l'arbre sont automatiquement déduites des informations entrées par l'utilisateur (en l'occurrence le couple transmis).

La zone médiane comprend une série de radio-boutons³ surmontés de graphiques permettant de déclarer la topologie de la liaison pivot. On entend par topologie la nature de la liaison pivot ainsi que l'agencement qualitatif de ses DTLM (montage en X ou en O, droite ou gauche). La liste est non-exhaustive, nous n'avons retenu que les cas *classiques* présentant des méthodes de calcul courantes. Il aurait été envisageable de proposer des topologies utilisant conjointement une rotule et un contact unilatéral par exemple, mais nous aurions alors eu à mettre en place des méthodes de calcul tenant compte de l'hyperstatisme comme dans le cas des contacts obliques. Nous proposons les topologies suivantes (de gauche à droite sur la figure 3.8) :

- rotule + linéaire annulaire,
- linéaire annulaire + rotule,
- contacts obliques avec un montage «en X»,
- contacts obliques avec un montage «en O»,
- rotule à contacts unilatéraux internes,
- rotule à contacts unilatéraux externes.

La sélection se fait en appuyant sur un des radio-boutons (sur la figure 3.8, c'est le montage rotule + linéaire annulaire qui est sélectionné). Notons que ces informations correspondent directement au niveau 3 de la hiérarchie objet que nous avons présentée en §2.2.1.

La dernière zone (partie inférieure de la fenêtre) est dévolue aux informations de positionnement des liaisons. L'opérateur dispose des quatre cellules de saisie (données numériques) pour fournir la cote des quatre centres géométriques des liaisons (cf. §2.1.4.2). Ces positions sont relatives à une origine commune et se situent sur un axe dont le sens positif est orienté vers la droite (cf. figure 2.5).

Le schéma de la vignette située à droite renvoie une illustration graphique de l'agencement des liaisons. Il est automatiquement mis à jour dès qu'une position est modifiée. L'utilisateur conceptualise ainsi plus facilement la topologie du mécanisme grâce à la substitution des valeurs numériques des positions par une représentation graphique. Cette dernière a le mérite de mettre en avant d'autres traits conceptuels du mécanisme ; par exemple, le fait de voir les deux paliers d'un même côté doit susciter chez le concepteur des réflexes de conception liés à un montage en porte à faux (ce qui réclame un effort intellectuel supplémentaire dans le cas d'une simple lecture des valeurs numériques des positions des liaisons).

Rappelons enfin que toutes les informations (hormis le matériau d'arbre que nous avons situé dans cette fenêtre pour son caractère d'information « générale ») contenues dans cette fenêtre sont issues des niveaux de définition supérieurs (cf. §1.6.2). Elle constitue la base du problème de définition de la tâche T4 et ne doit, en principe, être renseignée qu'une seule fois au cours d'une séquence de conception assistée.

La fin des opérations de renseignement du CdC global coïncide avec la fin de la phase d'initialisation de *Sicam* 2. À la fermeture de la fenêtre du CdC global, l'outil d'assistance crée un fichier contenant toutes les informations qui ont été recueillies. L'opérateur peut aborder l'action principale de la tâche T4, la synthèse des DTLM. Il doit tout d'abord s'attacher à définir les CdC liaisons pour mener à bien la synthèse découplée des DTLM. Nous présentons donc au chapitre suivant la démarche liée à cette phase de synthèse découplée.

3. Ensemble de boutons dont un seul peut être activé à la fois. Ils sont caractéristiques des questionnaires à choix multiple et réponse unique

Chapitre 4

Synthèse découplée des DTLM.

Résumé

LA synthèse découplée constitue le premier degré d'assistance offert par *Sicam2*. Nous discutons dans ce chapitre des principes directeurs de la phase de synthèse découplée. Nous présentons ensuite les menus liés à la définition des cahiers des charges locaux pour les liaisons pivot et complète. Nous distinguons en particulier les cas de DTLM liés à la liaison pivot selon qu'elle est décomposable ou non. Nous présentons également quelques dispositifs logiciels épaulant le concepteur dans le choix des DTLM. Enfin nous concluons sur l'intérêt et les conditions d'utilisation d'une telle méthodologie de conception.

Contenu du chapitre

4.1	Principes de la synthèse découplée.	95
4.2	Préalable à la synthèse de DTLM : détermination des torseurs d'efforts aux liaisons.	95
4.2.1	Liaison pivot décomposable.	96
4.2.2	Liaison pivot non décomposable.	97
4.3	Synthèse des DTLM de la liaison pivot.	98
4.3.1	Cas de deux DTLM indépendants.	98
4.3.2	Cas de deux DTLM dépendants.	100
4.4	Synthèse des DTLM des liaisons complètes arbre/moyeu.	102
4.4.1	Menu de définition commun aux diverses technologies de liaison.	102
4.4.2	Menu de définition caractéristique à la technologie de liaison.	104
4.4.2.1	Liaison complète par clavette.	104
4.4.2.2	Liaison complète par cannelures.	106
4.4.2.3	Liaison complète par frettage.	107
4.4.3	Remarques.	108

4.4.4	Exemple de synthèse découplée d'une liaison complète arbre/moyeu.	109
4.4.4.1	Présentation du problème.	110
4.4.4.2	Étude de la liaison complète par clavette.	110
4.4.4.3	Étude de la liaison complète par cannelures.	110
4.4.4.4	Étude de la liaison complète par frettage.	112
4.4.4.5	Observations.	112
4.5	Procédure de synthèse découplée de l'ensemble.	113
4.5.1	Gestion manuelle des contraintes.	114
4.5.1.1	Les interférences axiales.	114
4.5.1.2	Les interférences de montage.	116
4.5.1.3	Pertinence de la gestion manuelle des contraintes.	119
4.5.2	Aide au choix des DTLM.	119
4.5.3	Exemple de synthèse découplée.	122
4.6	Conclusions.	124

4.1 Principes de la synthèse découplée.

La phase de synthèse découplée fait suite à la phase d'initialisation du CdC global. Au début de cette phase, les paramètres généraux du mécanisme, issus des niveaux de définition supérieurs, sont connus. Le concepteur aborde la définition individuelle des DTLM composant le mécanisme. Il va pour cela effectuer des choix technologiques, pour chaque liaison définie précédemment, et procéder au dimensionnement des DTLM. Selon quels principes cette phase de synthèse découplée doit-elle se dérouler et quelle assistance peut-on attendre d'un outil d'aide à la conception ?

Le principe récurrent de la phase de synthèse découplée est un principe d'indépendance. Cette indépendance concerne l'étude des liaisons et se traduit par la non-prise en compte des contraintes de liaisons entre DTLM. Dans un premier temps, le concepteur doit avoir le loisir de définir les DTLM qu'il compte associer à chaque liaison mécanique du système étudié. Cela sous-entend la possibilité de dimensionner les DTLM indépendamment de leurs voisins, c'est-à-dire sans se soucier de l'impact des choix sur l'environnement direct du DTLM. Cette démarche autorise une recherche exhaustive des solutions pour une liaison donnée. En effet, n'étant pas contraint par les autres composants du système, le concepteur peut se concentrer sur la seule liaison qu'il étudie. Il peut tester les différents DTLM pour la classe de liaison considérée, et déterminer les extremums du domaine des solutions envisageables (minimum d'encombrement axial ou diamétral par exemple). Il peut ainsi évaluer l'importance qu'il compte attribuer à chacune des liaisons formant le système.

L'autonomie du concepteur est un autre principe également important. Nous visons à laisser au concepteur la plus grande capacité d'arbitrage des conflits survenant lors de la réunion des DTLM de chaque liaison. Nous devons pour cela mettre à sa disposition des mécanismes lui permettant d'agir sur les processus de calcul dimensionnel des DTLM. Ce moyen de peser sur le processus de production des solutions est fourni par la mise en place de jeux de contraintes propres à chaque liaison. Il peut ainsi orienter la recherche des DTLM solutions et par conséquent générer *manuellement* une solution globale. Notre démarche est volontaire. Elle s'oppose aux processus de résolution automatique qui, bien souvent, produisent des solutions selon des heuristiques internes et ne fournissent aucun des éléments de décision ayant permis d'obtenir de tels résultats.

Le découplage permet également de disposer d'objectifs d'optimisation différents pour chaque liaison. Il n'est pas rare de vouloir produire une solution dont les différentes parties respectent des critères d'optimisation différents. Cela se justifie au niveau local pour chaque liaison.

Nous présentons ci-dessous la problématique liée à la détermination des torseurs d'efforts aux liaisons.

4.2 Préalable à la synthèse de DTLM : détermination des torseurs d'efforts aux liaisons.

Le dimensionnement de chaque DTLM implique la connaissance des données nécessaires aux calculs. Celles-ci sont issues du CdC global et du CdC liaison. Pour autant, nous ne disposons pas toujours de tous les éléments indispensables à l'achèvement des calculs ; les éléments de réduction des torseurs d'efforts pour chaque liaison en sont un exemple. Ceux-ci permettent d'établir les diagrammes d'efforts qui sont requis lors de la validation de la géométrie de l'arbre par les modules de calcul. Deux cas de figure sont à distinguer ; la liaison pivot est décomposable ou non.

4.2.1 Liaison pivot décomposable.

La liaison pivot décomposable a été présentée en §2.2.1 au travers de l'approche orientée objet. Nous savons qu'à l'issue de la tâche T2, le concepteur a défini la topologie des liaisons pivot du mécanisme. Il a pu opter pour une décomposition en une liaison linéaire annulaire et une liaison rotule. En ce cas, nous pouvons *instantanément* déterminer la forme des torseurs d'efforts pour chaque liaison. Comme nous l'avons précisé dans notre approche objet, cette information peut être capitalisée à un niveau supérieur à celui du DTLM. En effet, quelle que soit la technologie que nous retiendrons pour réaliser la liaison pivot et la liaison linéaire annulaire, les torseurs associés resteront inchangés et de la forme :

$$\begin{cases} X & 0 \\ Y & 0 \\ Z & 0 \end{cases} \text{ pour la liaison rotule et } \begin{cases} 0 & 0 \\ Y' & 0 \\ Z' & 0 \end{cases} \text{ pour la liaison linéaire annulaire}$$

Les calculs de statique sont donc indépendants de la procédure de synthèse des DTLM et peuvent être lancés avant d'atteindre le stade du choix des DTLM comme le laisse apparaître la structure objet présentée précédemment (voir notamment figure 2.15(b)).

Ce calcul peut être mené simplement, en faisant appel à un algorithme qui met en œuvre le principe fondamental de la statique. Considérons l'exemple de la figure 4.1. Il modélise un système

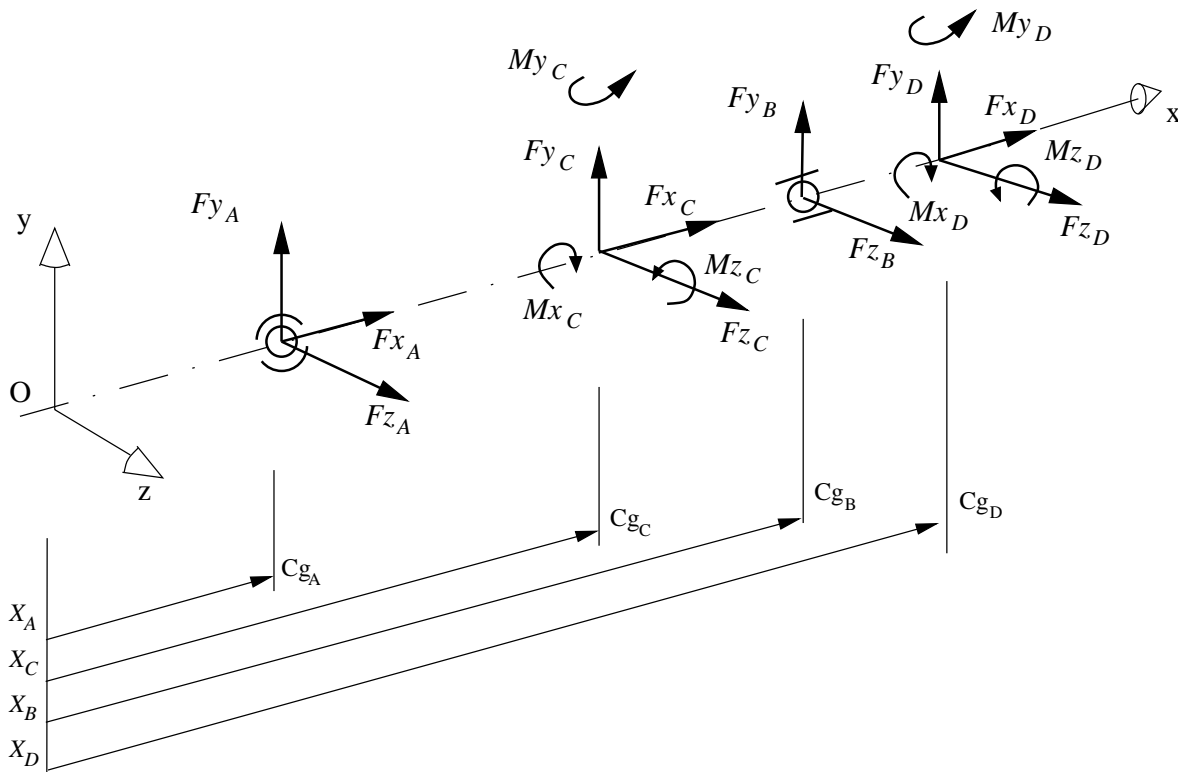


FIG. 4.1 – Exemple d'un cas de statique associé à une synthèse de DTLM.

composé d'une liaison pivot, décomposée en une liaison rotule et une liaison linéaire annulaire, ainsi que de deux liaisons complètes. Ces dernières transmettent la puissance transitant par des liaisons à bases d'engrenages à dentures hélicoïdales (d'où la composante d'effort axial). Les torseurs d'efforts des liaisons sont exprimés aux centres de liaison respectifs. La solution d'un tel problème s'obtient en écrivant :

$$\sum F_i = 0 \quad \text{et} \quad \sum M_i = 0$$

soit pour l'expression des efforts :

$$\begin{cases} Fx_A + Fx_C + Fx_D = 0 \\ Fy_A + Fy_B + Fy_C + Fy_D = 0 \\ Fz_A + Fz_B + Fz_C + Fz_D = 0 \end{cases}$$

et pour l'expression des moments (au point O) :

$$\begin{cases} Mx_C + Mx_D = 0 \\ -Fz_A \cdot X_A - Fz_B \cdot X_B - Fz_C \cdot X_C - Fz_D \cdot X_D + My_C + My_D = 0 \\ Fy_A \cdot X_A + Fy_B \cdot X_B + Fy_C \cdot X_C + Fy_D \cdot X_D + Mz_C + Mz_D = 0 \end{cases}$$

La résolution informatique ne pose aucune difficulté et peut être obtenue par toute méthode adaptée à ce type de système d'équations. Les cinq inconnues de ce problème sont les variables :

$$\{Fx_A, Fy_A, Fz_A, Fy_B, Fz_B\}$$

Bien entendu, pour un cas de calcul impliquant une décomposition différente de la liaison pivot (par exemple linéaire annulaire + rotule), nous obtiendrons une formulation du problème et un jeu d'inconnues différents. Tous les cas ont été envisagés.

4.2.2 Liaison pivot non décomposable.

La situation provoquée par la détermination d'une liaison pivot réalisée par roulements à contact oblique est problématique. Cette dernière se décompose *effectivement* en deux composants technologiques dissociés, mais leur dimensionnement doit être mené *simultanément* (cas des DTLM dépendants). Selon la méthode de calcul généralement fournie par les fabricants, on ne peut, à l'avance, déterminer quel sera le roulement qui assurera la transmission de l'effort axial de l'arbre au carter. De plus, cette détermination est directement liée au roulement que l'on désire utiliser (angle de contact). Il est donc impossible de calculer les torseurs d'efforts sans avoir auparavant choisi les roulements intervenant dans chacun des DTLM. Ceci implique que le calcul de statique et le dimensionnement des DTLM devront être menés simultanément.

Ceci a une conséquence évidente sur le processus de synthèse découplée : ne connaissant pas à l'avance les torseurs d'efforts pour chaque liaison, il est impossible de déterminer les diagrammes d'efforts dans l'arbre. Ceci pénalise les modules de calcul qui, sans indication sur les moments de flexion, de torsion, etc, ne peuvent évaluer les concentrations de contraintes liées aux géométries de l'arbre.

Une solution aurait consisté à mettre en place un calcul dégradé de synthèse dimensionnelle en négligeant, ou en évaluant grossièrement là où cela était possible (entre les liaisons arbre/moyeu), les sollicitations internes de l'arbre. Nous avons renoncé à cette option qui aurait rendu nécessaire un second calcul pour affiner les résultats.

En conclusion, le concepteur intégrant une liaison pivot non décomposable doit impérativement commencer par la synthèse des DTLM de la liaison pivot afin d'obtenir conjointement les résultats du calcul de statique ainsi que les diagrammes d'efforts.

La section suivante présente les menus sur lesquels le concepteur s'appuiera pour mener à bien la synthèse des DTLM de la liaison pivot. Cette section est divisée en deux parties pour tenir compte des spécificités des liaisons pivots décomposables (DTLM indépendants) et non décomposables (DTLM dépendants).

4.3 Synthèse des DTLM de la liaison pivot.

Le traitement de la liaison pivot présente quelques singularités. Nous avons détaillé en §2.2.1 les particularités de la structure de données propre à la liaison pivot. Les menus dont nous allons analyser le contenu reflètent cette structure. Nous présentons la définition des DTLM de la liaison pivot en distinguant deux cas : tout d'abord le cas des DTLM indépendants (liaison pivot parfaitement décomposable), puis celui des DTLM non indépendants (typiquement le cas des liaisons mettant en jeu des roulement à contact oblique).

Les menus de définition des DTLM de la liaison pivot sont accessibles à partir du bouton **CdC Pivot** du menu principal. Ils dépendent du choix de décomposition de la liaison pivot par le concepteur (données des niveaux précédents contenues dans le CdC global).

4.3.1 Cas de deux DTLM indépendants.

La fenêtre du menu de définition des DTLM de la liaison pivot est représentée sur la figure 4.2. Elle est composée des éléments suivants. Une cellule, dans la partie haute, présente un rappel

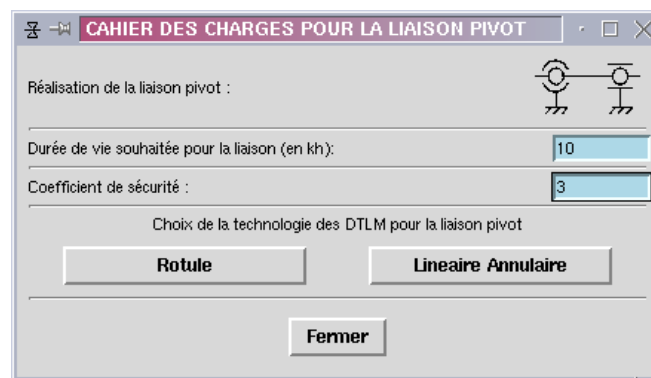


FIG. 4.2 – Menu de définition des DTLM d'une liaison pivot : cas des DTLM indépendants.

de la topologie de la liaison. Le graphisme utilisé est identique à celui figurant dans le menu du CdC global (figure 3.8). Dans l'exemple de la figure, le concepteur a choisi de réaliser la liaison pivot en utilisant une liaison rotule à gauche et une liaison linéaire annulaire à droite. Juste en dessous, nous trouvons les zones de saisie concernant la durée de vie de la liaison ainsi que le coefficient de sécurité de calcul.

La zone inférieure comporte deux boutons qui donnent accès aux sous-menus de définition des DTLM. Lorsque le concepteur presse le bouton **Rotule** ou **Lineaire Annulaire** il ouvre les fenêtres relatives à chacun des DTLM de la liaison rotule et linéaire annulaire (figures 4.3).

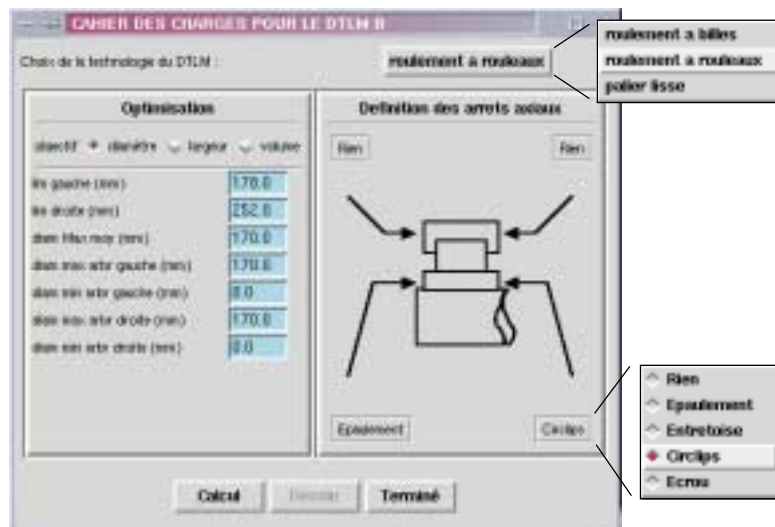
Ces fenêtres ont une disposition commune que nous allons détailler. La zone supérieure comprend un menu-bouton fixant la technologie principale du DTLM. On peut voir que la rotule utilise un roulement radial à billes tandis que la linéaire annulaire utilise un roulement à rouleaux cylindriques choisi parmi les composants suivants : roulement radial à billes, roulement à rouleaux cylindriques et palier lisse (ce dernier est dimensionné de telle façon qu'il soit toujours assimilable à une liaison linéaire annulaire).

Viennent ensuite deux zones permettant de préciser les paramètres du DTLM. La zone de gauche autorise un choix de critère d'optimisation (minimisation du diamètre d'arbre, de la largeur du DTLM ou de son volume) ainsi que la saisie des valeurs limites (borne axiale gauche et droite, diamètre limites d'encombrement du DTLM et limites inférieures et supérieures de diamètre de tronçon d'arbre à droite et à gauche). La zone de droite est réservée au choix des

arrêts axiaux. Autour d'une image symbolisant la technologie de roulement retenu sont répartis des menu-boutons proposant les arrêts axiaux disponibles pour les bagues intérieures (arbre) et extérieures (alésage). Parmi les technologies d'arrêts axiaux, on trouve des éléments « stan-



(a) DTLM de la liaison rotule.



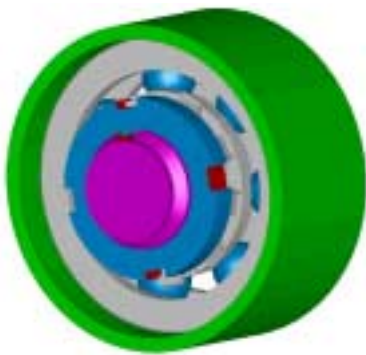
(b) DTLM de la liaison linéaire annulaire.

FIG. 4.3 – Fenêtres de définition complémentaires pour DTLM indépendants.

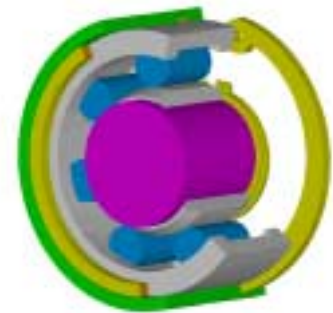
dards» (tels qu'anneaux élastiques, entretoise...) qui sont également proposés pour les liaisons arbre/moyeu complètes. A celles-ci vient s'ajouter l'écrou crénelé pour les bagues intérieures.

Une fois toutes les données fournies au système, le concepteur lance les calculs (bouton **Calcul**). Il est alors généré un fichier résultat contenant les caractéristiques technologiques et dimensionnelles des DTLM optimaux de la liaison pivot. Le concepteur peut demander l'affichage du modèle 3D (bouton **Dessin**). La figure 4.4 donne un aperçu de modèles 3D obtenus après calcul des DTLM de la liaison pivot. On notera enfin que toutes les opérations concernant les

DTLM de la liaison rotule et linéaire annulaire sont indépendantes, conformément à leur structure objet.



(a) Liaison rotule.



(b) Liaison linéaire annulaire.

FIG. 4.4 – Modèles 3D de DTLM associés à une liaison pivot.

4.3.2 Cas de deux DTLM dépendants.

Le menu de définition de la liaison pivot, dans le cas de DTLM dépendants, reprend l'organisation générale du menu présenté en §4.3.1. Le seul élément distinctif est la présence d'un unique bouton d'accès au sous-menu de définition (figure 4.5). C'est en appuyant sur le bouton **Les deux DTLM** que le concepteur ouvre la fenêtre de définition commune aux deux DTLM. Ce

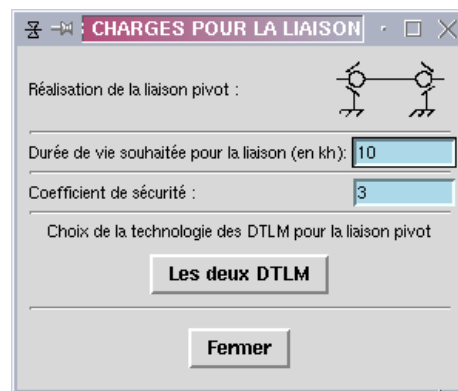


FIG. 4.5 – Menu de définition des DTLM d'une liaison pivot : cas des DTLM dépendants.

bouton unique matérialise l'obligation du concepteur de définir simultanément les deux DTLM, faute de quoi nous ne pouvons lancer le calcul de dimensionnement.

Cette fenêtre (figure 4.6) adopte la même disposition que les fenêtres de sous-menu des liaisons pivot décomposables. Le concepteur a également accès aux mêmes éléments. La zone supérieure comprend le menu-bouton permettant de sélectionner la technologie principale des DTLM. Dans notre exemple, le concepteur a opté pour une liaison pivot non décomposable réalisée à l'aide de roulements à rouleaux coniques (contact oblique). Le bouton unique reflète la singularité discutée en §4.2.1 concernant les méthodes de calcul itératives propres à ce type de composants. Nous aurions éventuellement pu envisager le panachage des technologies de roulement au sein

de la liaison pivot (par exemple combiner un roulement à rouleaux coniques avec un roulement à billes à contact oblique) mais il aurait alors été nécessaire d'élaborer de nouvelles règles de dimensionnement.

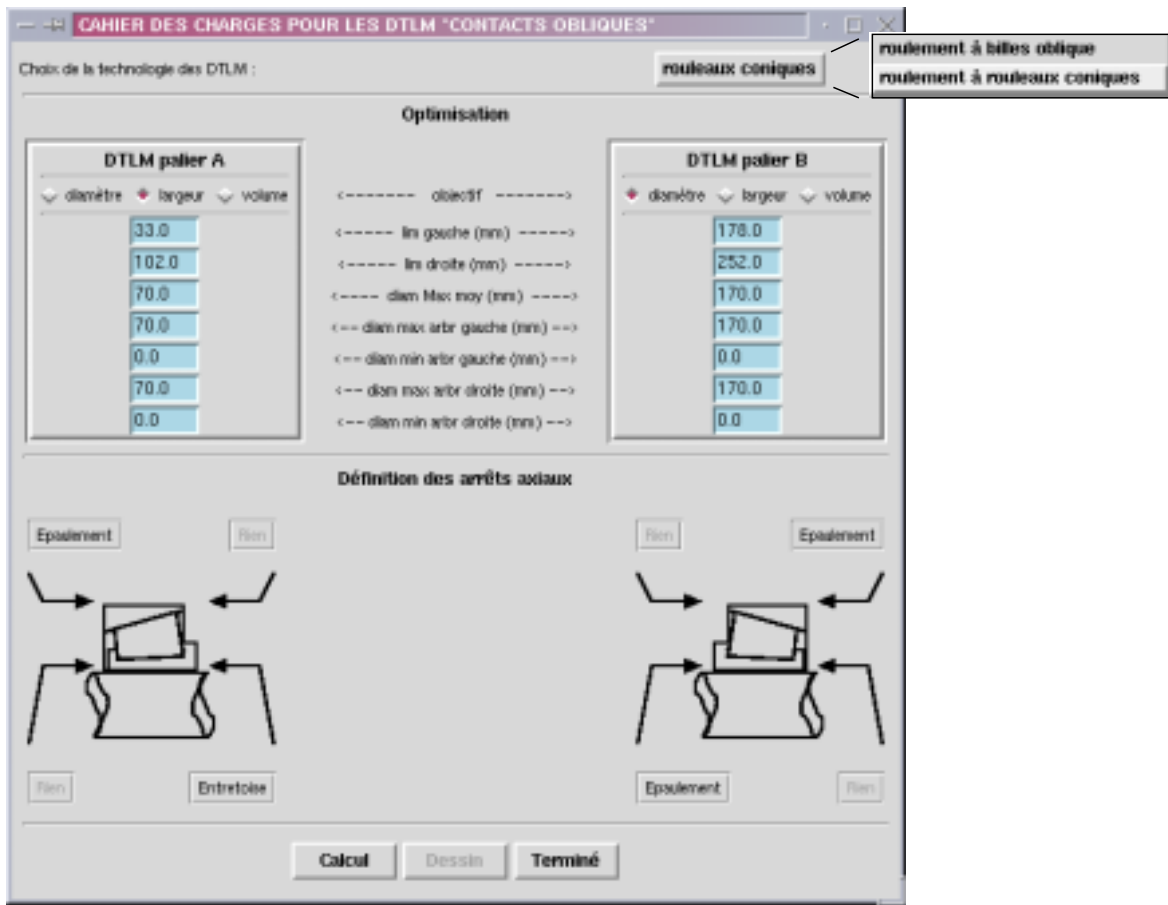


FIG. 4.6 – Fenêtre de définition complémentaire pour DTLM dépendants.

La suite de la définition des DTLM respecte la même logique que précédemment. La zone médiane permet de définir simultanément les valeurs limites liées à chacun des DTLM tandis que la zone inférieure est consacrée à la définition des arrêts axiaux. Nous avons volontairement désactivé la moitié des boutons définissant les arrêts axiaux afin de se conformer à la topologie de la liaison. Dans le cas représenté sur la figure, le montage en « X » impose des arrêts axiaux sur les faces externes des bagues extérieures et sur les faces internes des bagues intérieures. Les autres arrêts risquent d'engendrer un hyperstatisme de la liaison et requièrent en conséquence la mise en place de jeux fonctionnels. Il sont donc neutralisés (valeur par défaut = « rien »).

La figure 4.7 présente le modèle 3D associé à la liaison pivot définie ci-dessus.

Pour clore la fenêtre, les boutons **Calcul**, **Dessin** et **Terminé** commandent respectivement le lancement du calcul couplé des deux DTLM, l'affichage des modèles 3D et la fermeture de cette fenêtre.

La présentation des menus de définition des DTLM de la liaison pivot respecte les particularités qui ont été mises en évidence par la structure objet. Elle autorise une démarche de synthèse de DTLM homogène malgré des différences notables entre liaisons pivots décomposables et non décomposables. La section suivante poursuit la présentation de la synthèse des DTLM au travers des liaisons complètes arbre/moyeu.

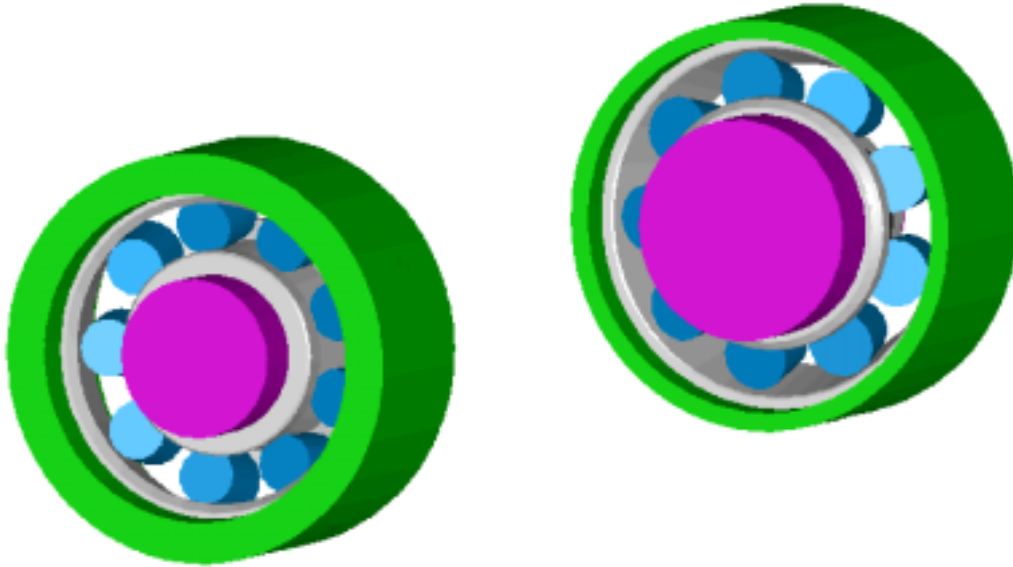


FIG. 4.7 – Modèles 3D de DTLM à base de roulements à rouleaux coniques.

4.4 Synthèse des DTLM des liaisons complètes arbre/moyeu.

Le concepteur accède au menu de définition des DTLM d'une des deux liaisons complètes arbre/moyeu par les boutons `CdC arbre/moyeu C` ou `CdC arbre/moyeu D` du menu principal (cf. figure 3.6). Il active ainsi le menu de définition commun d'une des liaisons complètes.

4.4.1 Menu de définition commun aux diverses technologies de liaison.

La définition des DTLM s'effectue en deux temps, reflétant ainsi la structure de la figure 2.34. La première étape implique la fenêtre de la figure 4.8. Cette fenêtre est identique pour toutes les liaisons complètes traitées (seul son titre, précisant à l'opérateur la liaison en cours de traitement, diffère). Nous définissons cette fenêtre comme un menu de définition commun, car il ne dépend pas de la technologie de DTLM qui sera retenue par le concepteur. Ce menu ne recense qu'une partie des informations concernant la liaison. Le reste des éléments de définition du DTLM dépend du choix de technologie effectué dans ce menu commun.

Notre outil propose trois grandes familles de liaisons complètes que l'on identifie par la technologie d'arrêt en rotation utilisée. Le choix de cette technologie s'effectue par les boutons situés dans la partie supérieure de la fenêtre. Le concepteur peut opter pour une des trois liaisons complètes arbre/moyeu suivantes :

- clavette parallèle (bouton `Clavette_//`),
- cannelures (bouton `Cannelure`),
- frettage (bouton `Frettage`).

Ce choix de la technologie conditionnera l'ouverture de la fenêtre suivante.

La zone suivante (partie gauche de la fenêtre) regroupe les informations concernant le matériau du moyeu, nécessaire pour les calculs de résistance des matériaux, et le critère d'optimisation.

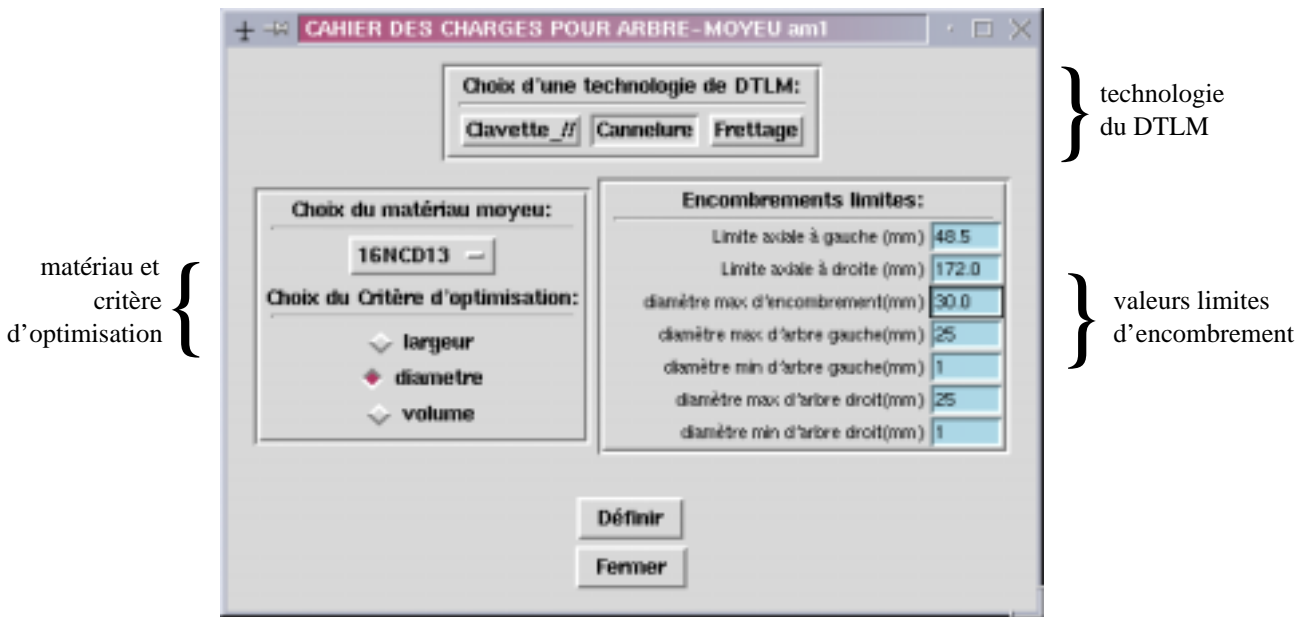


FIG. 4.8 – *Cahier des charges pour la liaison complète arbre/moyeu.*

Le matériau est choisi en parcourant la liste du menu-bouton¹ constituée des différents items de la base de données matériau. La figure 4.9 propose cette liste en marquant le choix courant d'un losange foncé. Le critère d'optimisation est choisi en cliquant sur un des radio-boutons



FIG. 4.9 – *Liste des options du menu-bouton « choix de matériau ».*

correspondant à une minimisation de la largeur, du diamètre ou du volume du DTLM.

La dernière zone contient des cellules de saisie consacrées aux valeurs limites d'encombrement du DTLM. Il s'agit des limites axiales (gauche et droite), du diamètre maximal du cylindre d'encombrement, ainsi que des diamètres maximaux et minimaux (gauche et droit) de l'arbre.

Les deux derniers boutons ont pour fonction respectivement l'ouverture de la fenêtre suivante de définition du DTLM (bouton **Définir**) et la fermeture de la fenêtre actuelle (bouton **Fermer**) lorsque elle sera devenue inutile. Lorsque le concepteur ouvre la fenêtre suivante de définition du DTLM, il obtiendra une fenêtre adaptée à la technologie retenue pour la liaison.

1. Bouton proposant, lorsqu'il est activé, une liste d'options à choix unique. Seul l'élément sélectionné subsiste en facade du bouton.

4.4.2 Menu de définition caractéristique à la technologie de liaison.

4.4.2.1 Liaison complète par clavette.

Notre outil propose une liaison arbre/moyeu complète, démontable, par clavetage libre. Deux formes de clavettes parallèles sont proposées, que nous associons volontairement à deux types d'usinages distincts : les formes de type A (clavette à bouts ronds) et B (clavette à bouts carrés).

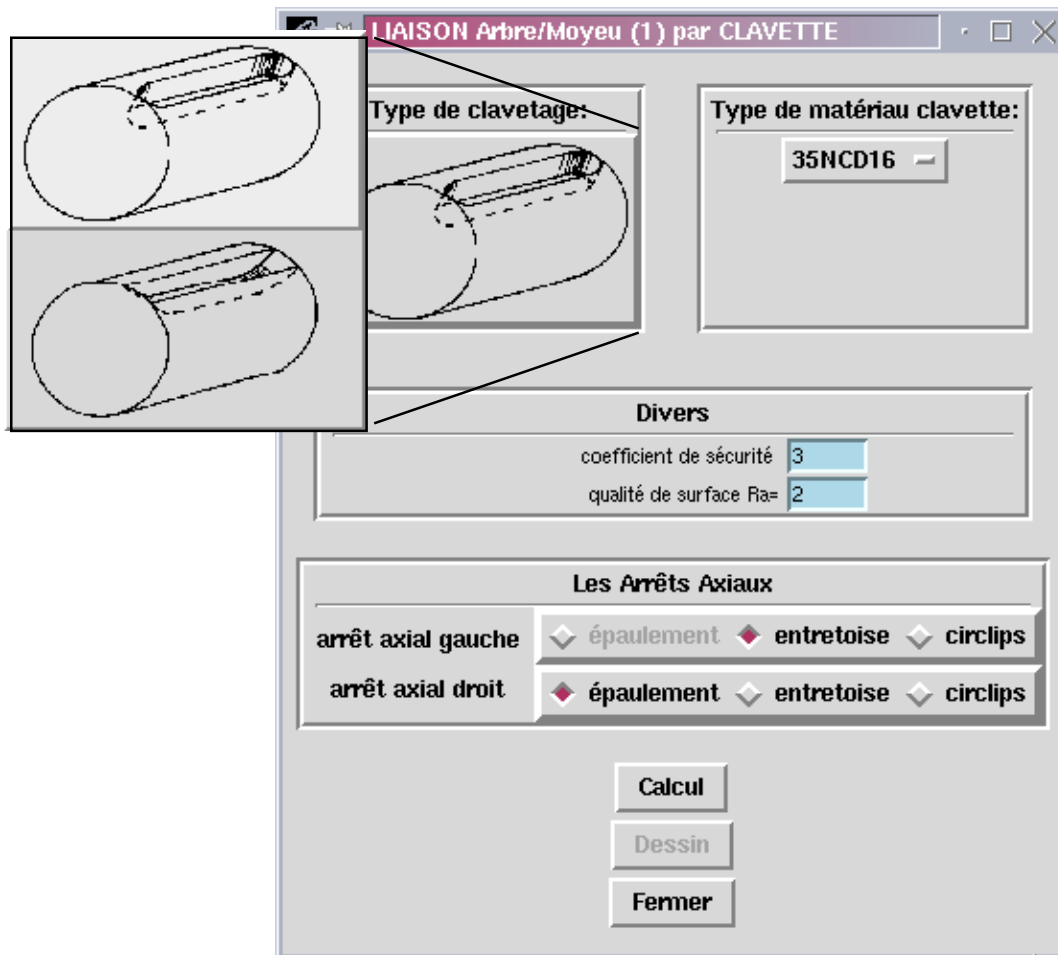


FIG. 4.10 – Menu de définition caractéristique de la liaison complète arbre/moyeu par clavette.

Analysons le contenu des cellules de gauche à droite et de haut en bas. La première cellule est formée d'un menu-bouton graphique qui détermine la forme de clavette (symbolisée par son empreinte sur le tronçon d'arbre). La cellule suivante reçoit les informations de matériau de la clavette. La troisième cellule recense les informations diverses telles que le coefficient de sécurité de calcul du DTLM ainsi que l'état de surface de la rainure de l'arbre. La dernière cellule présente deux séries de radio-boutons destinées au choix des arrêts axiaux, respectivement à gauche et à droite de l'arbre. En bas de la fenêtre, nous trouvons des boutons permettant de lancer les calculs (**Calcul**), d'exécuter la modélisation 3D du DTLM (**Dessin**) et de fermer le menu caractéristique de définition (**Fermer**).

Dans le cas représenté sur la figure 4.10, le concepteur a retenu une solution impliquant une clavette de forme A réalisée en 35 NCD 16, dont le coefficient de sécurité de calcul est fixé à une valeur de 3 pour un Ra de 2 μm . Les arrêts axiaux retenus pour compléter l'immobilisation du

moyeu par rapport à l'arbre sont une entretoise du côté gauche et un épaulement du côté droit. Les résultats (dimensions optimales) produits par le module de calcul sont enregistrés dans un fichier au format texte afin que le concepteur puisse, s'il le désire, les consulter. Ce type de fichier enchaîne les champs de données composés d'une ligne de description du champ et d'une ligne de valeur numérique ou alphanumérique (figure 4.11). Ce fichier est lu par la procédure de dessin

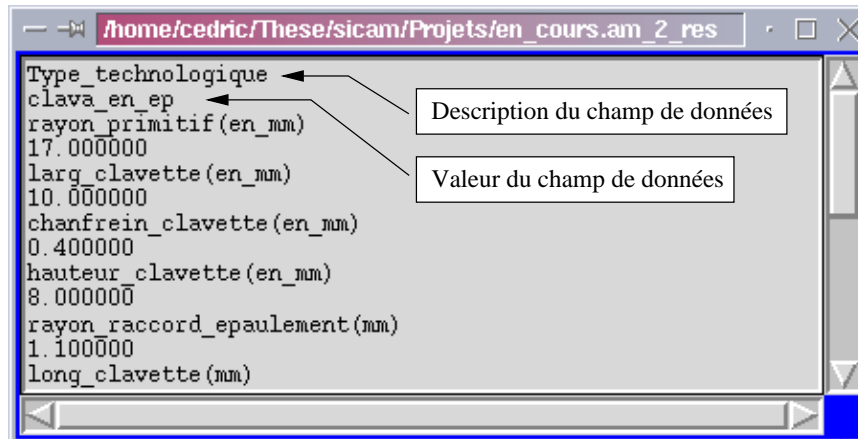


FIG. 4.11 – Fenêtre de visualisation d'un fichier « résultats de calcul ».

du modèle 3D pour produire le résultat représenté sur la figure 4.12.

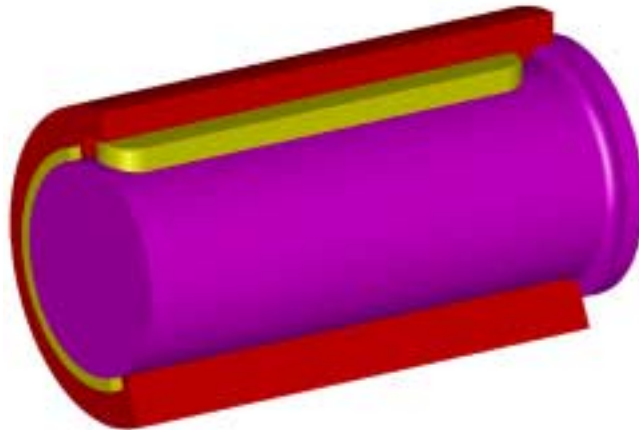


FIG. 4.12 – Modèle 3D d'un DTLM à base de clavette.

Nous n'avons pas procédé à un nouvel éclatement de la fenêtre après le choix de la forme de clavette. Nous avons privilégié la lisibilité à la rigueur du découpage, et les informations requises actuellement pour les modèles de calcul implémentés sont identiques. Dans l'hypothèse où les données impliquées dans les calculs dépendraient fortement de la forme de clavette, il serait naturellement envisageable de procéder à un découpage du menu caractéristique, afin de refléter la structure objet du DTLM. Nous conserverions alors dans le nouveau menu caractéristique, les éléments communs aux formes A et B tels que le coefficient de sécurité et le matériau de la clavette.

4.4.2.2 Liaison complète par cannelures.

Plusieurs types de cannelures peuvent assurer une liaison arbre/moyeu. Nous n'avons retenu que les cannelures à flancs en développante à centrage sur flancs ([AFN55], [AFN85] et [AFN87a]), car elles fournissent un panel plus large de solutions. Nous éliminons donc les cannelures à flancs parallèles (de fabrication souvent plus coûteuse et de performances moindres) ainsi que les arbres dentelés (trop faible couple transmissible). Bien entendu, ces types de liaisons ont vocation à être intégrés dans *Sicam2* afin d'élargir le champ d'application de l'outil ; ainsi la liaison par arbre dentelé trouve pleinement sa place dans les applications de faible puissance où une certaine précision de liaison angulaire est requise.

La figure 4.13 représente la fenêtre du menu de définition caractéristique de la liaison complète par cannelures. Décrivons les cellules dans l'ordre adopté en §4.4.2.1. La première cellule

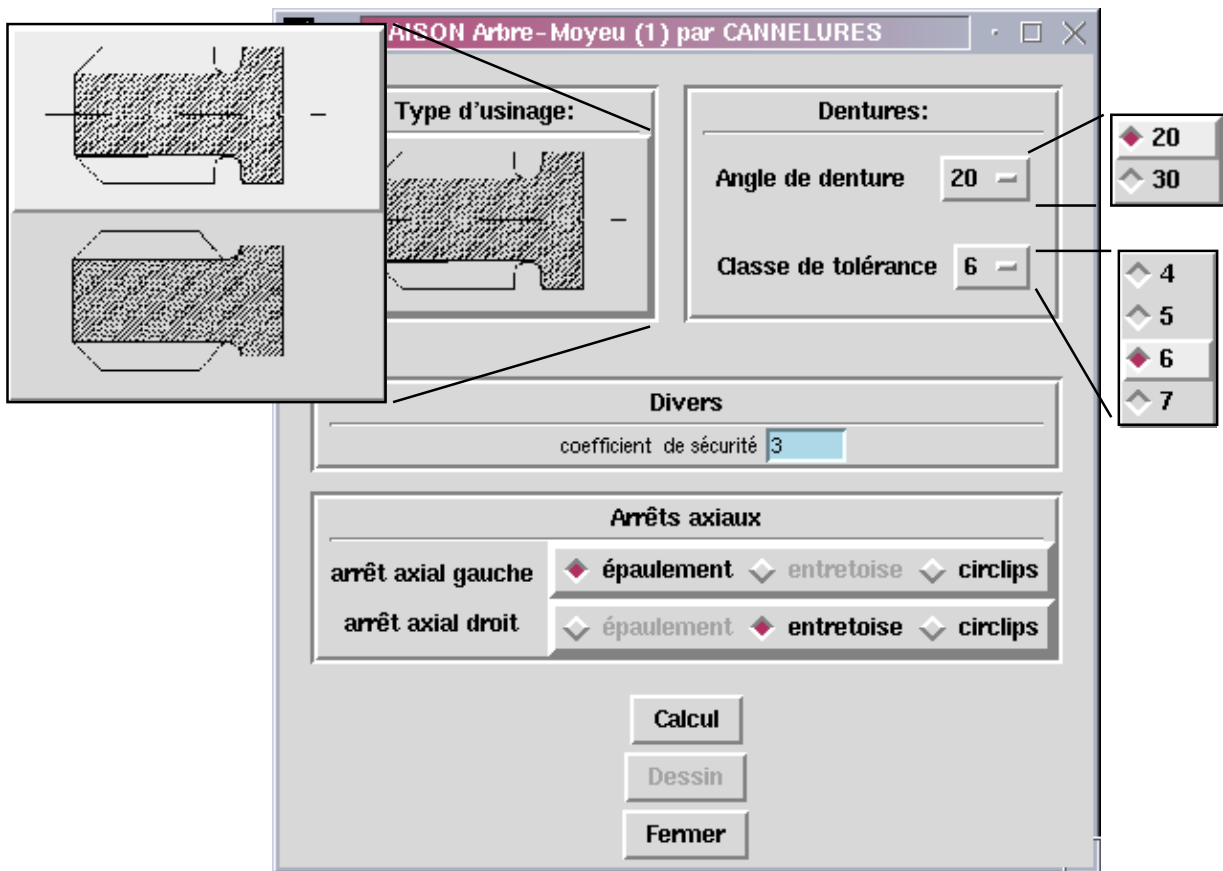


FIG. 4.13 – Menu de définition caractéristique de la liaison arbre/moyeu par cannelures.

reçoit, grâce à un menu-bouton graphique, le type d'usinage de cannelures (roulage ou taillage). La deuxième cellule recense les caractéristiques des dentures. Nous y trouvons donc l'angle de denture pour lequel nous avons retenu deux valeurs : 20° correspondant à un taillage identique aux engrenages courants, et 30° pour un aperçu des capacités de charge des séries à fort angle de pression. Le choix de la classe de tolérance normalisée (à choisir parmi les quatre considérées pour l'angle de contact de 30°) complète les caractéristiques de la denture. Nous retrouvons dans la cellule suivante le coefficient de sécurité de calcul puis la dernière cellule proposant les arrêts axiaux. Les boutons **Calcul**, **Dessin** et **Fermer** closent enfin la fenêtre.

Le DTLM défini par le menu de la figure 4.13 est une liaison par cannelures à flancs en développante d'angle 20° , de classe de tolérance 6 avec un épaulement à gauche et une entretoise

à droite. Les calculs de dimensionnement optimal sont réalisés avec un coefficient de sécurité de valeur égale à 3. Le modèle 3D du résultat obtenu est représenté sur la figure 4.14.

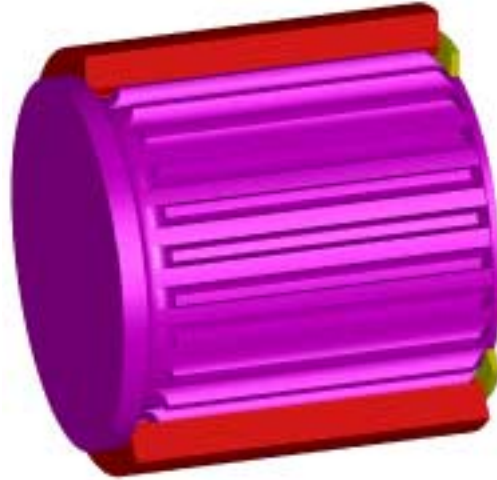


FIG. 4.14 – *Modèle 3D d'un DTLM à base de cannelures.*

4.4.2.3 Liaison complète par frettage.

Le dernier type de DTLM proposé par notre outil est une liaison par frettage. Il diffère des solutions précédentes en deux points :

- il assure la liaison par adhérence, et non par obstacle ;
- il n'est pas toujours démontable (sans endommagement des surfaces concernées).

Les modèles de calcul retenus sont les modèles normalisés ([AFN80] et [AFN84]). Le diamètre extérieur du moyeu sera toujours pris égal à sa valeur limite maximale imposée par le concepteur. La prise en compte de cette valeur est en effet indispensable au calcul. La figure 4.15 représente le menu de définition caractéristique de la liaison arbre/moyeu par frettage. En partant du haut de la fenêtre, nous remarquons une cellule recensant les qualités d'arbre et d'alésage. Nous considérons que l'alésage est « normal »², c'est-à-dire que les difficultés d'usinage sont reportées sur l'arbre (raisons essentiellement économiques). Le concepteur définit des classes de qualité dimensionnelles pour l'arbre et l'alésage, et *Sicam* recherche, compte tenu des matériaux respectifs, le diamètre de l'arbre assorti de son ajustement et la longueur de frettage. Les coûts engendrés seront fortement liés aux méthodes de réalisation, donc aux indices de qualité retenus. Les calculs seront menés avec le coefficient de sécurité figurant dans la cellule médiane. En dernière position se situe la cellule proposant les arrêts axiaux. Remarquons l'apparition d'un nouveau type d'arrêt axial qui n'en est pas un ! Le concepteur a la possibilité, pour la liaison par frettage, de laisser libre un (ou les deux) des côtés de l'assemblage. Nous avons rendu cela possible en intégrant la prise en charge des efforts axiaux dans le calcul de frettage. Ainsi, il n'est plus utile de disposer d'arrêts axiaux. La présence d'un épaulement est néanmoins souhaitable pour le positionnement du moyeu lors du frettage.

². Alésage normal signifie que l'incertitude dimensionnelle liée aux défauts de réalisation de l'alésage entraîne un diamètre réel égal ou supérieur au diamètre nominal.

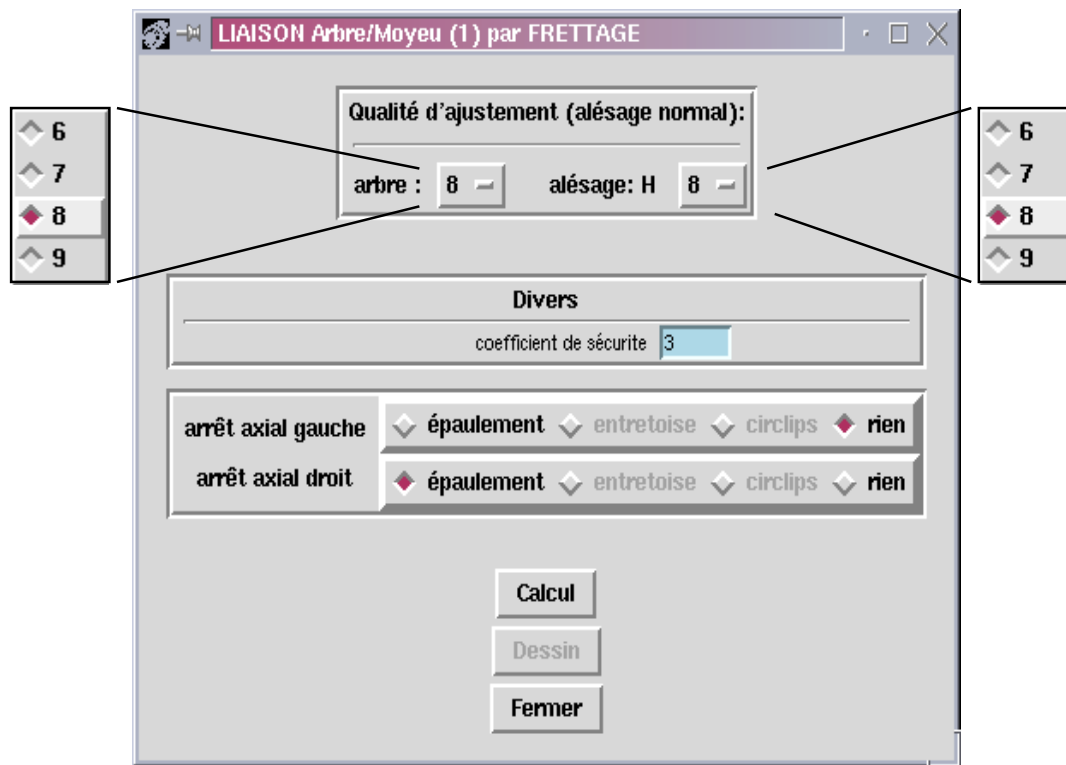


FIG. 4.15 – Menu de définition caractéristique de la liaison arbre/moyeu par frettage.

Le DTLM défini par le menu caractéristique de la figure 4.15 est une liaison par frettage à alésage normal, de qualité 8, et d'arbre de qualité 8 également. La liaison présente pour seul arrêt axial un épaulement du côté droit. Le calcul des dimensions de l'assemblage sera mené avec un coefficient de sécurité de valeur égale à 3 dès la commande de l'opérateur par le bouton **Calcul**. Les résultats optimaux obtenus seront utilisés pour la construction du modèle 3D. Sa visualisation (figure 4.16) sera commandée par le bouton **Dessin**. Le concepteur quittera le menu caractéristique en appuyant sur le bouton **Fermer**.

4.4.3 Remarques.

Les menus que nous venons de présenter amènent quelques remarques. Le lecteur aura pu constater que certaines valeurs figuraient en grisé dans les cellules définissant les arrêts axiaux. Leur sélection est rendue impossible par un mécanisme dynamique de filtrage des combinaisons d'arrêts. Il désactive les radio-boutons qui peuvent mener à des combinaisons irréalisables ou indisponibles. Cela évite au concepteur de « composer » un DTLM qui ne figure pas dans la liste des DTLM pris en charge par *Sicam2* (cf. tableau 2.1). Nous considérons ce mécanisme comme un détrompeur, plus qu'un système intelligent, car son fonctionnement est algorithmique et basé sur des règles combinatoires extrêmement simples.

La localisation du coefficient de sécurité dans les menus caractéristiques nécessite quelques explications. Le coefficient de sécurité peut être généralement lié à :

- la fonction (donc ici la liaison). En ce cas, il doit avoir la même valeur, quelle que soit la technologie choisie. Nous devons alors le faire figurer dans la partie commune du cahier des charges de la liaison, c'est-à-dire le menu de définition commun.

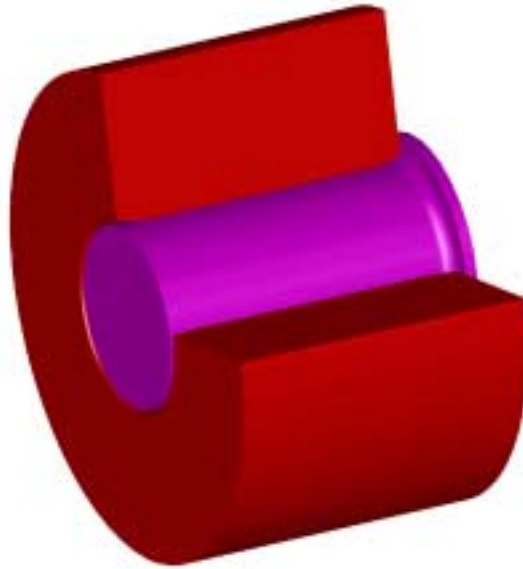


FIG. 4.16 – *Modèle 3D d'un DTLM de liaison arbre/moyeu par frettage.*

- la technologie. Les méthodes de calcul sont plus ou moins fiables, et le coefficient de sécurité s'adapte aux modèles utilisés. Le concepteur doit donc pouvoir régler simultanément la valeur du coefficient et les paramètres de définition propres à la technologie retenue. Dans ce cas, le coefficient doit prendre place dans les menus de définition caractéristiques.

Nous avons retenu le second principe et situé le coefficient de sécurité dans les menus caractéristiques.

Le découpage hiérarchique des menus reflète le plus fidèlement possible la structure objet mise en place. Tout attribut pouvant être capitalisé à un niveau donné de la définition du DTLM a été intégré dans un menu de définition correspondant. Pour des raisons de simplification de l'interface graphique, nous avons conservé un faible nombre de niveaux dans la hiérarchie des menus. Certains attributs présentent une versatilité qu'il est obligatoire de maîtriser. Considérons l'exemple des arrêts axiaux. La présence d'arrêts axiaux est commune à toutes les technologies que nous avons implémentées. Il serait donc logique de placer cet attribut dans le menu de définition commun, mais nous avons constaté que la liaison par frettage ne requiert pas les mêmes éléments que ses homologues. Nous en déduisons donc que l'éventail des propositions d'arrêts axiaux *dépend de la technologie* principale du DTLM. Il est donc légitime de proposer ces attributs une fois la technologie du DTLM définie.

Cette capitalisation des attributs présente un intérêt certain lors de modifications du système. L'exemple que nous présentons ci-dessous illustre notre propos par un cas d'étude où le concepteur doit évaluer trois technologies différentes pour une même liaison complète.

4.4.4 Exemple de synthèse découplée d'une liaison complète arbre/moyeu.

Dans l'exemple que nous présentons, le concepteur met en concurrence trois DTLM qu'il soumet au dimensionnement selon un cahier des charges identique.

4.4.4.1 Présentation du problème.

L'énoncé du problème ne présente que peu d'intérêt en soi et nous pouvons le résumer par le schéma de la figure 4.17. Le montage proposé est celui d'un arbre réalisant un étage de

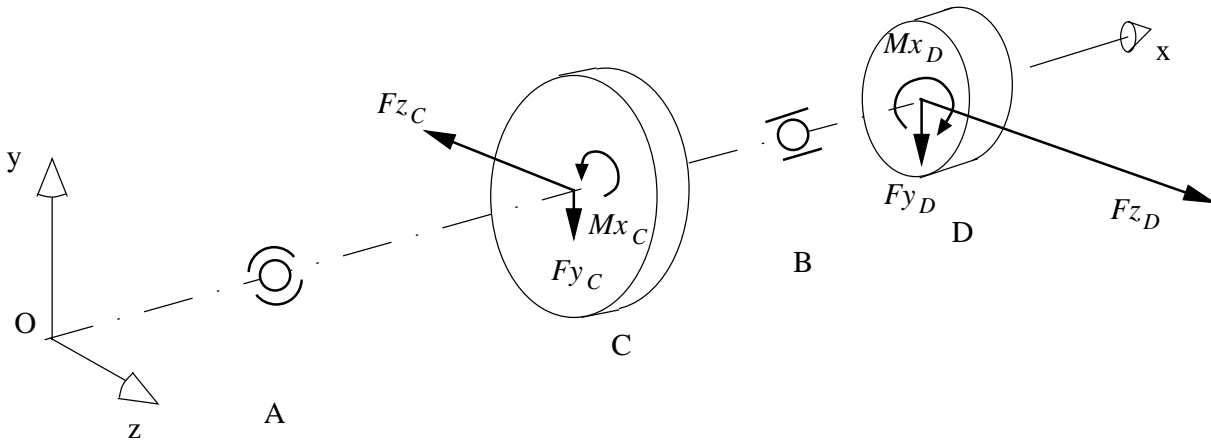


FIG. 4.17 – Configuration de l'exemple de synthèse découplée d'une liaison arbre/moyeu.

transmission (une entrée et une sortie) par denture d'engrenage. Le guidage en rotation est réalisé par une combinaison de liaisons rotule et linéaire annulaire disposées de part et d'autre d'une des deux dentures. Les efforts extérieurs appliqués aux dentures correspondent à un couple sur l'arbre de 80 N.m (soit une puissance transmise de l'ordre de 17 kW à 2000 tr/min). Le CdC global correspondant à ce problème est celui de la figure 3.8. Nous nous attachons à la définition d'une solution technologique visant à remplir la fonction de liaison complète sur la denture C. Pour les trois DTLM envisagés, nous avons retenu un arrêt axial à droite par épaulement, compatible avec un montage par la gauche. L'objectif retenu consiste à minimiser la largeur du DTLM. L'encombrement radial maximum admissible (cf. figure 4.18) est directement lié aux dimensions de la denture d'engrenage (cf. figure 3.8). Nous avons relaxé les contraintes sur les diamètres des tronçons d'arbre (de part et d'autre du DTLM) en affectant la plus grande valeur possible (resp. plus petite valeur) aux diamètres maximums (resp. minimums) admissibles. La partie commune du cahier des charges de la liaison est représentée sur la figure par le biais du menu de définition commun.

4.4.4.2 Étude de la liaison complète par clavette.

Le DTLM testé est architecturé autour d'une clavette de forme A et de sa rainure associée. Les arrêts axiaux sont confiés à un épaulement à droite et un anneau élastique à gauche. Le matériau de la clavette ainsi que la qualité d'usinage sont également présents dans le menu de définition caractéristique représenté sur la figure 4.19. Sur cette figure est également représenté le résultat du calcul sous forme d'un modèle 3D pour lequel nous n'avons retenu que les dimensions significatives : la largeur totale du DTLM, un diamètre d'arbre (côté montage) et le diamètre de moyeu.

4.4.4.3 Étude de la liaison complète par cannelures.

Nous avons sélectionné un DTLM réalisant une liaison complète par cannelures dont les caractéristiques sont données par le menu de définition caractéristique de la figure 4.20. L'optimisation en largeur débouche sur une compacité axiale extrême (valeur bien inférieure à celle obtenue par

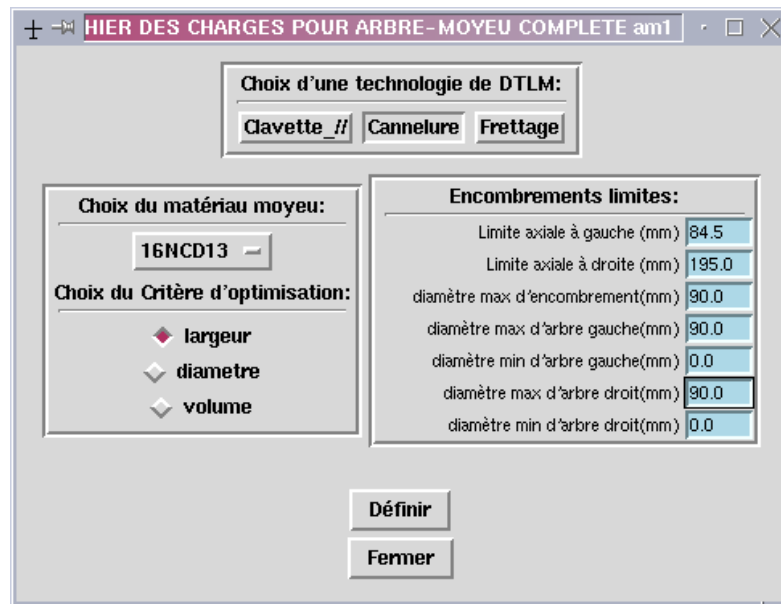


FIG. 4.18 – Menu de définition commun de la liaison complète arbre/moyeu.

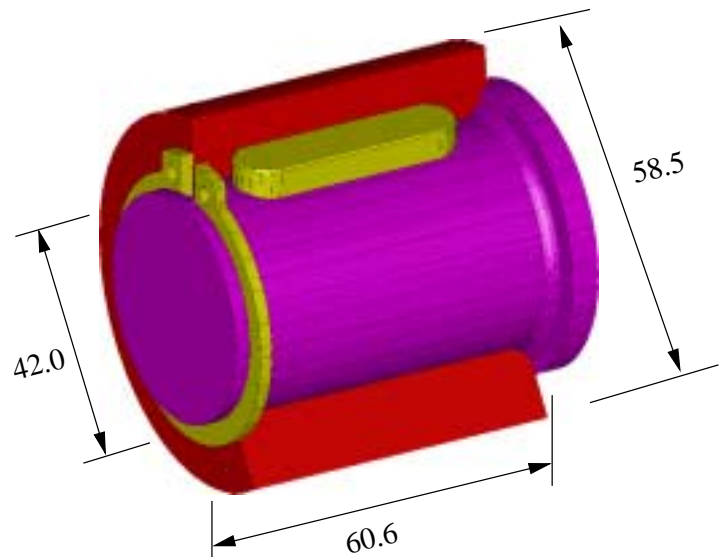


FIG. 4.19 – Menu de définition caractéristique du DTLM à base de clavette et modèle 3D résultant.

la solution à base de cannelures). La limite radiale n'est cependant pas atteinte car le processus de dimensionnement bute sur une longueur minimale de cannelure (paramètre lié au taillage). Le module proposé est de faible dimension (1 mm) ce qui a pour effet de limiter la hauteur des cannelures et d'en multiplier le nombre.

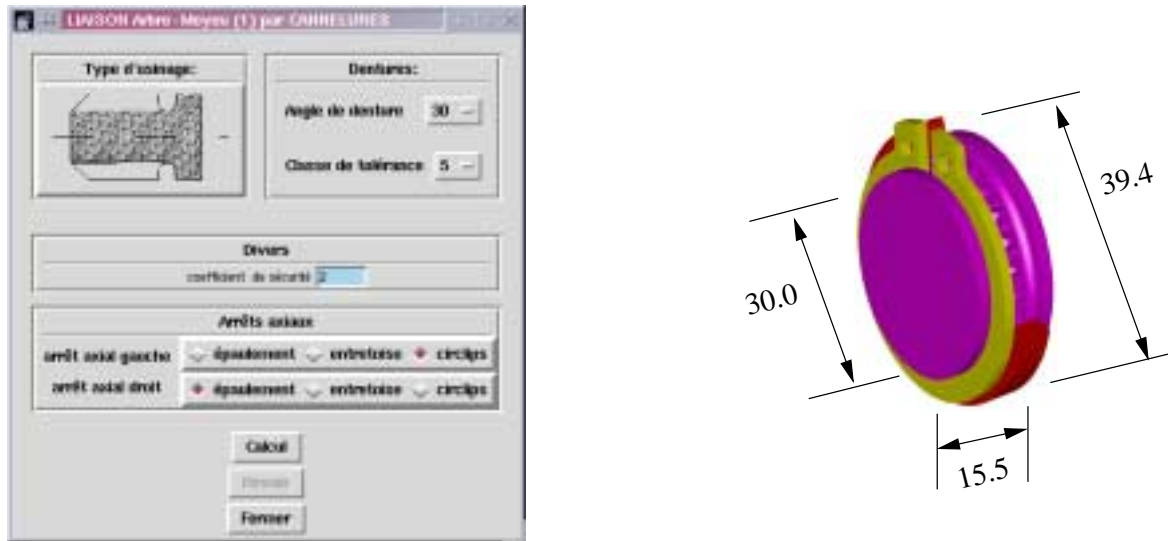


FIG. 4.20 – Menu de définition caractéristique du DTLM à base de cannelures et modèle 3D résultant.

4.4.4.4 Étude de la liaison complète par frettage.

La dernière solution testée est basée sur la technique de frettage. Le menu de définition caractéristique présenté sur la figure 4.21 recense les paramètres technologiques de la liaison. Le concepteur a retenu un alésage normal de qualité 7 et un ajustement de qualité équivalente pour l'arbre. Les résultats obtenus donnent un serrage minimum de $39\ \mu\text{m}$ et maximum de $75\ \mu\text{m}$ correspondant à un ajustement de type z7 pour un arbre de diamètre 18 mm (le type d'ajustement fourni par le module de calcul n'est normalisé qu'en tolérance, et non en écart fondamental). Il est évident qu'en limitant les réponses à des ajustements « courants », nous aurions sensiblement modifié les résultats du calcul. Afin de conserver une échelle de dimensions identique pour les représentations 3D, nous avons volontairement tronqué le moyeu. En effet, son diamètre extérieur, égal à la limite autorisée (90 mm), ne permettait pas, dans le cadre de ce document, une représentation complète.

4.4.4.5 Observations.

Les résultats obtenus montrent des différences significatives sur le plan dimensionnel. Il convient de noter que la hiérarchie qui semble se dégager de cet exemple (si nous classons les solutions par importance d'encombrement axial) n'est valable que dans ce cas de figure précis. Il serait maladroit de vouloir étendre ces observations à d'autres cas d'étude (optimisation en volume, contraintes actives sur les diamètres d'arbre, etc.). Notre propos n'est pas de vouloir dégager une quelconque prépondérance de telle ou telle technologie dans un cas d'étude précis. Les résultats obtenus dépendent fortement des méthodes de calcul mises en œuvre dans les modules de dimensionnement des DTLM. Une évolution de la complexité des modèles, un changement dans les hypothèses de calcul, ne conduiraient pas obligatoirement à des résultats similaires.

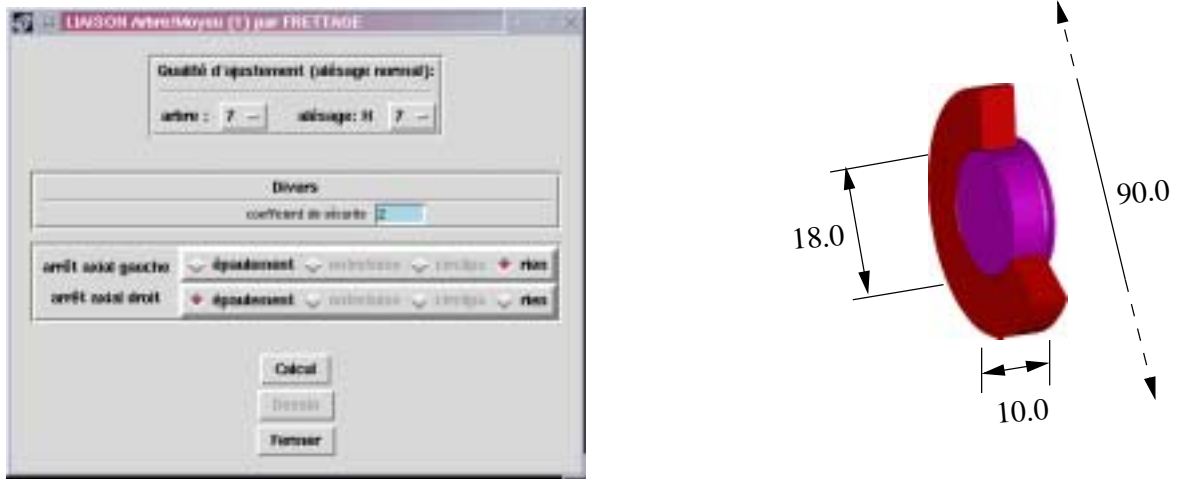


FIG. 4.21 – Menu de définition caractéristique du DTLM à base de frettage et modèle 3D résultant.

Nous devons plutôt nous intéresser à l'intérêt que représente la possibilité de tester, à volonté, les solutions technologiques pour une même liaison. La souplesse introduite par le concept de DTLM permet de rapidement mener des investigations en libérant le concepteur des contraintes inhérentes aux calculs. La structure orientée objet favorise la conservation des informations communes aux différents DTLM, c'est-à-dire propres à la fonction de liaison même. Le passage de l'un à l'autre des DTLM ne requiert que le renseignement des données propres à chaque technologie.

Nous avons volontairement laissé dans cet exemple un grand espace pour loger la liaison. L'introduction de contraintes sur les diamètres d'arbre, par le biais de contraintes de montage, ou la restriction de l'encombrement axial entraînerait inmanquablement l'apparition de nouvelles solutions. Le concepteur est ainsi libre d'expérimenter l'influence de ces contraintes sur les solutions générées.

La section suivante présente la démarche de synthèse découplée en abordant l'aspect de la gestion manuelle des contraintes.

4.5 Procédure de synthèse découplée de l'ensemble.

Le concepteur aborde la phase de synthèse découplée avec l'intention de définir les DTLM de toutes les liaisons du mécanisme. Son objectif est d'évaluer, séparément, les différentes solutions qu'il peut mettre en place pour chaque liaison. À partir des CdC, l'outil d'assistance calcule et optimise les dimensions des DTLM automatiquement, à la demande du concepteur. Si la chronologie de détermination des DTLM est théoriquement indifférente, nous avons vu que dans le cas particulier d'une liaison pivot non décomposable il était nécessaire de procéder à sa synthèse en premier lieu.

Une fois obtenus les premiers DTLM solutions de chaque liaison, leur prise en compte globale fera éventuellement apparaître des interférences géométriques. La phase de synthèse découplée doit permettre de résoudre manuellement ces interférences. Nous analyserons les moyens proposés par notre outil pour peser sur le processus de dimensionnement. Enfin, il nous semble indispensable de pouvoir apporter des éléments de choix au concepteur, sans pour autant le guider dans l'élaboration de sa solution. Cette aide doit être ponctuelle et pertinente. Nous en présentons un aperçu à la fin de cette section.

4.5.1 Gestion manuelle des contraintes.

Au fur et à mesure que les DTLM sont dimensionnés, ils prennent place dans le mécanisme. A cette occasion, le concepteur peut constater deux types d'interférences que nous détaillons en suivant : les interférences axiales et les interférences de montage.

4.5.1.1 Les interférences axiales.

Elles sont caractérisées par un chevauchement des volumes occupés par chacun des DTLM. La figure 4.22 illustre un cas d'interférences axiales entre les DTLM 2, 3 et 4. Ces conflits surviennent

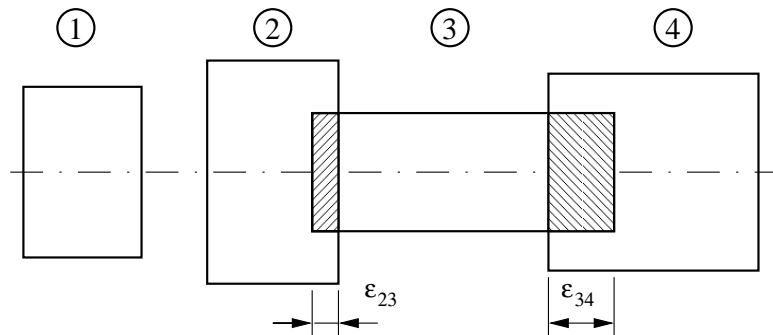


FIG. 4.22 – Interférences axiales entre DTLM.

lorsque le concepteur n'a pas borné strictement l'espace alloué aux DTLM dans les cahiers des charges des liaisons. Nous rappelons que l'espace alloué axialement à chaque DTLM est défini par les paramètres *lag* (limite à gauche) et *lad* (limite à droite) (cf. figure 2.16) qui sont fixés dans les CdC liaisons présentés ci-avant.

Afin d'éviter au concepteur une navigation fastidieuse entre les différentes fenêtres de menu de définition commun des DTLM, nous avons regroupé les informations nécessaires à la gestion des interférences axiales dans une fenêtre unique nommée *topologie du mécanisme*. Cette fenêtre est la seconde fenêtre permanente de notre outil (elle demeure toujours active). Elle permet de visualiser les valeurs des limites d'encombrement axiales des DTLM mais aussi de les modifier. Les positions des centres géométriques de liaison *y* sont également représentées. Les limites sont représentées par des bandes sombres dont les extrémités délimitent l'espace axial alloué aux DTLM. Tous les éléments auxquels le concepteur doit avoir accès dans cette phase de la synthèse du mécanisme sont manipulables au sein de la fenêtre à l'aide de la souris. L'opérateur peut ainsi modifier les valeurs des paramètres qui seront automatiquement mises à jour dans les fenêtres CdC appropriées (la réciproque est également vraie). L'opérateur bénéficie cependant d'une meilleure résolution de définition des valeurs par le biais des zones de capture³ des fenêtres CdC.

La figure 4.24 représente une fenêtre de topologie caractéristique d'un cahier des charges que nous qualifierons de « risqué ». L'espace axial laissé aux DTLM des liaisons A et C ne comporte pas de recouvrement mutuel. Les DTLM fournis par les modules de calcul ne risquent donc pas d'interférer. A l'inverse, la définition des liaisons C, B et D peut mener à des conflits si les DTLM solutions viennent à occuper intégralement l'espace alloué.

Bien que source de conflits, ce genre de pratique ne doit pas être proscrit. En ouvrant largement le domaine alloué à chaque DTLM, le concepteur élargit la gamme des solutions pour

³. Zones de saisie de texte ou de valeurs numériques (la résolution de la fenêtre n'est que de l'ordre du millimètre).

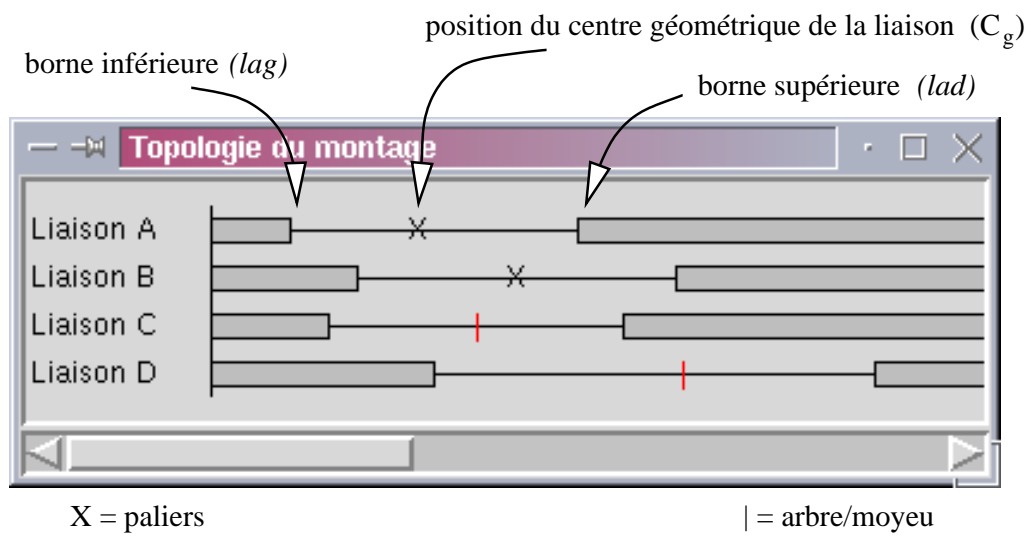


FIG. 4.23 – Fenêtre de topologie du mécanisme.

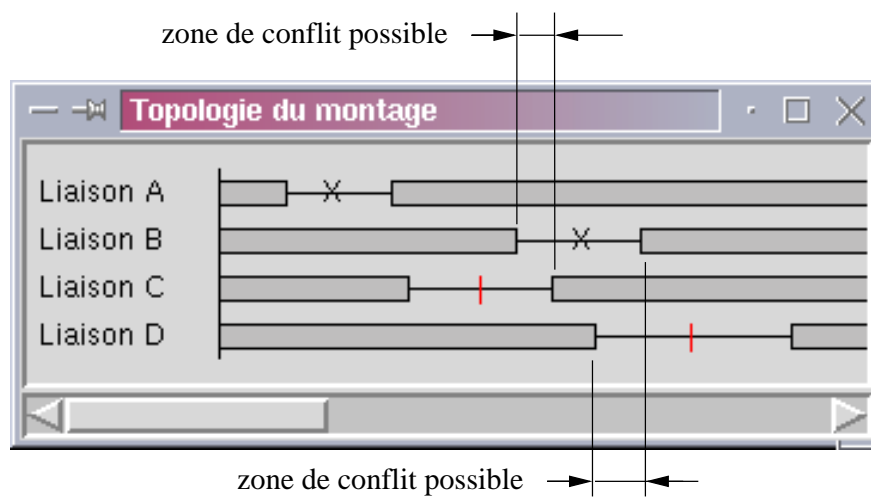


FIG. 4.24 – Exemple de définition « à risque » des espaces de DTLM.

chaque liaison. Il augmente ainsi le nombre de combinaisons de DTLM parmi lesquelles pourrait se trouver la combinaison optimale. Par contre, en réduisant délibérément l'espace alloué, le concepteur peut se priver de DTLM solutions qui mèneraient à une combinaison optimale. Il faut donc jouer intelligemment de ces bornes pour donner toute latitude aux modules de calcul de fournir une bonne collection de DTLM. Nous insistons sur l'intérêt de pouvoir expérimenter indépendamment sur chaque liaison l'influence des éléments du cahier des charges. Le concepteur peut ainsi déterminer quel sera le (ou les) DTLM « clé » pour la recherche d'une solution globale.

La démarche générale que nous proposons est la suivante :

1. Le concepteur choisit les DTLM pour chaque liaison et lance les calculs de dimensionnement en relaxant autant que possible les contraintes axiales.
2. Il évalue les résultats obtenus. En jouant sur les critères d'optimisation, il peut, moyennant une relance des calculs, déterminer les dimensions extrêmes des DTLM (diamètre minimal, longueur minimale, volume minimal). Ceci lui permet de déterminer la marge de manœuvre propre à chaque DTLM et d'en déduire un positionnement des frontières communes entre DTLM.
3. Il repositionne les limites axiales d'encombrement en accord avec les observations précédentes. Lorsqu'un module de calcul ne fournit plus de solution, il relaxe les contraintes du dit DTLM et tente de contraindre le DTLM voisin.

Quelques remarques. Nous avons constaté précédemment que les DTLM présentent généralement une symétrie par rapport à leur centre géométrique de liaison. Ceci implique une quasi-symétrie des limites axiales. Le concepteur doit donc en tenir compte lorsqu'il manipule les limites axiales pour « ouvrir » ou « fermer » un domaine de recherche. Considérons l'exemple de la figure 4.25. Les limites axiales fixées dans les CdC liaisons conduisent à la solution conflictuelle représentée en (a). Nous avons représenté en (b) et (c) les options extrêmes de résolution de conflit : dans le cas (b), le concepteur fixe les frontières communes C/B et B/D en pénalisant respectivement C et D. Remarquons alors que la modification de la limite droite du DTLM C entraîne de facto une modification de la limite gauche (rappelons que le point de référence du DTLM doit rester confondu avec le centre géométrique de la liaison⁴). Nous devons donc faire face à la modification dimensionnelle de deux DTLM pour lesquels la réduction de longueur est équivalente à deux fois la violation rencontrée ($2 \varepsilon_{CB}$ pour C et $2 \varepsilon_{BD}$ pour D)! Cela peut conduire à une absence de solution. A l'inverse, dans le cas (c), la pénalisation de B n'entraîne qu'une réduction de longueur sur B de valeur égale au double de la plus grande violation ($2 \max(\varepsilon_{CB}, \varepsilon_{BD})$). Bien entendu, toutes les solutions intermédiaires sont possibles, à savoir le placement des frontières communes entre les positions extrêmes que nous avons représentées.

Tout ceci laisse entrevoir un processus itératif de recherche de solutions qui peut être long et fastidieux. Les heuristiques potentielles peuvent être variées. Nous allons voir que l'ajout des contraintes de montage complique sensiblement la tâche du concepteur.

4.5.1.2 Les interférences de montage.

Elles correspondent à un étagement des diamètres d'arbre incohérent avec le sens de montage. Ces derniers sont issus des règles de montage édictées lors des tâches précédentes, dont il sont une expression simplifiée. Ils concernent les éléments (roulements, roues dentées, etc.) qui doivent prendre place sur l'arbre. Littéralement, ces règles correspondent aux expressions «...se monte

4. Les encombrements diamétraux ne sont pas représentatifs de la réalité.

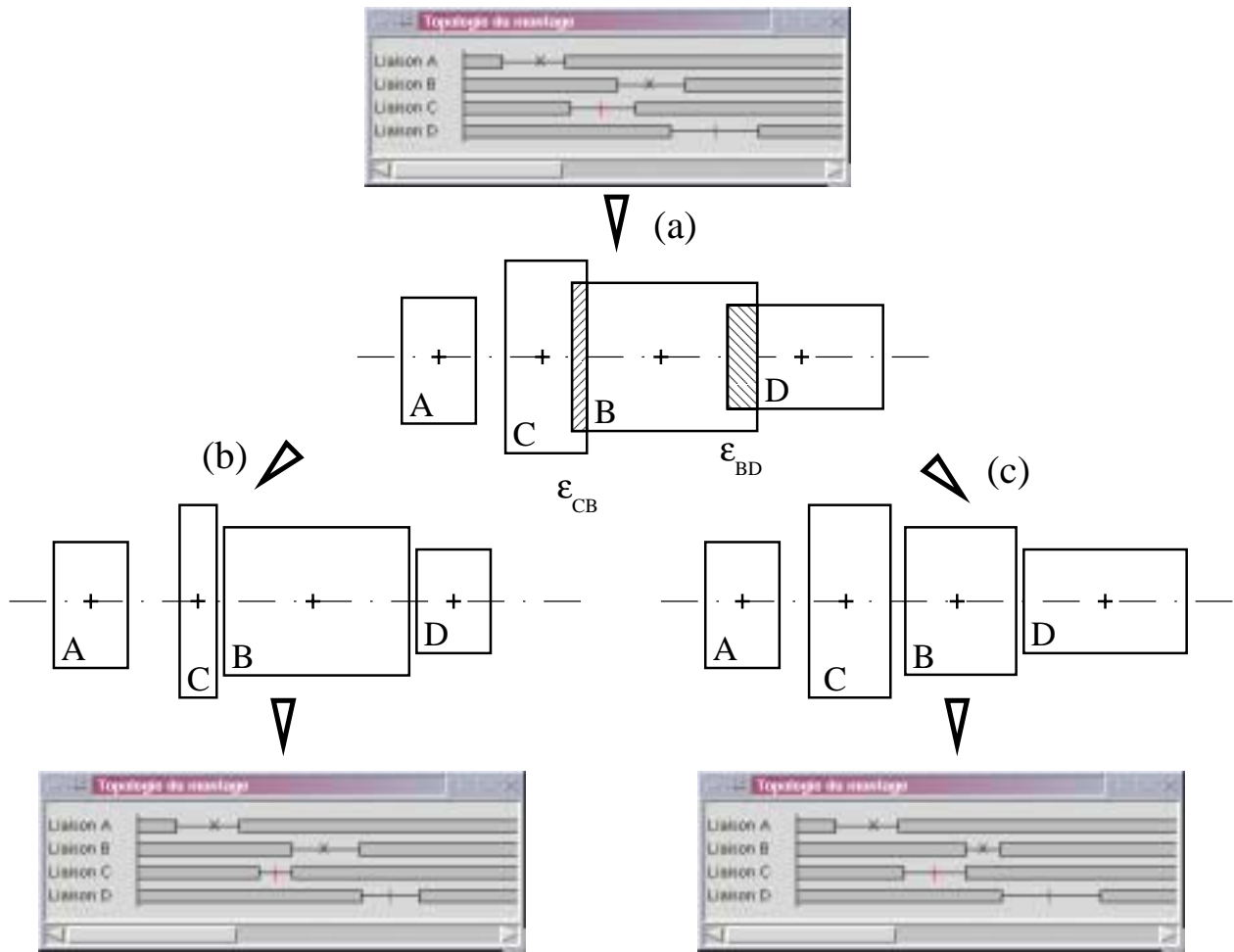


FIG. 4.25 – Exemple de résolution manuelle de conflits axiaux.

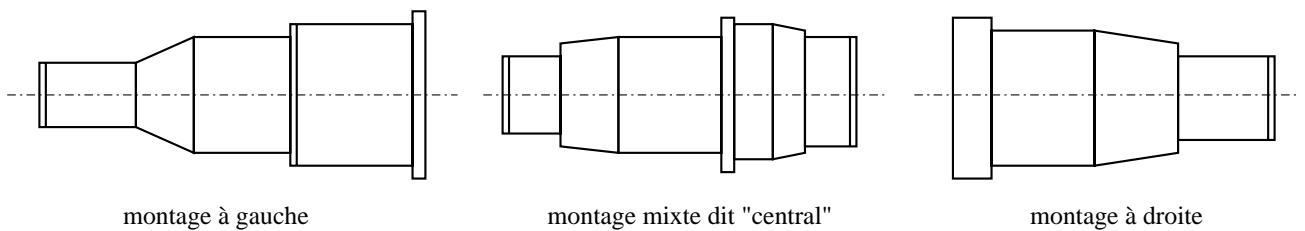


FIG. 4.26 – Géométries d'arbres liées aux sens de montage des DTLM.

par la gauche... » ou « ...se monte par la droite... ». Le concepteur définit un *point haut* sur l'arbre qui matérialise le point de convergence, ou point central d'arrêt, du montage des divers éléments. Ce point peut se situer sur la partie centrale de l'arbre comme à ses extrémités. Il engendre alors les géométries d'arbre représentées sur la figure 4.26.

Nous avons présenté (cf. §2.2.2) les valeurs limites permettant d'exprimer les contraintes de montage associées à chaque DTLM. Il s'agit de valeurs minimales et maximales de diamètre d'arbre associées aux extrémités de chaque tronçon d'arbre des DTLM. Il paraît à priori inopportun de proposer le contrôle de ce genre de contraintes au concepteur durant la phase de synthèse découplée si son unique objectif est de tester *indépendamment* les différents DTLM. Néanmoins, il devient indispensable d'introduire ces contraintes dans le processus de calcul dès lors que l'on veut assembler les DTLM. Ces contraintes de montage sont donc disponibles et initialisées à une valeur nulle pour les diamètres minimums et une valeur infiniment grande pour les diamètres maximums (elles sont ainsi neutralisées). Certains points doivent cependant être considérés :

- L'évolution des diamètres d'arbre est intimement liée à l'évolution du diamètre extérieur du moyeu, donc à l'encombrement diamétral du DTLM. Imposer des contraintes sur les diamètres d'arbre ainsi que sur le diamètre de moyeu peut s'avérer redondant voire parfois contradictoire.
- Le couplage des contraintes d'encombrement et de montage peut, dans certains cas complexes, s'avérer rapidement incompréhensible pour le concepteur. Ces deux types de contraintes sont parfois considérées comme *redondantes* lorsque la longueur et le diamètre d'arbre sont liés entre eux par des équations fonctionnelles ou implicites (transmission d'efforts dans le cas des liaisons complètes, dimensions intrinsèques des composants dans le cas des liaisons rotule, linéaire annulaire, etc). Cependant, la compréhension de ces couplages reste difficile et leur prise en compte malaisée. La multiplication des paramètres de contrôle (encombrement *et* montage) permet donc de diriger plus finement la recherche de solution. Ceci est particulièrement vrai dans des cas simples, i.e. faiblement contraints, où la modification d'une unique contrainte peut mener à une solution.
- Le concepteur dispose de valeurs limites minimum et maximum pour chaque côté du tronçon d'arbre. Il convient de rappeler qu'une seule de ces limites est utile de chaque côté de l'arbre. Ceci est dû au fait que la contrainte de montage n'impose qu'une relation unilatérale à chacun des DTLM concernés. Cette relation se traduit par une inéquation (cf. §2.2.2).

En conséquence, l'introduction du contrôle des contraintes de montage dans le processus de synthèse découplée permet, dans des cas simples, de pouvoir mener à bien une synthèse globale du mécanisme. Il existe des propriétés qui sont de nature à faciliter la gestion commune des contraintes de montage et d'encombrement.

Nous avons mis en évidence des relations entre l'évolution des dimensions axiales et radiales. Ces relations sont qualitatives ; il s'agit de tendances générales que le concepteur peut mettre à profit de la façon suivante. Nous distinguons dans leur comportement deux catégories de DTLM. La première (qui regroupe tous les DTLM de liaison complète arbre/moyeu) est basée sur la caractéristique de réciprocity inverse d'évolution des dimensions axiales et radiales. En effet, lorsque le concepteur tend à réduire l'espace axial alloué au DTLM, celui-ci compense grâce à une expansion radiale. Cela s'explique parfaitement par la *dualité* des dimensions intervenant dans les calculs de transmission d'effort. A surface constante, la diminution d'une dimension entraîne l'augmentation de l'autre (exemple des liaisons par clavette, cannelures ou frettage).

Les liaisons pivot à base de roulements présentent un tout autre comportement. L'évolution des dimensions radiales est proportionnelle à celle des dimensions axiales. Ceci est dû à la technologie même des composants : les éléments roulants (billes, rouleaux...) subissent des variations dimensionnelles isotropes. Cette affirmation est valable pour les roulements d'une même série. La prise en compte de plusieurs séries de roulements (larges, étroites...) génère l'apparition de plusieurs composants solutions pour un même diamètre. Néanmoins, la tendance d'évolution générale reste vraie.

4.5.1.3 Pertinence de la gestion manuelle des contraintes.

La méthode de gestion manuelle des contraintes est assez rudimentaire. Elle permet cependant un contrôle total du concepteur sur le dimensionnement des DTLM. De proche en proche, le concepteur réduit son espace des solutions pour chaque DTLM en tendant vers une solution globale. Cette solution, il l'a perçue en déclinant individuellement les différentes solutions concurrentes pour chaque DTLM, et en visualisant les dimensions extrêmes de chacun d'eux, donc leur domaine d'existence. Le processus que met en place le concepteur implique sa réactivité face aux solutions proposées par les modules de calcul. Automatiser un tel processus semble impossible sans devoir modéliser le comportement intelligent du concepteur (ce qui est hors de notre propos).

Le concepteur procède donc par itérations successives, avec la possibilité d'effectuer des retours en arrière, comme par exemple relaxer des contraintes trop fortes. La gestion manuelle présente aussi l'avantage de faire progresser la conception à des rythmes différents. Cette vitesse d'évolution du processus est liée aux variations (faibles ou importantes) que le concepteur applique aux contraintes. Il peut donc progresser rapidement au début de la phase de synthèse découplée, puis affiner la résolution de ses contraintes en approchant d'une solution finale.

4.5.2 Aide au choix des DTLM.

La gestion manuelle des conflits peut être facilitée par un certain nombre d'aides ciblées. Ces aides n'imposent en aucun cas des règles de choix ou une quelconque expertise au concepteur. Elles ont pour seul but d'informer, de la façon la plus pertinente, sur l'état du processus de conception.

La première des aides que l'on peut apporter au concepteur concerne les sollicitations dans l'arbre. Lorsque le concepteur sélectionne un DTLM, il le sélectionne en fonction de la technologie de ses constituants. Cette technologie suppose des aménagements de forme dans les différentes pièces. Le choix d'une clavette ou d'un anneau élastique par exemple n'est pas innocent. Il doit tenir compte des accidents de forme (gorge dans le cas de l'anneau élastique) provoqués dans l'arbre qui, compte tenu des sollicitations de l'arbre, risquent de pénaliser le dimensionnement du DTLM. Nous pouvons, dès qu'elles sont disponibles, proposer au concepteur la visualisation de ces sollicitations. Nous avons retenu une modélisation graphique des moments de torsion et flexion de l'arbre, comme représenté sur la figure 4.27. Ces informations sont connues à l'issue de la détermination de la topologie de la liaison pivot ou du dimensionnement des DTLM de la pivot dans le cas de contacts obliques. Le concepteur a le loisir de consulter cette fenêtre pour repérer les zones fortement sollicitées et choisir en conséquence une technologie de DTLM adaptée. Ce type d'assistance ne propose pas de solution. Son unique but est de donner une représentation des résultats intermédiaires du processus de synthèse, en l'occurrence les sollicitations dans l'arbre, au plus tôt, c'est-à-dire dès qu'elles sont disponibles !

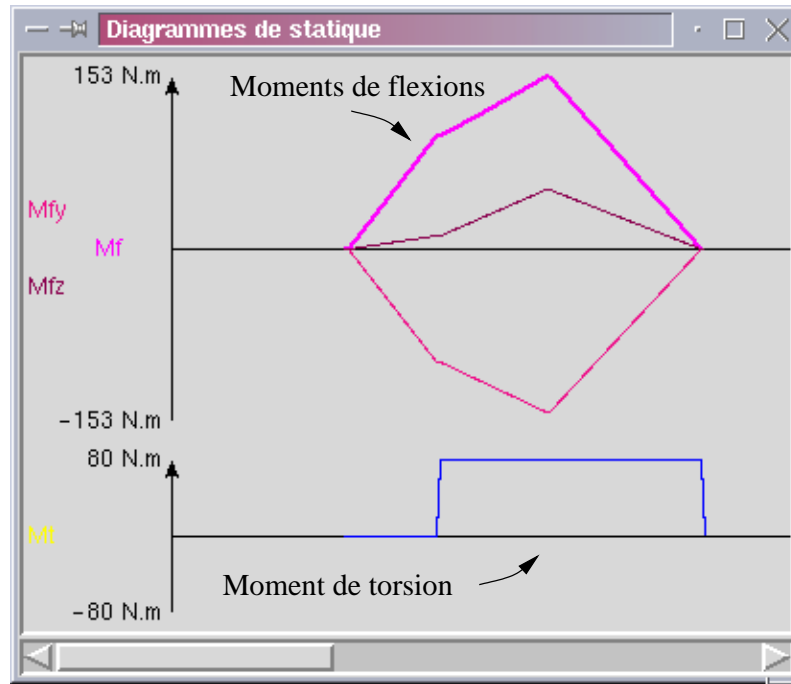


FIG. 4.27 – Diagrammes des moments.

Un autre type d'assistance est proposé au concepteur dans la phase de synthèse découplée. Nous avons discuté d'une méthodologie de travail (cf. §4.5.1) propre à la synthèse découplée. Nous y avons jugé de l'intérêt de la phase de synthèse découplée dans l'évaluation de l'ensemble des solutions concurrentes pour un même DTLM. Il est apparu que la possibilité de matérialiser les encombrements extrêmes de toutes les solutions d'un DTLM constituait un élément important de décision dans la recherche d'une solution globale. Nous avons donc imaginé une aide à la visualisation de ces solutions. Elle est rendue possible par l'emploi d'une procédure de recherche énumérative des DTLM solutions (cf. §2.3.1.4). L'assistance que propose la fenêtre suivante (figure 4.28) est destinée à la représentation, sous forme schématique, de l'encombrement de l'ensemble des solutions dimensionnelles de chacun des DTLM. Ceci s'explique de la façon suivante. Pour chaque DTLM choisi, un CdC local est établi. Au problème local ainsi formulé peut correspondre une multitude de solutions dimensionnelles, c'est-à-dire une multitude de DTLM technologiquement identiques mais présentant des dimensions différentes. Ces solutions respectent cependant toutes les limites imposées par les CdC. Bien entendu, une seule de ces solutions est optimale au sens de l'objectif qui aura été choisi (et c'est celle que le concepteur a le loisir de visualiser en représentation 3D). La fenêtre d'assistance présentée a pour but de reproduire rapidement l'ensemble de ces solutions. Le concepteur évalue alors facilement la quantité de solutions potentielles répondant au CdC. L'analyse de l'image proposée sur la figure 4.28 conduit aux remarques suivantes :

- Deux DTLM sont susceptibles de provoquer des interférences axiales (DTLM B et C).
- Certains DTLM présentent de fortes variations de longueur et/ou de diamètre (DTLM B et C).
- Pour chaque DTLM, il existe un nombre important de solutions réalisables (caractéristiques d'un domaine de solutions étendu). Le concepteur va donc pouvoir durcir les contraintes pour réduire le nombre de solutions réalisables et tendre vers une solution globale qu'il considérera comme optimale (les objectifs locaux pouvant être variés). On peut cependant

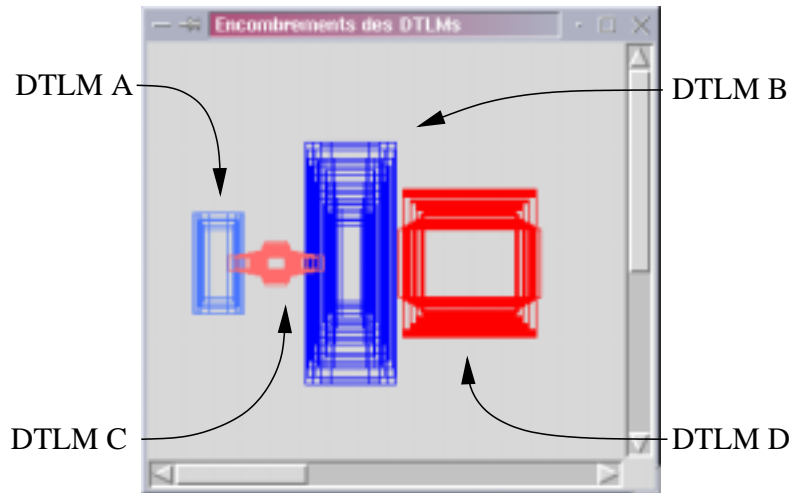


FIG. 4.28 – Fenêtre de visualisation de l'encombrement de tous les DTLM solutions.

constater que certains DTLM présentent des solutions très proches (en terme d'encombrement) les unes des autres. C'est notamment le cas des DTLM A et D de notre exemple. Le concepteur aura donc le loisir de balayer plus finement l'espace des solutions réalisables en jouant sur les contraintes. Mais ce n'est vraisemblablement pas sur ces DTLM que le concepteur gagnera le plus d'espace en vue d'éliminer les conflits avec les DTLM voisins.

Dans le même ordre d'idée, nous proposons une fenêtre d'aspect similaire qui autorise non plus la visualisation des encombrements, mais celle des diamètres d'arbre. Le concepteur peut ainsi évaluer les DTLM concurrents sur le critère des contraintes de montage. Il lui est alors

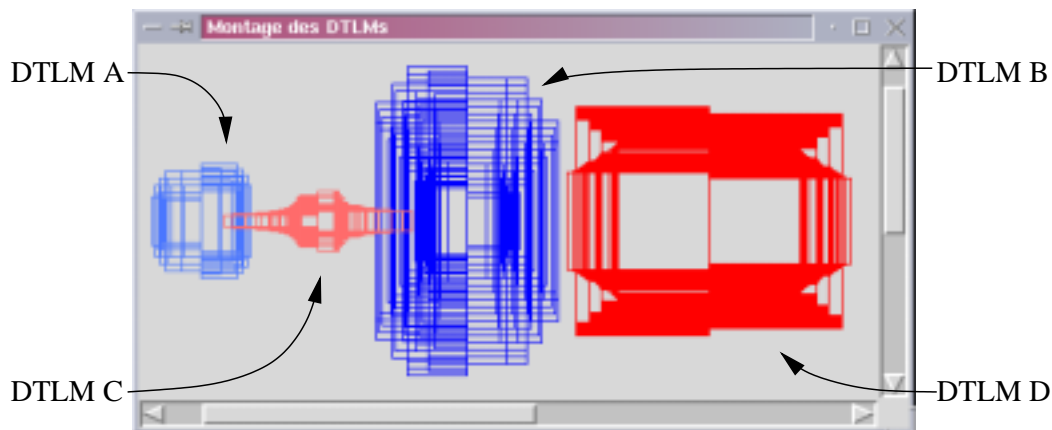


FIG. 4.29 – Fenêtre de visualisation des diamètres d'arbre de tous les DTLM solutions.

facile de déterminer si les choix de chaque DTLM peuvent mener à une solution globale en respectant les règles de montage. On constate clairement sur la figure 4.29 que le DTLM C risque de provoquer des interférences de montage. On conçoit aussi rapidement que le point central de montage ne pourra se situer qu'entre les DTLM C et B (ceci semble confirmé par l'orientation des épaulements).

Ces éléments d'aide sont perfectibles, notamment sur le plan graphique. Ils ne sont pas indispensables au concepteur dans la démarche de synthèse découplée. Une partie des informations qu'ils fournissent au concepteur sont également accessibles par le biais des modèles 3D. De plus, l'information fournie est essentiellement qualitative, principalement à cause de la représentation

schématique adoptée. Cependant, ces éléments d'aide présentent l'avantage de centraliser les informations sur un même graphique afin que le concepteur puisse, d'un coup d'oeil, orienter sa stratégie de recherche de solution. En visualisant grossièrement l'étendue des domaines de solutions, le concepteur acquiert des informations qui lui permettent de modifier les contraintes là où il estime que le gain (en terme de réduction des conflits) sera le plus sensible.

4.5.3 Exemple de synthèse découplée.

Considérons le problème dont les CdC sont fournis en annexes (cf. annexes A). La première passe de calcul sur chacun des DTLM fournit une solution globale conflictuelle à plusieurs titres, récapitulée dans le tableau 4.1. Nous y avons fait figurer les variables d'encombrement axial (bg et bd), les diamètres d'arbre (dag et dad) et l'objectif local que nous avons nommé F_i . En

	DTLM 1	DTLM 2	DTLM 3	DTLM 4
bg_i	74	102,3	178,5	246,7
bd_i	107,5	175,7	213,5	307,3
dag_i	14	26	28	48,6
dad_i	17	32,7	20	42
F_i	33,5	26	20	60,6
	synthèse découplée			

TAB. 4.1 – Résultats obtenus à l'issue de la première passe de calcul.

gras sont distinguées les valeurs conflictuelles. La représentation des diamètres d'arbre limitée à la meilleure solution de chaque DTLM est donnée sur la figure 4.30(a). Le concepteur décide

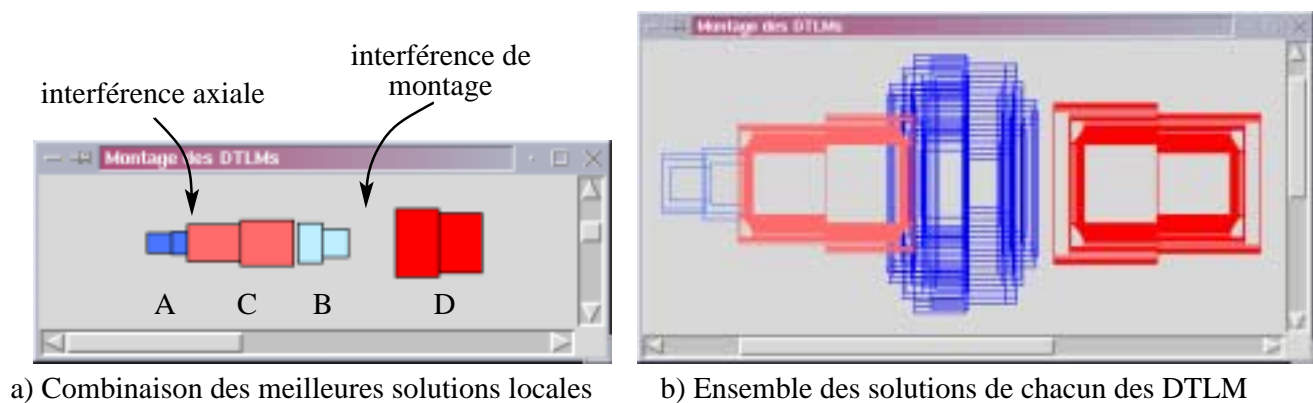


FIG. 4.30 – Résultat (tronçons d'arbre) d'une première passe de calculs.

de prendre en charge le conflit sur les encombrements axiaux des DTLM A et C. Pour ce faire, il évalue l'étendue du domaine des solutions réalisables de chacun des DTLM, grâce à la figure 4.30(b), afin de voir si ces conflits ne sont pas insurmontables. Il restreint alors les encombrements axiaux des DTLM A et C en imposant de nouvelles valeurs limites. L'opération semble profitable puisqu'il obtient les propositions de la figure 4.31. Cependant, on constate que la combinaison des meilleures solutions (fenêtre de droite sur la figure) présente toujours un conflit de montage (entre B et D). Ceci est principalement dû au fait que l'objectif d'optimisation retenu pour le DTLM B est la minimisation du diamètre. Le concepteur doit maintenant résoudre le conflit de montage entre les DTLM B et D. Il modifie pour cela la valeur limite sur le diamètre d'arbre à

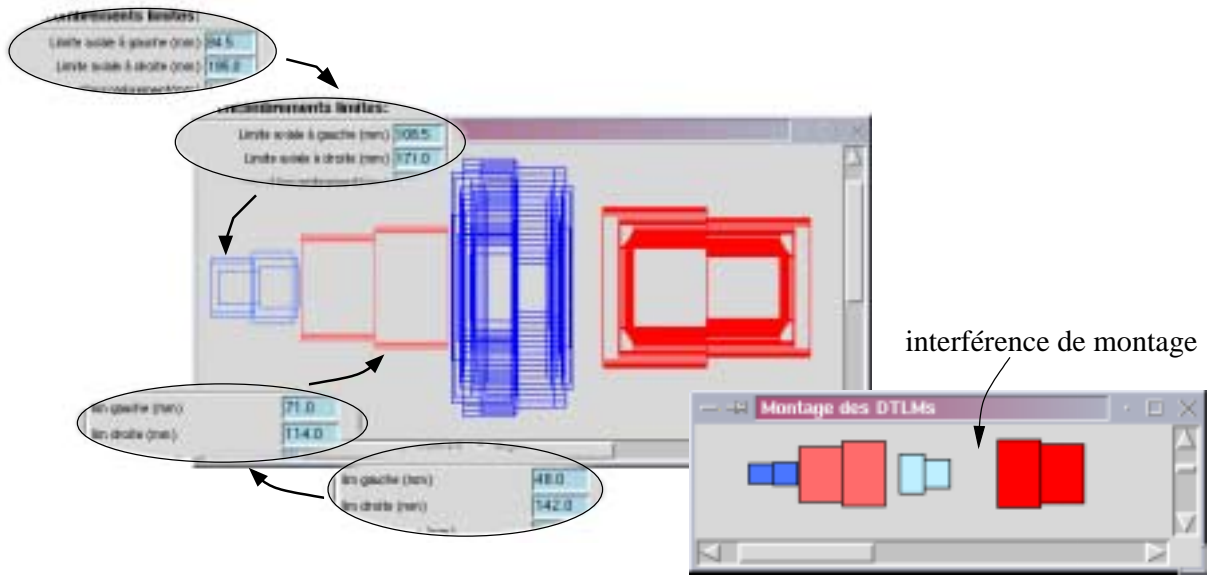


FIG. 4.31 – Après restriction des encombrements axiaux.

droite du DTLM B. Il pourrait, tout aussi bien tenter de diminuer le diamètre d'arbre à gauche du DTLM D, voire même un peu des deux méthodes. Son choix conduit à la solution de la figure 4.32.

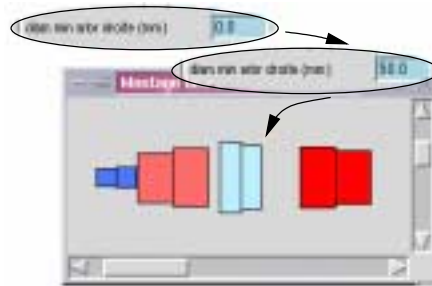


FIG. 4.32 – Après modification de la valeur limite sur le diamètre d'arbre.

La solution obtenue est réalisable, c'est-à-dire qu'elle ne présente plus de conflit. On ne peut cependant affirmer qu'il s'agit de la solution optimale globale du problème initial. En effet, cette solution est optimale au vue du dernier cahier des charges imposé, c'est-à-dire celui comportant les contraintes modifiées. Il est important de noter que ces contraintes ne figuraient pas dans les CdC initiaux ou n'avaient pas les mêmes valeurs. Le concepteur a donc *orienté* la recherche d'une solution dans une voie qu'il a jugé digne d'intérêt mais ce faisant, il s'est peut-être éloigné de la solution optimale globale. Seule une recherche exhaustive peut nous indiquer la qualité de cette solution.

Nous avons donc développé une méthode énumérative. Celle-ci prend appui sur les résultats des modules de calculs. Ces derniers ont été conçus pour fournir, en plus de la solution optimale, la liste complète des DTLM *satisfaisant le cahier des charges* sous forme d'un fichier texte. Ainsi, lorsque le concepteur lance une première fois les calculs pour chaque liaison, les modules de calcul retournent une liste des DTLM solutions accompagnés de leur valeur d'objectif local (notées F_i). Les objectifs locaux des meilleurs DTLM de chaque liste (notés F_{opt_i}) sont mémorisés par l'algorithme d'énumération. Ce dernier puise alors dans ces fichiers-liste de DTLM et combine

toutes les solutions afin d'isoler les solutions globales réalisables (ne présentant plus de conflits). Il en extrait la solution globale optimale en employant une fonction d'évaluation multicritère, que nous notons f , formulée de la façon suivante :

$$f = \sum_{i=1}^4 \left(\frac{F_i - F_{opt_i}}{F_{opt_i}} \right)^2 \quad (4.1)$$

Les termes F_{opt_i} sont donc des valeurs cibles d'objectif local optimal que le processus ne pourra, au mieux, qu'égaliser. Dans l'exemple traité ci-dessus, les F_{opt_i} ont pour valeurs les F_i du tableau 4.1. La fonction d'évaluation doit être considérée comme une somme d'écart au carré, normalisés, que la méthode énumérative tente de minimiser.

DTLM	Nb. de solutions
A	3
B	46
C	23
D	29

TAB. 4.2 – Nombre de DTLM solutions pour chaque liaison.

La méthode énumérative a ainsi été appliquée au cahier des charges initial, i.e. avant toute modification de la part du concepteur, dans le but de connaître la solution globale optimale. Elle a évalué 92046 combinaisons, en moins d'une seconde, formées par les DTLM solutions (cf. tableau 4.2). La meilleure combinaison formant la solution globale optimale est donnée dans le tableau 4.3. Nous constatons, à la lecture de ce tableau, que la solution obtenue par synthèse découplée

	DTLM 1	DTLM 2	DTLM 3	DTLM 4	DTLM 1	DTLM 2	DTLM 3	DTLM 4
bg_i	74	108,57	179	246,7	71	105,85	173,5	242
bd_i	107,5	169,42	214,25	307,3	105	172,15	218,5	311,95
dag_i	14	39	57	48,6	25	30	45	34,7
dad_i	17	45,7	50	42	30	36,7	35	28
F_i	33,5	39	50	60,6	34	30	35	69,9
f	2,5				0,61			
	synthèse découplée				méthode énumérative			

TAB. 4.3 – Comparaison des résultats obtenus par synthèse découplée et méthode énumérative.

diffère de la solution globale optimale. Les DTLM centraux sont particulièrement éloignés de leurs valeurs optimales globales. On peut en déduire que la direction de recherche établie par le concepteur, bien que menant à une solution réalisable, n'a pas été la meilleure. Le choix, par exemple, d'augmenter le diamètre minimal du DTLM 3 n'était pas judicieux puisque son objectif est la minimisation du diamètre d'arbre !

4.6 Conclusions.

La phase de synthèse découplée est une première étape vers l'obtention d'une solution globale. Elle a pour objectif de faciliter l'émergence d'une solution proche de la solution optimale. Les cahiers des charges qui en résultent peuvent servir de point de départ à une démarche automatisée. Cette approche s'accompagne d'une gestion manuelle des contraintes à travers laquelle le concepteur contrôle la progression du processus.

Le concepteur est au centre du processus de synthèse découplée. Il fixe les contraintes pour chaque DTLM, lance les calculs et peut visualiser les résultats grâce aux modèles 3D. Il peut également accéder à des informations fragmentaires grâce aux fenêtres d'assistance présentées ci-dessus. La figure 4.33 représente les flux d'informations de la phase de synthèse découplée. Les

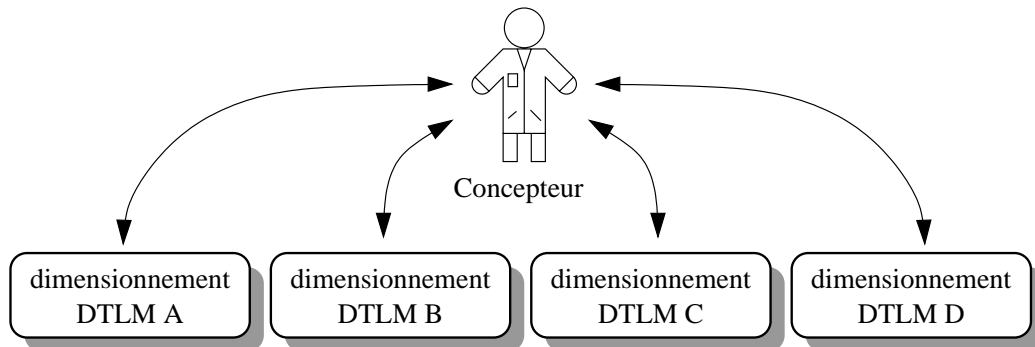


FIG. 4.33 – Modélisation des flux d'informations de la phase de synthèse découplée.

problèmes de dimensionnement sont totalement découplés les uns des autres car les contraintes de liaison existant entre les divers modules de dimensionnement de chaque DTLM sont prises en charge par le concepteur qui, en jouant sur les contraintes, tente de résoudre les conflits. En ce sens, le concepteur peut être considéré comme un processus maître ayant en charge la coordination des sous-problèmes d'optimisation.

Le découplage des problèmes de dimensionnement favorise l'usage de critères d'optimisation non identiques. Le concepteur arbitre directement les conflits, sans faire appel à des règles pré-établies de hiérarchisation des DTLM. Il peut ainsi, durant sa progression vers une solution globale, réviser à la hausse ou à la baisse, la prépondérance de certains DTLM compte tenu des éléments qui lui sont fournis (espace restant entre DTLM, nombre de DTLM solution...).

La synthèse découplée atteint cependant ses limites dans certaines conditions. Malgré la simplicité de la procédure de modification des contraintes, celle-ci peut se révéler fastidieuse pour des cas complexes. La raison en est une multiplication des phases de modification des contraintes dues à la nature itérative du processus essai-erreur mis en place. Les allers-retours entre les menus de définition et les résultats peuvent également alourdir la tâche lorsqu'ils sont en trop grand nombre (Ces manipulations gagneraient grandement à être automatisées). La multiplicité des contraintes de liaison, et en général de tous les paramètres de contrôle des modules de dimensionnement, conduit rapidement à une surcharge cognitive du concepteur. Par exemple, l'ajout de conditions de montage sur l'alésage induirait un supplément de contraintes de liaison entre les DTLM. Ces contraintes seraient autant de paramètres ajoutés à la liste de ceux que le concepteur doit déjà prendre en compte. La multiplication du nombre de DTLM dans le mécanisme étudié participerait également à l'inflation du nombre de paramètres. Ces propos sont cependant à modérer pour deux raisons :

- La nature itérative du processus de résolution limite la surcharge du concepteur puisqu'il modifie séquentiellement les DTLM, c'est-à-dire un par un.
- Les contraintes de liaison n'affectent que les DTLM voisins. Ainsi, dans le cadre de notre étude, soit les lignes d'arbre, un DTLM ne peut avoir que deux DTLM voisins. Le nombre de relations pour chaque DTLM est donc limité. Le concepteur doit cependant tenir compte de l'effet « domino » qui se traduit par une répercussion des modifications d'un DTLM sur tous les suivants.

Quel peut être l'intérêt d'un tel outil? La réponse tient en deux points principaux :

1. En début de processus, notre outil offre au concepteur la possibilité de dimensionner chaque DTLM de façon indépendante, sans préjuger d'un quelconque ordre de définition. En contraignant au minimum chaque DTLM, le concepteur obtient en résultat le maximum de ce que chaque DTLM peut offrir pour une liaison donnée. Par exemple, s'il décide de minimiser le diamètre d'une liaison complète par clavette, l'objectif sera d'autant mieux atteint que le DTLM ne sera pas contraint dans sa longueur, c'est-à-dire par des bornes axiales. Cette liberté d'investigation est très importante pour le concepteur. Elle lui permet de tester individuellement les différents DTLM pour un même cahier des charges (global et de liaison) et d'affiner son expertise face à des cas types (par exemple, « quel type de DTLM peut-on retenir dans le cas d'un encombrement axial réduit? »).
2. Cette approche n'interdit pas de mener ensuite une recherche de solution globale cohérente. En jouant sur les contraintes de chaque DTLM, le concepteur peut obtenir une solution cohérente dont il aura choisi les caractéristiques de façon dynamique. C'est à raison de multiples essais qu'il construira sa solution en gérant les différents conflits intervenants entre DTLM. L'outil lui facilite les calculs et met en avant les conflits grâce, notamment, à la modélisation 3D.

Le concepteur peut éprouver certaines difficultés à gérer conjointement les contraintes d'encombrement et de montage. Il peut, instinctivement, mettre en place certaines méthodologies que nous présentons dans le chapitre suivant, sans pour autant s'assurer de son succès. De plus, la synthèse manuelle s'avère fastidieuse lorsque l'on s'approche d'une situation où peu de DTLM restent en compétition. Il est alors souvent indispensable de procéder à des retours en arrière pour explorer d'autres combinaisons. C'est pourquoi nous avons étudié une démarche automatique de synthèse qui prend en charge la gestion de toutes les contraintes. Ce processus que nous appelons « synthèse coordonnée » est l'objet du chapitre suivant.

Chapitre 5

Synthèse coordonnée des DTLM.

Résumé

LA synthèse coordonnée fait suite à la phase de synthèse découplée. Elle a pour objectif de produire un mécanisme dimensionné de façon optimale. Elle s'appuie pour cela sur les cahiers des charges établis lors de l'étape d'initialisation (CdC global) et de synthèse découplée (CdC liaisons). Nous discutons dans ce chapitre des méthodes envisageables pour un tel processus automatique en relevant les particularités de notre problème. Nous présentons ensuite les méthodes que nous proposons pour coordonner les sous-problèmes locaux. Dans un premier temps, nous mettons en œuvre une méthode d'optimisation collaborative que nous adaptons au problème de synthèse des DTLM. N'étant pas entièrement satisfaits des résultats obtenus, nous présentons une méthode originale de relaxation sélective des contraintes. Celle-ci est testée avec des objectifs locaux hétérogènes, puis homogènes, afin d'en évaluer l'impact sur le processus. Tous les résultats obtenus par les différentes méthodes sont comparés à ceux issus d'une méthode purement énumérative.

Contenu du chapitre

5.1	Identification du problème de synthèse coordonnée.	129
5.1.1	Objectif.	129
5.1.2	Nature du problème.	129
5.1.3	Paramètres initiaux de la synthèse coordonnée.	130
5.2	Les systèmes décomposés et leur optimisation.	131
5.2.1	Les différents types de décomposition.	131
5.2.1.1	Décompositions hiérarchiques.	131
5.2.1.2	Décompositions non-hiérarchiques.	132
5.2.2	L'optimisation pluridisciplinaire (MDO).	133
5.2.3	Optimisation collaborative.	134
5.3	Application de l'optimisation collaborative à notre problème.	137

5.3.1	Formulation initiale du problème sous forme non-hiérarchique.	137
5.3.2	Problème global exprimé sous forme collaborative.	138
5.3.3	Choix d'une méthode pour l'implémentation du niveau système: l'algorithme de Powell.	141
5.3.4	Expression du problème du niveau système en problème sans contrainte.	142
5.3.5	Mise en œuvre de l'algorithme de Powell.	142
5.3.6	Influence du comportement des modules de calcul.	145
5.3.7	Conclusions.	149
5.4	Processus spécifique de synthèse coordonnée.	149
5.4.1	Présentation de la méthode.	149
5.4.2	Description de l'algorithme de relaxation sélective des contraintes.	151
5.4.2.1	Point de départ du processus.	151
5.4.2.2	Recherche d'une solution réalisable.	151
5.4.2.3	Optimisation d'une solution réalisable.	156
5.4.2.4	Remarques	158
5.4.3	Relaxation sélective avec un objectif multicritère.	159
5.4.4	Relaxation sélective avec des objectifs homogènes.	160
5.4.4.1	Caractérisation des problèmes.	160
5.4.4.2	Cas des problèmes semi-contraints.	162
5.4.4.3	Cas des problèmes sous-contraints.	163
5.4.4.4	Remarques.	166
5.5	Conclusions.	168

5.1 Identification du problème de synthèse coordonnée.

5.1.1 Objectif.

Le processus de synthèse coordonnée est une étape supplémentaire vers l'émergence d'une solution globale du problème. Il s'inscrit dans le cadre de la synthèse de DTLM en prenant place à la suite de la synthèse découplée. À ce titre, il ne doit pas remettre en cause la structure de notre outil présenté dans les chapitres précédents. Il ne doit pas non plus invalider les choix structurels effectués lors de la synthèse découplée. Enfin, il va de l'intérêt même du concepteur que cette étape ne se démarque pas trop fortement du reste du processus de synthèse. En privilégiant l'homogénéité du logiciel, nous pensons assurer une certaine ergonomie d'utilisation.

L'objectif de la synthèse coordonnée est de réaliser une optimisation globale de l'ensemble des DTLM en partant des éléments de choix fournis par le concepteur. Deux nouveaux aspects sont à prendre en compte :

- la gestion automatique des contraintes (interférences et montage) par le logiciel ;
- la prise en charge d'une optimisation globale.

5.1.2 Nature du problème.

Le dimensionnement des DTLM a été présenté au chapitre 2. Nous avons complété cette information par une présentation des contraintes d'assemblage et de montage au chapitre 4. Ces contraintes sont les seuls éléments tangibles dont nous disposons pour contrôler les opérations de dimensionnement de chaque DTLM. Établir un processus de synthèse coordonnée revient donc, en partie, à établir une stratégie de gestion de ces contraintes.

On pourrait penser que ce type de problème s'apparente à du placement (ou ordonnancement spatial). Ces problèmes se rencontrent en Génie Civil [Alz93] (aménagement de locaux, construction de bâtiments), en Génie Électrique (placement de circuits imprimés [Cho92]), en Génie Mécanique (placement de robots [Pam92] ou installation thermique [SCW98]) voire même dans le domaine de la confection [Mao95], en fait dans tous les domaines où la conception d'un système passe par un assemblage modulaire de blocs prédéfinis. Ils ont débouché sur des méthodes de résolution diverses et variées. La problématique consiste à positionner de façon optimale, sous contraintes, des blocs élémentaires composant un système. L'objectif souvent retenu est la compacité du système, et les contraintes peuvent être de natures diverses :

- encombrement des blocs ;
- agencement imposé de certains composants par les interfaces avec l'extérieur (bloc alimentation ou modules d'affichage en électronique, installations sanitaires à proximité d'un raccord d'eau, etc.) ;
- liaisons physiques ou fonctionnelles entre blocs (routage des circuits en électronique ou proximité de certains mobiliers en aménagement)...

Nous devons cependant relever deux *différences majeures* entre notre problème et ceux évoqués ci-avant :

- la position de nos composants (les DTLM) ainsi que leur agencement est déjà défini et immuable (c'est notre hypothèse de travail découlant de l'organisation opérationnelle présentée au chapitre 2) ;

- les dimensions de nos composants sont inconnues (puisqu’elles restent à définir) alors qu’elles sont préalablement déterminées dans les cas de placement.

Ces deux points nous éloignent fondamentalement des méthodes traditionnelles de placement [Alz93]. Nous avons donc dirigé nos recherches dans la voie d’un problème d’optimisation de système décomposé.

5.1.3 Paramètres initiaux de la synthèse coordonnée.

Le processus de synthèse coordonnée requiert quelques informations complémentaires à celles issues des étapes précédentes de la conception. Le concepteur est invité à renseigner ces paramètres avant tout lancement d’une procédure de synthèse coordonnée. La figure 5.1 représente la fenêtre de menu du processus de synthèse coordonnée grâce à laquelle le concepteur fixe ces paramètres.

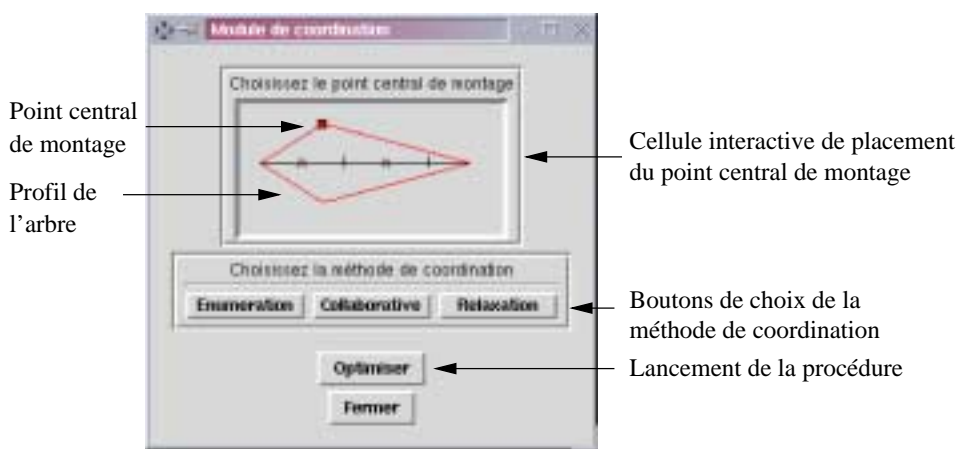


FIG. 5.1 – Fenêtre du menu de synthèse coordonnée.

La partie supérieure de la fenêtre concerne la position du point central de montage. Un rappel du placement des DTLM y est fait en utilisant la représentation schématique utilisée dans les menus précédents. Le point central de montage représente le *point haut de l'arbre* tel que nous l’avons défini en §4.5.1.2. Il fixe l’étagement des diamètres de tronçons d’arbre (figuré par des lignes de contour triangulé symbolisant le profil théorique de l’arbre) et, ce faisant, reflète les règles de montage qui ont été exprimées en amont dans la démarche de conception (cf. chapitre 1). En numérotant les quatre DTLM de la gauche vers la droite, nous constatons que le concepteur a placé le point central de montage entre le DTLM 1 et le DTLM 2. Ceci correspond à un cas de montage mixte où le DTLM 1 verra un montage de ses composants sur l’arbre par le côté gauche et, pour les autres DTLM, un montage de leurs composants sur l’arbre par le côté droit.

La partie médiane de la fenêtre concerne le choix de la méthode de coordination. Le concepteur peut opter pour trois méthodes de recherche d’une solution optimale globale. En réalité, seuls les algorithmes « Collaborative » et « Relaxation », que nous détaillons ci-après, sont des méthodes de synthèse coordonnée. La première option, qui permet d’effectuer la recherche *exhaustive* des combinaisons de DTLM, n’est proposée que dans l’unique objectif de servir d’outil de validation des deux autres méthodes.

Enfin, les derniers boutons lancent la procédure de synthèse et ferment la fenêtre de dialogue.

Nous détaillons dans la section suivante les particularités liées aux systèmes décomposés et à leur résolution.

5.2 Les systèmes décomposés et leur optimisation.

Pour bien comprendre quelles méthodes de coordination peuvent être adaptées à notre tâche de synthèse coordonnée, nous détaillons ci-après les différents types de décomposition communément utilisés. Nous détaillons ensuite quelques méthodes de résolution de systèmes décomposés et commentons leur éventuelle adéquation avec notre problématique.

5.2.1 Les différents types de décomposition.

Les méthodes de décomposition sont un outil précieux dans le traitement des problèmes d'optimisation complexes ou de grandes dimensions. Elles permettent de découper le problème initial en sous-problèmes d'optimisation comportant chacun une fonction objectif et des contraintes associées. Deux grandes familles cohabitent : les décompositions hiérarchiques et les décompositions non-hiérarchiques. Nous analysons ces deux familles dans les sections suivantes, en nous intéressant particulièrement à leurs caractéristiques propres. Nous ne détaillons pas les méthodes de décomposition ([MGM98] et [Mar99]), puisque notre organisation opérationnelle s'appuie, de facto, sur une décomposition par liaisons mécaniques (les DTLM). Nous cherchons simplement à trouver la meilleure méthode de résolution de notre problème.

5.2.1.1 Décompositions hiérarchiques.

La décomposition hiérarchique fait apparaître une structure par niveaux. Elle est née de la complexité grandissante des politiques organisationnelles des industries. Elle est représentative d'une politique organisationnelle qui autorise, à de multiples agents concepteurs d'un même niveau hiérarchique, le travail en totale indépendance les uns vis à vis des autres. Les cahiers des charges et résultats ne sont transmis qu'au supérieur immédiat. Il n'existe aucun couplage entre les sous-problèmes d'un même niveau. Ce genre d'architecture peut présenter autant de niveaux que nécessaire. La limite à ce découpage est fonction de la complexité et de la taille du problème global. Le problème est décomposé en sous-problèmes comprenant des variables globales et des

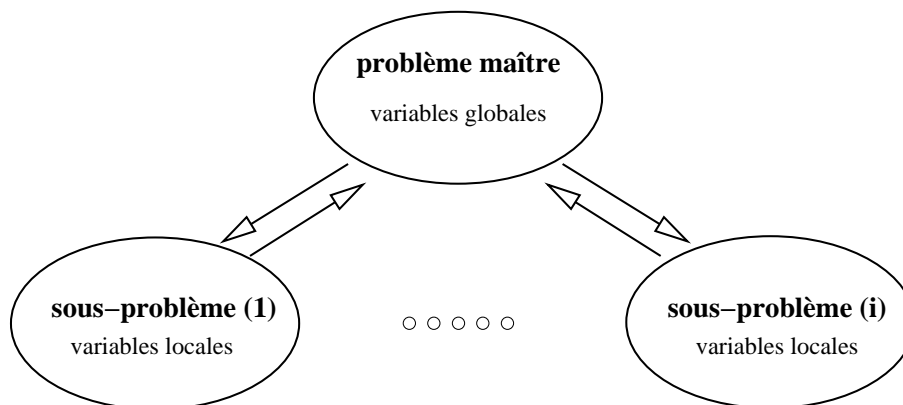


FIG. 5.2 – Décomposition hiérarchique d'un problème d'optimisation.

variables locales. Chaque ensemble de sous-problèmes est supervisé par un problème de niveau supérieur qui doit gérer les liaisons entre eux. Au sommet de la hiérarchie se trouve le problème maître. Cette structure est à rapprocher des structures humaines présentant des autorités de supervision. Toute la difficulté de gestion du problème d'optimisation globale est reportée sur les problèmes de niveaux hiérarchiques supérieurs. Afin de limiter la quantité d'éléments échangés

entre le problème maître et les sous-problèmes, on a souvent recours à des modèles simplifiés et à des indicateurs. L'expertise technique spécifique reste donc confinée dans les sous-problèmes, tandis que les problèmes maîtres les dirigent en fonction des résultats obtenus.

5.2.1.2 Décompositions non-hiérarchiques.

A l'instar de la décomposition hiérarchique, la décomposition non-hiérarchique relève d'un éclatement en sous-problèmes. Ceux-ci peuvent être couplés par des paramètres de liaison qui sont des informations établies au sein d'un sous-problème mais utilisées aussi comme données par d'autres sous-problèmes. Ces paramètres ne figurent pas nécessairement au rang des variables locales du sous-problème (cf. figure 5.3). La différence notable se situe au niveau des interactions

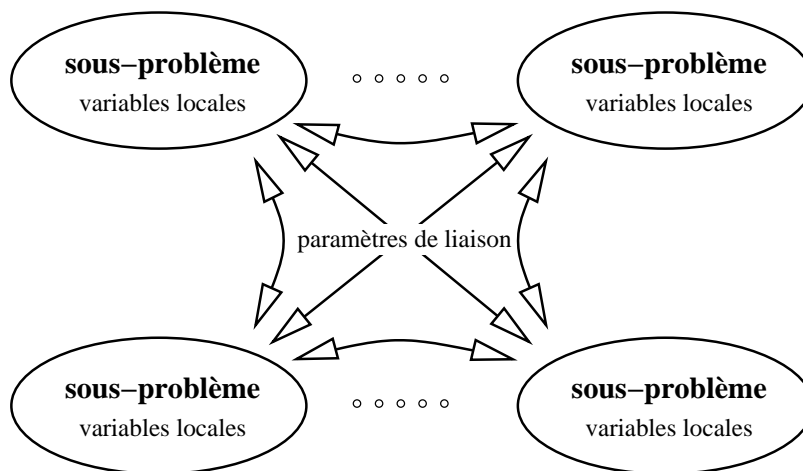


FIG. 5.3 – Décomposition non-hiérarchique d'un problème d'optimisation.

entre sous-problèmes. Il n'existe pas de notion de problème maître et de problèmes esclaves. Tout sous-problème peut posséder des couplages avec tout ou partie des autres sous-problèmes. La gestion de ce genre de problème est délocalisée. Le type de liens ainsi que le choix de l'algorithme de résolution sont très influents sur la qualité du résultat. Tous les sous-problèmes ont le même niveau d'importance.

La difficulté principale inhérente à une décomposition non-hiérarchique réside dans la mise en œuvre d'une méthode de coordination entre les sous-problèmes permettant d'aboutir (ou de s'approcher) au résultat optimal.

Le problème de synthèse de mécanisme par DTLM est du type non-hiérarchique. On peut assimiler le dimensionnement des DTLM, via les modules de calcul, à des sous-problèmes couplés entre eux par des contraintes (contraintes d'interférence et de montage telles que décrites en §2.2.2). Ces contraintes mettent en jeu des paramètres tels que bg_i , bd_i , dag_i ... qui ne sont pas des variables globales. La figure 5.4 illustre certaines de ces contraintes à partir de la représentation des tronçons d'arbre. On y distingue les paramètres de liaison. Parmi les méthodes de résolution existantes pour les problèmes non-hiérarchiques, nous nous sommes intéressés à la MDO¹ qui nous est apparue comme la mieux adaptée à la structure modulaire de notre problème.

1. MultiDisciplinary Optimisation.

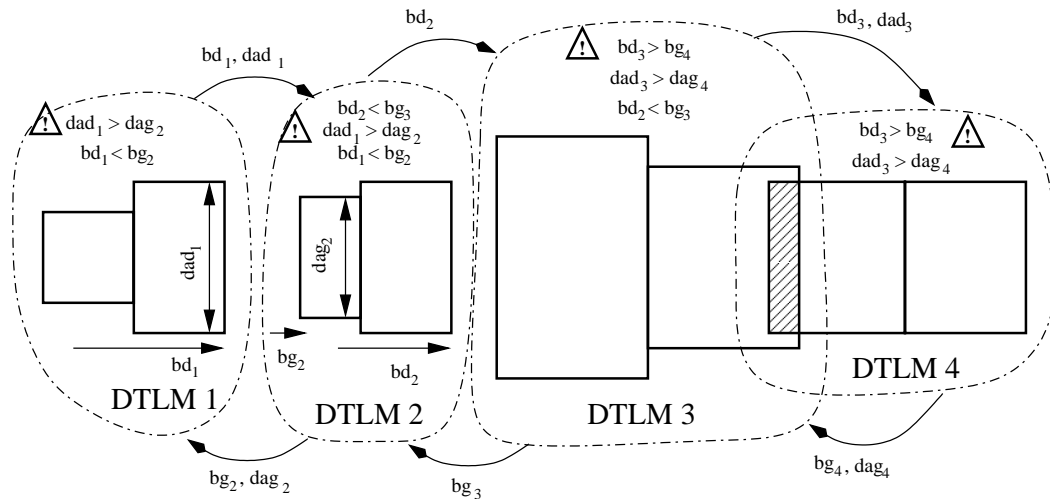


FIG. 5.4 – Exemple de mise en place de paramètres de liaison.

5.2.2 L'optimisation pluridisciplinaire (MDO).

De nombreux travaux ont porté sur les approches ainsi que sur les méthodes adaptées à la gestion des problèmes de conception, notamment du point de vue des mécanismes de coordination ([WCDH00] et [CWDH00]). Les décompositions présentées ci-avant ont en commun le besoin, parfois impérieux, de viser des objectifs différents, voire souvent antagonistes. La résolution de ces problèmes, présentant une hétérogénéité de la nature des problèmes et objectifs locaux, a favorisé l'émergence des approches en optimisation multidisciplinaire ou MDO. Nous présentons les points caractéristiques de ces démarches et nous évaluons l'adéquation potentielle de cette méthode à notre problème.

L'optimisation pluridisciplinaire a vu le jour dans le domaine de la construction aéronautique [Ano91]. Traditionnellement, ce type de construction regroupe des métiers aussi variés que la propulsion, la structure, les commandes ou encore l'aérodynamique. Les recherches les plus récentes ont montré que le couplage entre ces domaines, possédant chacun leurs spécificités théoriques et techniques, pouvait s'avérer très fort. L'optimisation pluridisciplinaire doit être considérée comme un environnement de travail combinant la puissance calculatoire de l'informatique à l'expertise humaine. Ses objectifs sont les suivants :

- condenser les phases majeures de vie du produit que sont la conception, la fabrication et la maintenance pour obtenir un gain de temps, de coûts et de qualité de développement ;
- intégrer les spécificités des différentes disciplines impliquées dans la conception d'un produit (c'est à dire donner une plus grande autonomie aux équipes de conception pour le choix des outils, des méthodes, etc.) ;
- mêler plus intimement les phases de décision et d'action qui bien souvent apparaissent purement séquentielles (travail en parallèle) ;
- retarder au maximum le choix définitif des valeurs des variables de conception (i.e. garder un maximum de liberté dans le processus de conception).

De nombreux aspects ont été pris en compte dans le développement de la démarche, comme les aspects calculatoires (directement liés aux outils et architectures informatiques), les aspects

humains (modélisation des processus et structures de décision) et l'aspect optimisation. Bien entendu, cette discipline s'inscrit dans la démarche d'ingénierie concourante initiée depuis quelques années. Elle participe activement à son développement en reprenant un certain nombre de ses caractéristiques principales [Ano91].

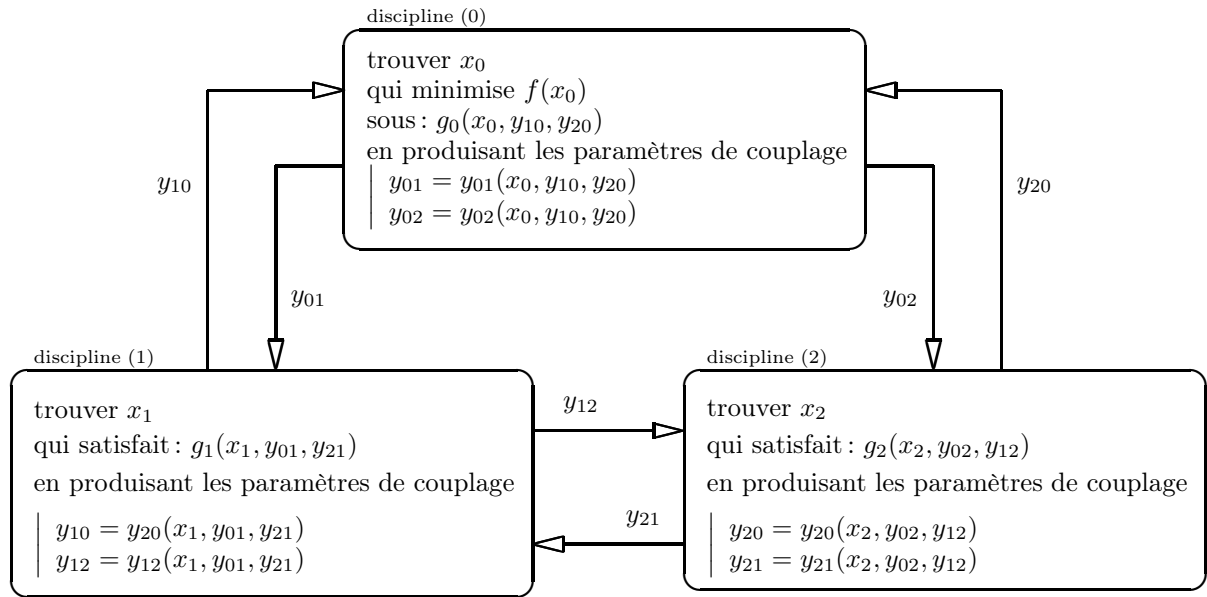
La conception des systèmes complexes, objet de l'optimisation pluridisciplinaire, fait inévitablement appel à des décompositions de tâches. Ceci requiert un processus de coordination intervenant au niveau du système et des sous-systèmes afin de gérer les couplages inévitables. Plusieurs méthodes formelles ont été proposées dans le cadre de la MDO [KS01]. Beaucoup ont en commun une décomposition de la tâche de conception en sous-tâches indépendantes et la présence d'une tâche de coordination, aboutissant alors à un système hiérarchique à deux niveaux. Notons cependant que ce type d'approche peut également s'appliquer à des problèmes comportant des variables discrètes [BG98]. Nous donnons ci-après quelques caractéristiques souvent associées aux structures MDO :

- une interconnexion aisée et rapide entre les outils d'analyse ;
- une structure robuste d'échange de données ;
- un accès à des méthodes d'études paramétriques ;
- une large bibliothèque d'outils d'optimisation et, plus important encore, une expertise quand au choix de ces méthodes ;
- la disponibilité de nombreuses techniques d'approximation (polynomiales, réseaux neuronaux ou autres ...) ;
- la prise en compte de la nature probabiliste et incertaine d'une part des données de conception ;
- une architecture informatique permettant le traitement parallèle ;
- une visualisation des données de conception à tout instant du processus, etc.

La plupart des méthodes utilisées en MDO font appel au calcul de gradients pour évaluer la sensibilité des différents processus d'optimisation. La détermination de ces gradients est coûteuse en temps de calcul [BS96] et leur nombre croît proportionnellement au nombre de variables et de disciplines (ou sous-problèmes). Certains ont tenté de mettre en œuvre de nouveaux algorithmes basés sur la limitation des domaines d'investigation de chaque sous-problèmes. Ceci a été rendu possible en découplant les sous-problèmes et en analysant la faisabilité et la concordance des résultats postcalculatoire. L'algorithme redistribue alors de nouvelles valeurs aux sous-problèmes modifiant ainsi leur point de départ et leur domaine d'investigation [TF02]. Cette méthode est fortement similaire à une démarche séquentielle et n'évite pas les itérations.

5.2.3 Optimisation collaborative.

Parmi les différentes méthodes relevant de l'approche MDO, nous avons retenu l'optimisation collaborative. Celle-ci est basée sur une architecture à deux niveaux, dont le niveau supérieur, nommé *niveau système*, manipule des consignes, dites variables multidisciplinaires, pour minimiser un objectif global tout en satisfaisant des contraintes de couplage [BS95]. Les objectifs des niveaux inférieurs (niveaux locaux) sont la minimisation de l'écart entre les requêtes du niveau système et les variables locales.

FIG. 5.5 – *Problème multidisciplinaire couplé*

Considérons l'un des modèles de base d'une structure non-hiérarchique, représenté sur la figure 5.5. La structure implique trois disciplines comportant leur propre jeu de variables x_i et de contraintes g_i . Ces disciplines reçoivent et, en retour, produisent un certain nombre de données (y_{ij}). Étudions en détail les différents sous-problèmes correspondant à chaque discipline :

- La discipline (0) est la seule à qui on a confié une tâche d'optimisation. C'est elle qui doit orienter la recherche d'une solution vers un optimum global. Elle a toutefois sa logique propre en ce sens qu'elle doit satisfaire un système de contraintes g_0 . Dans le cadre de son activité, elle produit des paramètres (y_{01} et y_{02}) qu'elle transmet aux disciplines (1) et (2).
- Les disciplines (1) et (2) ont des activités qui se limitent à la recherche d'une solution réalisable pour chaque sous-problème associé. À partir des données reçues des deux autres disciplines (dans le cas de la discipline (1) il s'agit de y_{01} et y_{21}) elles doivent trouver une solution x_i satisfaisant les contraintes g_i du sous-problème i . Si la discipline i est une activité de conception très générale, la recherche de la solution x_i peut faire appel à une méthode de nature quelconque comme, par exemple, un processus du type « essai-erreur » : on impose une géométrie de départ et on vérifie que les divers critères (résistance mécanique, encombrement...) sont satisfaits ($g_i \leq 0$). En cas de violation de contraintes, on fait évoluer la géométrie jusqu'à atteindre une solution réalisable. Dans le cadre de leur activité, ces disciplines sont amenées à produire des paramètres (y_{10} et y_{12} pour la discipline (1)) destinés aux autres disciplines. Bien sûr, les couplages qui existent entre les trois sous-problèmes constituent la difficulté majeure à surmonter lorsqu'on cherche à mettre en place une méthode de résolution globale. On se heurte à un phénomène de bouclage² puisque les y_{ij} ne peuvent pas être calculés tant que les y_{ji} ne sont pas connus et réciproquement.

L'optimisation collaborative consiste à transformer ce problème en le remplaçant par le problème illustré sur la figure 5.7. Dans cette structure, la discipline (0) (le niveau système) endosse le rôle

2. Également connu sous le nom de « problème de l'œuf et de la poule ».

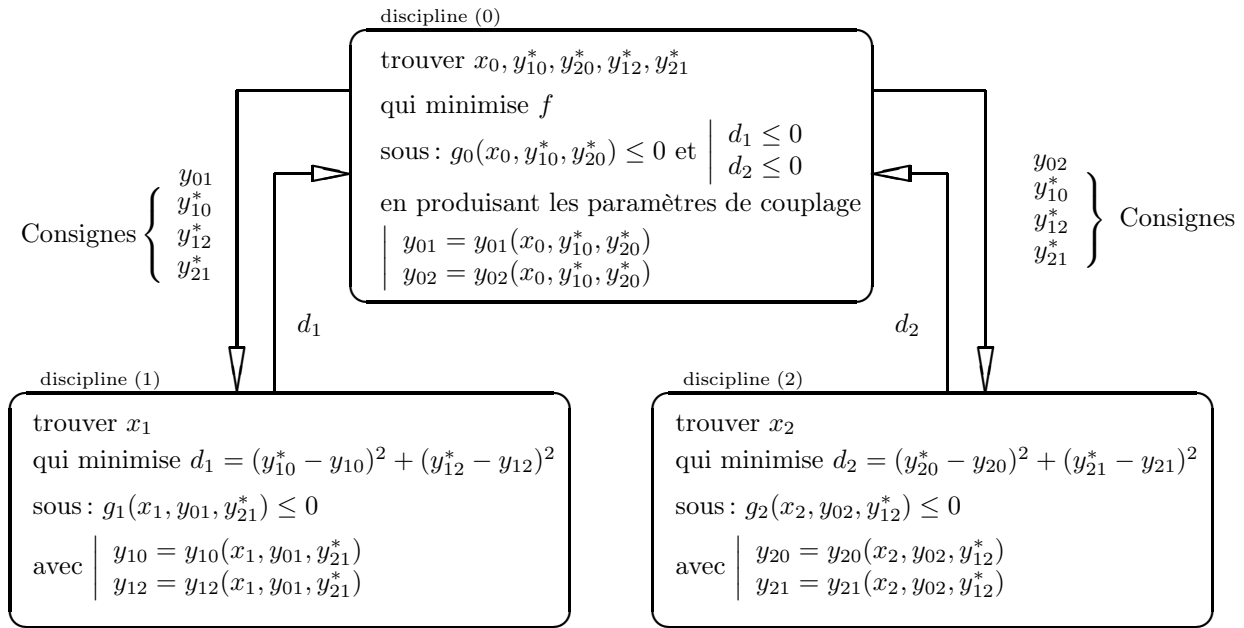


FIG. 5.6 – Formulation en optimisation collaborative

de superviseur. Elle a pour fonction de guider le travail des disciplines locales vers une solution globale. Pour cela, le niveau système doit déterminer les variables systèmes x_0 , soumises aux contraintes g_0 , et doit minimiser l'objectif système (f). Pour cela, il dispose de consignes y_{ij}^* qu'il transmet aux disciplines locales afin d'orienter leur recherche. Le rôle des disciplines locales est alors de trouver les variables locales x_i satisfaisant les contraintes locales g_i et minimisant les dépassement de consignes d_i . Ces variables matérialisent l'écart entre les consignes envoyées par le niveau système (y_{i0}^* et y_{ij}^*) aux disciplines locales, et les paramètres que ces dernières produisent (y_{i0} et y_{ij}).

La force de l'optimisation collaborative réside dans le fait que chaque discipline possède l'entière responsabilité de produire ses propres variables satisfaisant à ses propres contraintes. Ceci permet une grande souplesse sur le choix des variables, des contraintes et de la méthode de recherche de solution.

La résolution d'un tel problème fait traditionnellement appel, pour la partie système, à des algorithmes NLP³ basés sur les gradients des fonctions de dépassement qui peuvent être indifféremment récupérés des disciplines locales ou calculés à partir des variations de ces fonctions [BS95]. La méthode a été étendue à des problèmes pouvant présenter, au niveau local, des variables discrètes [BG98].

Notre problème présente quelques différences avec le cas type que nous avons présenté. Nous expliquons ci-après la démarche que nous avons mis en place pour appliquer la méthode de l'optimisation collaborative au problème de synthèse des DTLM.

5.3 Application de l'optimisation collaborative à notre problème.

5.3.1 Formulation initiale du problème sous forme non-hiérarchique.

Considérons dans un premier temps les modules de calcul de DTLM comme étant des disciplines à part entière. Nous pouvons alors mettre en évidence les variables et les contraintes locales, éléments communs à ce genre de problème. Nous devons également faire apparaître les contraintes de couplage. Dans cette formulation, chacun des modules, considéré comme un pro-

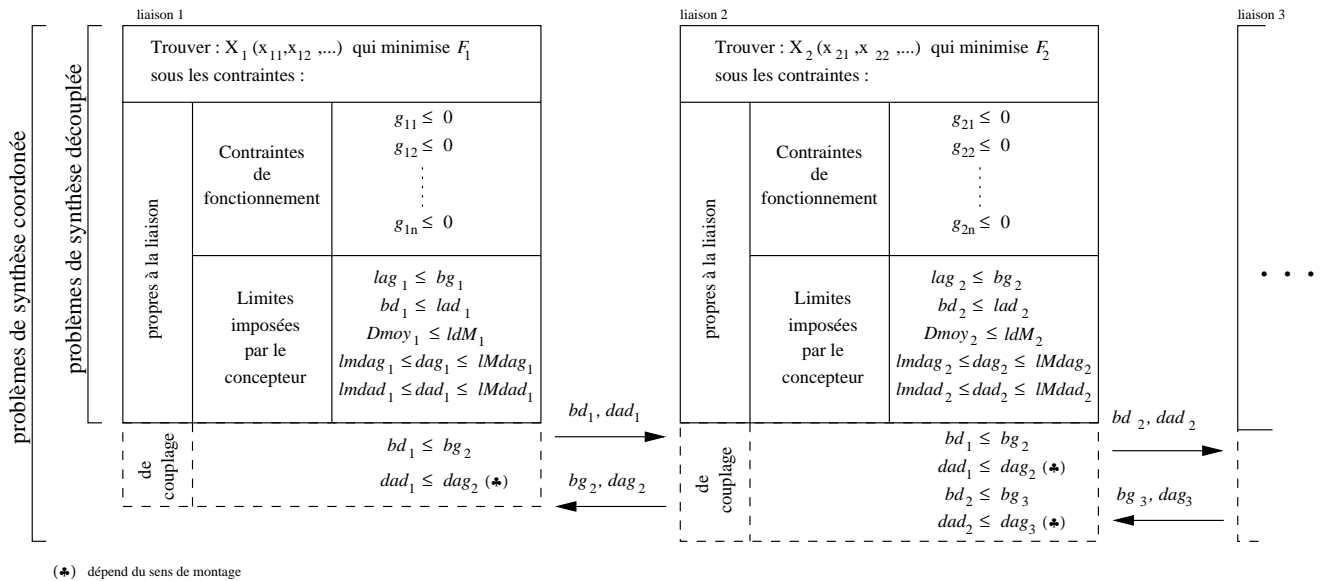


FIG. 5.7 – Découpage du problème d'optimisation collaborative.

blème d'optimisation sous contraintes, cherche à minimiser l'objectif local noté F_i qui lui a été assigné par le concepteur en respectant les contraintes :

- locales : on retrouve parmi celles-ci les contraintes présentées au chapitre 2, à savoir les contraintes de fonctionnement ainsi que les contraintes d'encombrement axial et radial imposées par le concepteur à chaque DTLM ;
- de couplage : elles assurent la cohérence de la solution d'ensemble en imposant des règles de non-interférence et d'ordre de montage.

La figure 5.7 représente une telle formulation. On distingue aisément les interactions entre les modules représentées par les contraintes de couplage (l'équivalent des y_{ij} dans la formulation type de la figure 5.5). Ces dernières mettent en jeu les variables d'encombrement axial (bg_i et bd_i) et de diamètre d'arbre (dag_i et dad_i). Cette formulation n'est cependant pas complètement déterministe dans la mesure où un objectif global n'est pas clairement défini. Le « poids » relatif⁴ des différents modules n'est pas exprimé, ce qui laisse le problème dans une forme indéterminée.

4. Nous entendons par « poids » relatif l'importance relative qu'attache le concepteur au respect des objectifs d'optimisation locaux.

5.3.2 Problème global exprimé sous forme collaborative.

Une nouvelle architecture du problème global est mise en place selon les principes du travail collaboratif énoncé précédemment. Les modules sont tous considérés comme des disciplines⁵. Notre problème présente cependant deux différences notables par rapport au cas type défini en §5.2.2 :

1. il n'y a pas, au départ, de discipline chargée de coordonner les autres et à laquelle on aurait confié la gestion d'un objectif global (l'équivalent de la discipline 0) ;
2. chaque sous-problème local est posé comme un problème d'optimisation avec un objectif local propre à chaque liaison.

Nous avons donc été amené à reformuler notre problème afin de rentrer dans le cadre de l'optimisation collaborative. En premier lieu, nous avons dû mettre en place une discipline *système* (cf. figure 5.8). Le rôle de cette dernière est de coordonner les opérations de dimensionnement optimal des modules. Le problème du niveau système prend pour cela la forme d'un problème d'optimisation de niveau global dont le but est de faire converger l'ensemble du système vers un optimum global respectant les contraintes de couplage. Afin d'assurer au collaborateur système un contrôle total sur les modules on met en place la *structure hiérarchique* à deux niveaux présentée en §5.2.3. Les modules locaux conservent ainsi leurs variables locales et une certaine autonomie de travail dans leur recherche. Les variables consignées sont de deux types :

- les bd_i^* , bg_i^* , dag_i^* et dad_i^* sont directement associées aux paramètres qui intervenaient initialement dans le couplage des modules locaux de la formulation précédente.
- les F_i^* sont directement associées aux objectifs locaux F_i . Il s'agit d'une valeur cible de l'objectif local. Ces valeurs sont toujours supérieures ou égales (dans le cas d'une minimisation des objectifs) aux valeurs atteintes par les objectifs locaux en synthèse découplée ($F_i^* \geq F_{iSD}$), c'est-à-dire sans avoir introduit de contraintes de couplage. Les objectifs locaux F_{iSD} (pour F_i en Synthèse Découplée) sont les objectifs locaux obtenus par les meilleurs DTLM de chacune des liaisons en synthèse découplée. Ces valeurs sont, par définition, idéales. En effet, si le concepteur a réussi à mener à bien la phase de synthèse découplée en tenant compte des contraintes de couplage, alors il a obtenu la solution globale optimale ; si il n'a pas intégré ces contraintes, alors les DTLM résultants devront être dégradés afin d'obtenir une solution optimale globale. Les valeurs des objectifs locaux sont alors des cibles qui, dans le meilleur des cas, pourront être atteintes, mais jamais dépassées. Si nous considérons l'exemple présenté en §4.5.3, nous voyons que les valeurs obtenues lors de la première passe de calcul, c'est-à-dire avant toute tentative d'amélioration par le concepteur, donne des valeurs d'objectifs (tableau 4.1) qui sont inférieures ou égales à celles de la solution finale (tableau 4.3).

La formulation de l'objectif du niveau système (objectif global) f est le suivant :

$$f = \left(\frac{F_1^* - F_{1SD}}{F_{1SD}} \right)^2 + \left(\frac{F_2^* - F_{2SD}}{F_{2SD}} \right)^2 + \left(\frac{F_3^* - F_{3SD}}{F_{3SD}} \right)^2 + \left(\frac{F_4^* - F_{4SD}}{F_{4SD}} \right)^2 = \sum_{i=1}^4 f_i \quad (5.1)$$

5. On parlera aussi de *collaborateur*.

Détaillons la formulation de l'objectif global (niveau système). Il est constitué de la somme des objectifs locaux normés (c'est une approche classique en optimisation multicritère⁶). Son rôle est de départager les solutions d'ensemble concurrentes. Il n'a pas été mis en place de pondération entre les différents DTLM ce qui signifie qu'ils ont tous la même importance. L'objectif tendra vers zéro, sa valeur minimale, lorsque tous les modules tendront vers leur solution optimale obtenue en synthèse découplée (F_{iSD}). Le cas idéal ($f = 0$) obtenu lorsque $\forall i, F_i^* = F_{iSD}$, ne peut avoir lieu que lorsque la synthèse découplée a fourni une solution non conflictuelle (ce que le concepteur peut vérifier avant de lancer le processus et qui, dès lors, ne justifie plus la synthèse coordonnée!). Le concepteur a bien entendu la possibilité de vérifier la validité de sa solution avant de lancer le processus de synthèse

Les f_i correspondent à des pourcentages de dégradation (sous forme de perte de performance) des objectifs locaux F_i relativement à leurs valeurs cibles F_{iSD} . La valeur f indique donc la somme des carrés normalisés des pertes par rapport à ces valeurs cibles, et le module système tend à minimiser cette valeur.

Les variables d_i représentent les indices de dépassement⁷ des consignes de chaque module local. Elles constituent aussi les objectifs locaux. Elles ont pour expression :

$$d_1 = \left(\frac{F_1 - F_1^*}{F_1^*} \right)^2 + \left(\frac{bd_1 - bd_1^*}{bd_1^*} \right)^2 + \left(\frac{dad_1 - dad_1^*}{dad_1^*} \right)^2 \quad (5.2)$$

$$d_i = \left(\frac{F_i - F_i^*}{F_i^*} \right)^2 + \left(\frac{bd_i - bd_i^*}{bd_i^*} \right)^2 + \left(\frac{dad_i - dad_i^*}{dad_i^*} \right)^2 + \left(\frac{bg_i - bg_i^*}{bg_i^*} \right)^2 + \left(\frac{dag_i - dag_i^*}{dag_i^*} \right)^2 \quad \text{pour } i = 2, 3 \quad (5.3)$$

$$d_4 = \left(\frac{F_4 - F_4^*}{F_4^*} \right)^2 + \left(\frac{bg_4 - bg_4^*}{bg_4^*} \right)^2 + \left(\frac{dag_4 - dag_4^*}{dag_4^*} \right)^2 \quad (5.4)$$

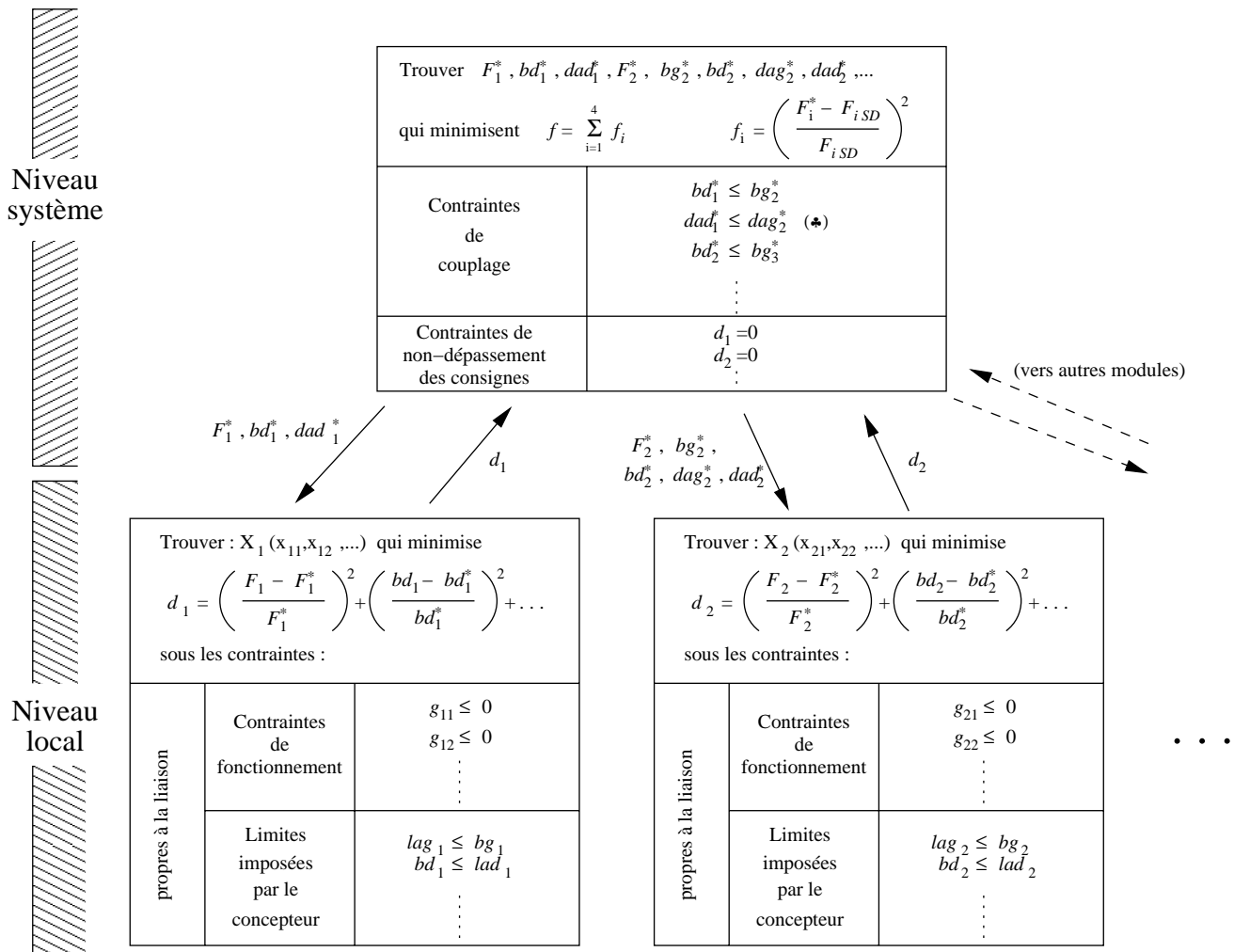
Intéressons-nous maintenant aux objectifs locaux. Nous constatons que ceux-ci se trouvent modifiés. Ils prennent également la forme d'un indicateur de dégradation par rapport aux consignes fournies par le niveau système. Une fois encore, on effectue une sommation des carrés des écarts normés. La conséquence de cette formulation de l'objectif est la suivante : le travail du niveau local est dorénavant de trouver une solution locale qui se rapproche le plus possible des consignes émanant du niveau système. Remarquons que le nouvel objectif intègre l'objectif initial (synthèse découplée) puisqu'il tend à diminuer l'écart entre le résultat local et la consigne ($F_i - F_i^*$), elle-même tendant à rapprocher F_i^* de la valeur de l'objectif en synthèse découplée F_{iSD} . Le processus global tend donc à chercher une solution rapprochant les F_i des F_{iSD} .

Les contraintes de couplage sont toutes regroupées dans le problème du niveau système. Elles sont intégralement formulées à partir des variables de consignes. C'est ainsi que le niveau système peut gérer la cohérence d'ensemble en élaborant un faisceau de consignes satisfaisant ces contraintes.

Les contraintes de non-dépassement des consignes jouent un rôle prépondérant dans la gestion de la cohérence du système puisqu'elles doivent empêcher le système de dériver vers des consignes inadaptées. Une valeur nulle de d_i indique que le module local correspondant a trouvé une solution conforme aux consignes (rappelons que le module local doit fournir une solution locale satisfaisant les contraintes locales et, dans le cas d'un ensemble de solutions réalisables, minimisant l'objectif local d_i). Si les consignes fournies par le niveau système ($F_i^*, bd_i^* \dots$) ne peuvent être suivies, le niveau local peut néanmoins fournir une solution respectant les contraintes locales, mais

6. Critère d'optimisation unique formé d'une multitude de critères pondérés, de natures différentes ou non.

7. Ou *discrepancy* en anglais.



(♣) dépend du sens de montage

FIG. 5.8 – Découpage hiérarchique du problème d'optimisation collaborative.

renvoyant une valeur non-nulle de dépassement de consignes. Cette dernière indique alors au niveau système qu'il n'a pas pu être trouvé de solution satisfaisant les consignes. L'amplitude de la valeur du dépassement de consignes informe le niveau système sur la proximité du domaine des solutions réalisables.

5.3.3 Choix d'une méthode pour l'implémentation du niveau système : l'algorithme de Powell.

Les méthodes les plus répandues pour la résolution numérique des problèmes d'optimisation sous contraintes sont les méthodes de descente basées sur l'utilisation du gradient de la fonction objectif et des fonctions contraintes [VAN84]. Le modèle d'optimisation collaborative présenté en 5.2.3 faisait appel aux gradients des fonctions de dépassement. Comme nous le verrons plus loin, la difficulté majeure que nous avons rencontrée dans notre problème est liée à la nature des fonctions d_i . En effet, compte-tenu de la nature discrète des variables des DTLM que nous manipulons, une variation continue de consigne peut s'accompagner d'un brusque changement de direction des fonctions d_i . En d'autres termes, les fonctions d_i ne sont pas continuellement différentiables. De plus, le principe d'encapsulation qui nous a guidé dans l'élaboration de notre structure de données nous donne une grande liberté concernant le choix des méthodes d'optimisation des modules locaux. Il est ainsi possible de mettre en œuvre des méthodes d'optimisation (stochastiques par exemple) qui ne peuvent pas fournir d'informations comme des gradients.

De ce fait, l'utilisation d'une méthode d'ordre 0 nous est apparue préférable. Parmi celles-ci, l'algorithme de Powell a retenu notre attention. Cette méthode d'optimisation sans dérivées est classée parmi les méthodes de relaxation cyclique [Min89]. Elle procède par minimisations unidimensionnelles successives le long de *directions conjuguées* qui sont progressivement déterminées à partir des directions associées aux variables du problème. Cette méthode a été conçue [Pow64] pour minimiser des fonctions à plusieurs variables sans avoir accès à leur gradient (calcul de dérivées impossible ou trop coûteux). Le principe de l'algorithme est succinctement présenté ci-dessous [PTVF92] :

1. Initialisation du jeu de directions avec le vecteur de base : $u_i = e_i$ avec $i = 1, \dots, N$
2. Sauvegarder le point de départ en tant que P_0 .
3. Pour $i = 1, \dots, N$ chercher le point P_i correspondant au minimum de la fonction selon la direction u_i en partant de P_{i-1} .
4. Pour $i = 1, \dots, N - 1$ décaler les directions $u_i \leftarrow u_{i+1}$
5. Affecter $u_N \leftarrow P_N - P_0$.
6. Chercher le point P_N correspondant au minimum de la fonction selon la direction u_N et renommer ce point P_0 .
7. Reprendre au point 2 jusqu'à convergence du processus.

Plusieurs travaux ont eu pour objectif d'améliorer la méthode, notamment pour s'affranchir des problèmes inhérents à la dépendance linéaire des directions conjuguées.

La formulation de notre problème doit encore subir une transformation. En effet, l'algorithme de Powell est exclusivement dédié à la résolution de problèmes d'optimisation sans contraintes. Nous devons donc intégrer les contraintes du niveau système à la fonction objectif.

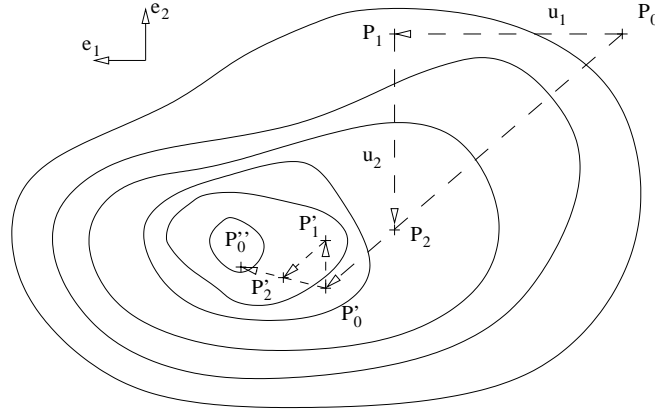


FIG. 5.9 – Exemple d’application de l’algorithme de Powell sur un problème bidimensionnel.

5.3.4 Expression du problème du niveau système en problème sans contrainte.

Nous avons opté pour la méthode des pénalités extérieures. Celle-ci se distingue par sa simplicité de principe, son efficacité pratique et sa facilité d’exploitation. Le problème du niveau système est ainsi reformulé de la façon suivante :

$$\text{Trouver : } F_1^*, bd_1^*, dad_1^*, F_2^*, \dots \quad \text{qui minimisent} \quad f_p = \sum_{i=1}^4 f_i + rH \quad (5.5)$$

$$\text{où } H = \left[\max \left(0, \frac{bd_1^* - bg_2^*}{bg_2^*} \right) \right]^2 + \left[\max \left(0, \frac{\mp dad_1^* \pm dag_2^*}{dag_2^*} \right) \right]^2 + \dots + \sum_{i=1}^4 d_i \quad (5.6)$$

et $\begin{cases} f_p \text{ est le nouvel objectif pénalisé} \\ r \text{ est le coefficient de pénalité} \\ H \text{ est le terme de pénalisation par violation de contraintes} \end{cases}$

Le terme de pénalisation par violation de contraintes (H) est composé de la somme normée des violations des contraintes de couplage et de dépassement des consignes (d_i).

5.3.5 Mise en œuvre de l’algorithme de Powell.

Nous avons décidé d’utiliser un code informatique éprouvé afin d’appliquer l’algorithme de Powell à la nouvelle formulation de notre problème. Notre choix s’est donc porté sur la librairie mathématique Optsolve++ [Cor02].

L’exemple que nous présentons ci-après reprend le cahier des charges de l’exemple présenté en §4.5.3. Nous avons appliqué la procédure de synthèse coordonnée par la méthode de Powell sur ce problème afin d’observer son comportement. Le coefficient de pénalité a été pris égal à 100 et la précision de l’algorithme fixée au millionième (10^{-6}). Les principales dimensions de la solution produite par l’optimisation collaborative sont données dans le tableau 5.1. Nous y avons également fait figurer la solution obtenue par la méthode énumérative (cf. §4.5.3) à titre de comparaison. Les valeurs des objectifs locaux figurent en bas de tableau tandis que les données conflictuelles se distinguent en gras.

Comparativement à la solution de départ, c’est à dire celle obtenue par le lancement individuel des modules de calculs sur le cahier des charges initial (cf. §4.5.3), nous constatons une

	DTLM 1	DTLM 2	DTLM 3	DTLM 4	DTLM 1	DTLM 2	DTLM 3	DTLM 4
bg_i	71	102.3	173.5	243	71	105.85	173.5	242
bd_i	106.5	175.7	218.5	310.9	105	172.15	218.5	311.95
dag_i	21	26	45	35.6	25	30	45	34.7
dad_i	25	32.7	35	29	30	36.7	35	28
F_i	35.5	26	35	67.9	34	30	35	69.9
	optimisation collaborative (Powell)				méthode énumérative			

TAB. 5.1 – *Solution obtenue par optimisation collaborative.*

amélioration des conflits (voir tableau 4.1). Au prix d'une dégradation des objectifs locaux sur les DTLM 3 et 4, l'algorithme tend vers une «quasi-solution» puisque l'interférence de montage est réduite à une très faible valeur. L'interférence axiale entre les DTLM 1 et 2 voit également sa valeur diminuer. Les résultats sont obtenus pour un temps de calcul de l'ordre de la seconde, ce qui dénote un bon comportement de l'algorithme de Powell dans sa progression vers une solution. La fonction objectif globale a été évaluée environ 1400 fois (études de sensibilité comprises) pour 10 itérations de l'algorithme de Powell. Ces résultats sont directement liés à la précision imposée à l'algorithme de Powell, et une baisse significative de celle-ci entraîne la diminution du nombre d'itérations, et donc d'évaluations de la fonction objectif. À titre d'exemple, l'abaissement de la précision à 10^{-1} conduit, pour des résultats identiques, à un gain de 30% sur les nombres d'itérations et d'évaluations. Notons que nos essais ont été menés avec différentes valeurs de précision [Cor02] de l'algorithme. Le paramètre de précision influe sur la convergence du système et sur le critère d'arrêt. En ce sens on dénote une légère amélioration de l'objectif final. Cela est uniquement dû au fait que les consignes sont mieux «affinées» lorsque la précision est élevée (cf. figure 5.12), et donc les violations plus faibles. Ceci ne change cependant en rien le résultat final et tend à allonger le processus par un supplément d'itérations (précisons enfin que les DTLM optimaux sont obtenus dès la sixième itération). Il y a donc un intérêt certain à adapter les paramètres de l'algorithme de Powell afin d'accélérer le processus de recherche d'une solution.

Nous ne proposons pas de comparer les valeurs des objectifs globaux obtenus par la méthode énumérative aux résultats de l'algorithme de Powell, car cela n'aurait aucun sens. En effet, la fonction objectif retenu pour la méthode énumérative (cf. formule 4.1) ne tient pas compte des pénalités nécessaires dans la formulation de Powell. Cette dernière est également proportionnelle au coefficient de pénalité. On peut cependant constater que la solution obtenue par «Powell» présentent des objectifs locaux assez semblables à la solution optimale obtenue par le biais d'une énumération de toutes les combinaisons. Cependant, et malgré la quasi-disparition du conflit initial de montage entre les DTLM 3 et 4, l'interférence axiale entre les DTLM 1 et 2 est restée significative. Nous analysons ci-dessous le processus afin de comprendre ces résultats.

Afin d'illustrer la progression de l'algorithme de Powell, nous avons choisi d'étudier les quatre variables impliquées directement dans les conflits initiaux, à savoir bd_1^* et bg_2^* pour l'interférence axiale et dad_3^* et dag_4^* pour le conflit de montage. La lecture des diagrammes de la figure 5.10 s'effectue de la façon suivante. En abscisse, est représenté le nombre d'évaluations de la fonction objectif pénalisée. En ordonnée, sont affichées les valeurs des variables représentées. Les valeurs initiales des consignes sont prises égales aux valeurs des variables issues de la dernière solution obtenue par la synthèse découplée. Nous sommes donc dans le cas présenté au chapitre précédent, où le concepteur cherche à optimiser une solution dont les DTLM, pris indépendamment, sont optimaux mais présentent des conflits une fois regroupés. La figure 5.10 présente deux diagrammes symbolisant l'évolution de certaines consignes du niveau système adressées au

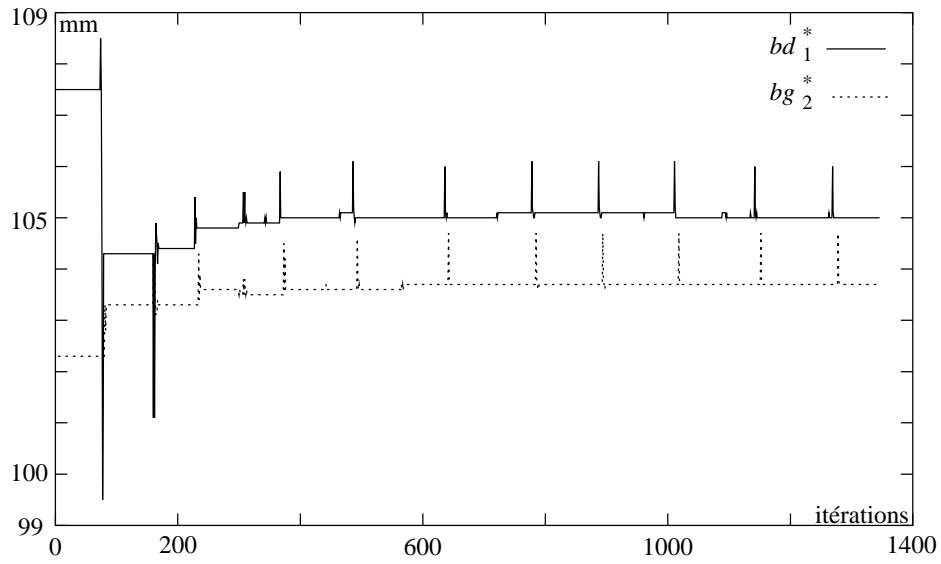
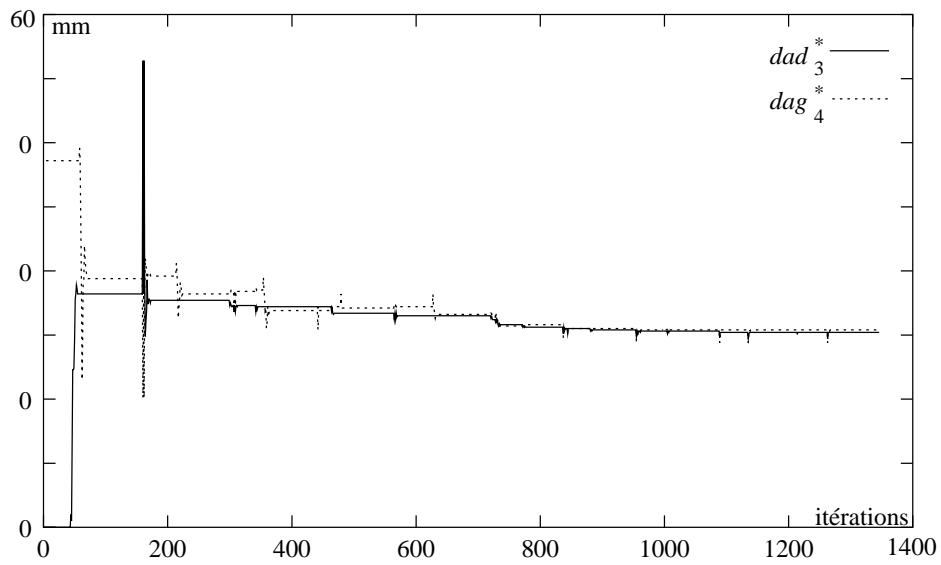
(a) évolution des consignes bd_1^* et bg_2^* (b) évolution des consignes dad_3^* et dag_4^*

FIG. 5.10 – Diagramme d'évolution des consignes du niveau système.

niveau local. Ces courbes se présentent sous forme d'escaliers avec, régulièrement, des variations brusques. Il apparaît, conformément à la présentation de l'algorithme de Powell que nous avons faite ci-dessus, que les variables sont séquentiellement soumises à variation lors de la progression de la solution dans une des directions conjuguées. La présence de paliers dénote une phase de stabilité de la variable (pendant laquelle d'autres directions sont testées). En ce qui concerne les oscillations en début de palier (cf. figure 5.10(a)), elles sont dues à l'algorithme de Brent (il s'agit d'un processus de minimisation unidimensionnelle sans dérivée, basé sur la méthode «golden section search» couplée à une interpolation parabolique) auquel fait appel l'algorithme de Powell [PTVF92]. C'est le moment précis où l'algorithme recherche, dans une direction donnée, le minimum de la fonction objectif. Lorsque l'algorithme compose les directions conjuguées, les changements de valeur de la variable tendent naturellement à devenir d'autant plus fréquent que l'implication de la variable dans la nouvelle direction est grande.

La figure 5.10(a) représente l'évolution des consignes bd_1^* et bg_2^* (rappelons que ces consignes sont issues du niveau système et à destination des modules locaux). Malgré un rapprochement significatif des valeurs des deux consignes, le processus ne parvient pas à résoudre le conflit. En revanche, la figure 5.10(b) illustre les deux tâches principales que doit assumer le niveau système :

1. La résolution de conflit. On peut voir que les deux variables dad_3^* et dag_4^* se rapprochent très rapidement l'une de l'autre, et ce dès la première itération de l'algorithme.
2. L'optimisation de la solution globale. Les courbes diminuent progressivement, ce qui laisse penser que l'optimisation de la solution globale passe par une réduction des diamètres d'arbre.

5.3.6 Influence du comportement des modules de calcul.

La nature discrète de notre problème peut perturber le processus de l'algorithme de Powell dans sa recherche de solutions. La définition des DTLM fait appel à de nombreux types de paramètres : continus, entiers ou discrets. Cette diversité dans la gamme de valeurs que peut prendre chaque paramètre, ainsi que l'étagement de ces valeurs, a une conséquence immédiate sur les résultats des modules locaux. A titre d'exemple, nous nous sommes intéressés à un cas de calcul de liaison arbre/moyeu par clavette parallèle. Les résultats que nous produisons sur la figure 5.11 ont été obtenus pour une optimisation en largeur. Les solutions représentées sont donc la largeur minimale du DTLM pour un diamètre d'arbre donné. Deux domaines⁸ sont définis, correspondant à un changement de contrainte active. Sur ces deux domaines, nous distinguons une non-continuité de fonction, matérialisée par des paliers de valeurs correspondant aux plages de diamètres d'arbre définissant les dimensions caractéristiques des assemblages clavetés (cf. §2.3.1.1).

Nous ne justifions pas la forme de ces courbes, ceci étant hors de notre propos. Considérons que le module du niveau système soit amené à faire varier la contrainte sur le diamètre d'arbre du DTLM. En faisant varier, par exemple, le diamètre minimum il va forcer le module de calcul à parcourir l'ensemble des solutions représentées sur la figure. On conçoit dès lors aisément que la solution fournie par le module de calcul risque de changer brusquement de dimensions en largeur alors que la modification de la contrainte sur le diamètre minimum sera faible (par exemple

8. Le domaine noté «I» est relatif à la partie décroissante de la courbe de réponse du module de calcul, tandis que le domaine noté «II» est relatif à la partie croissante.

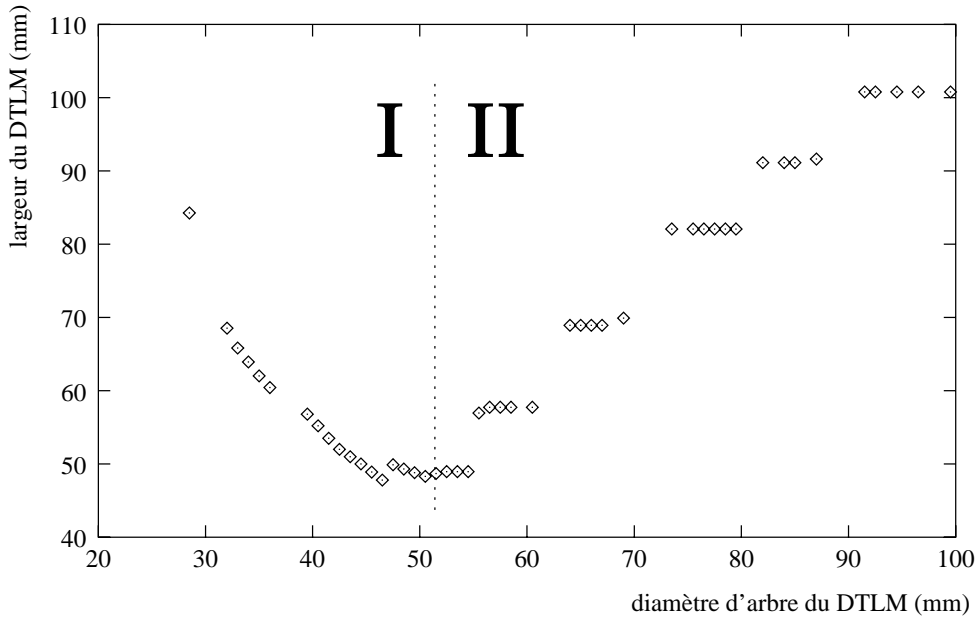


FIG. 5.11 – Dimensions de solutions concurrentes d'un DTLM pour une minimisation de la largeur.

pour un diamètre de 70 mm). Ceci a bien entendu des conséquences sur le processus de synthèse coordonnée que nous expliquons en suivant.

Considérons la figure 5.12. Elle représente la courbe de la consigne dad_3^* déjà présentée sur la figure 5.10(b). Nous y avons adjoint la courbe de réponse du module local pour la variable dad_3 . Le DTLM 3 se caractérise par un grand étagement⁹ de ses dimensions. Malgré les tentatives de l'algorithme pour faire évoluer ce DTLM (et ce quelle que soit la direction prise), nous constatons que ce dernier se fige assez tôt sur une solution locale pour ne plus en changer. Le niveau système n'a plus alors que la solution de rapprocher la consigne de la valeur constamment renvoyée par le module local afin d'en diminuer le terme de pénalisation, et donc la fonction objectif (cf. équations 5.5 et 5.6) :

$$\lim_{dad_3^* \rightarrow dad_3} \left(\frac{dad_3 - dad_3^*}{dad_3^*} \right)^2 = 0$$

L'algorithme est ainsi attiré vers un « puits » car il ne perçoit d'amélioration qu'en calquant la consigne sur la valeur (voisine) renvoyée par le module local.

Lorsque le DTLM présente un étagement plus réduit de ses dimensions, le module local répond beaucoup mieux aux consignes du niveau système. On se rapproche alors d'un comportement typique de fonction continue. La figure 5.13 représente les courbes relatives à la variable dag_4 , également présentée ci-dessus, et sa consigne. Les courbes sont plus intimement liées, traduisant une meilleure réponse du niveau local, ce qui permet au système d'essayer d'améliorer l'objectif global sans être excessivement pénalisé par les dépassements de consignes.

L'algorithme de Powell est une méthode de résolution ne pouvant s'appliquer qu'à des problèmes sans contraintes. La méthode de pénalité extérieure à laquelle nous avons eu recours, afin d'intégrer les contraintes à la fonction objectif, peut présenter certains inconvénients. La mise en place de la fonction objectif pénalisée associe des objectifs et des contraintes. L'algorithme se basant uniquement sur la valeur de la fonction objectif, il arrive souvent qu'une violation

9. Rappel: il s'agit d'un DTLM à base de roulement à billes.

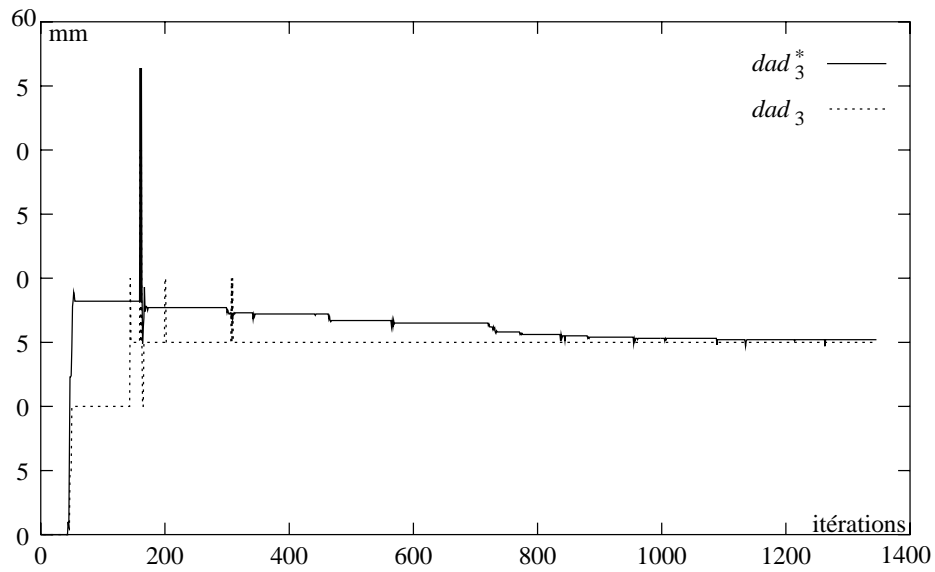


FIG. 5.12 – Réponses d'un module local aux consignes du niveau système.

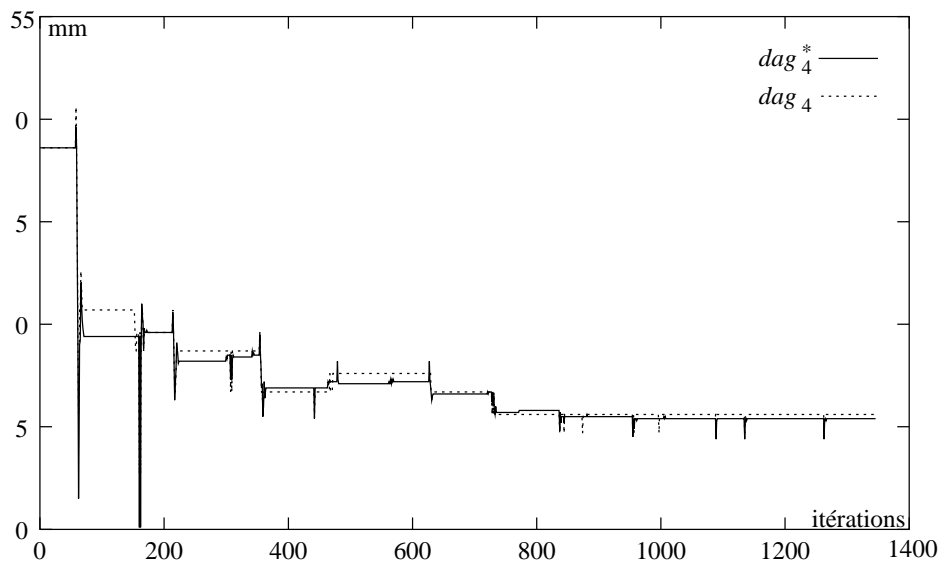


FIG. 5.13 – Influence du type de DTLM sur la réponse d'un module local.

de contrainte soit « masquée » par l'amélioration d'un objectif antagoniste. Il est alors utile de jouer sur la valeur du coefficient de pénalité afin de favoriser la résolution de tels ou tels conflits, à condition que les termes en conflits ne soient pas amplifiés par le même coefficient de pénalité (notre formulation, par exemple, regroupe les termes de dépassement et de violation de contraintes sous l'influence du même coefficient de pénalité). Une solution consiste en une mise en place d'une multitude de coefficients de pénalité. Ceci permet de dissocier le poids relatif de chacun des termes de la fonction objectif. Une telle formulation est cependant délicate à mettre en place sans avoir une connaissance préalable des conflits qui seront difficiles à résoudre ! On peut également envisager une variation pilotée du (ou des) coefficient de pénalité [Min89], mais sa mise au point reste délicate. Toutes ces méthodes ne garantissent pas à coup sûr l'émergence d'une solution. Le principe même de la méthode de pénalité extérieure est de proposer une fonction continue et continuellement dérivable. Or notre problème, de nature discrète, ne profite pas de cet avantage. Le bon comportement généralement admis pour cette méthode ne se retrouve pas dans notre étude.

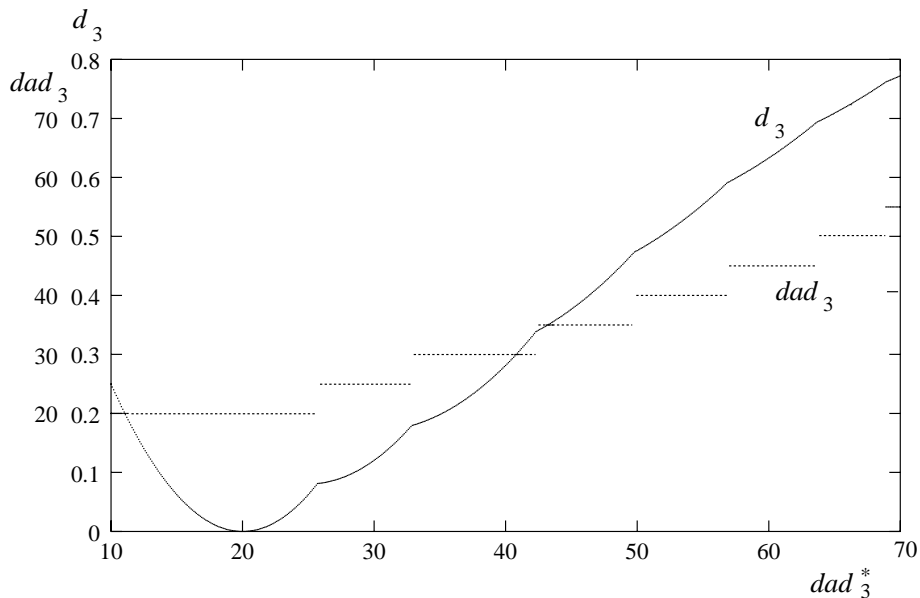


FIG. 5.14 – Dépassement de consignes pour le DTLM 3.

Nous avons voulu mettre en évidence l'influence du basculement d'un DTLM solution à un autre sur les fonctions de dépassement d_i , et donc sur la fonction objectif global f . Pour cela, nous avons isolé une variable consigne pour laquelle nous avons balayé la plage des valeurs possibles, et nous avons observé la fonction de dépassement (toutes choses égales par ailleurs). Les résultats sont présentés sur la figure 5.14 grâce à deux courbes relatives au même module : une des variables dimensionnelles (dad_3) du DTLM (témoin du changement de solution trouvée par le module de calcul) ainsi que la valeur de dépassement (d_3) renvoyée par le module (niveau local) au niveau système. En observant la courbe de dépassement, on aperçoit très clairement une inflexion de la courbe pour des valeurs de consignes (dad_3^*) correspondant précisément à des changements de DTLM (modification de dad_3). Cette courbe est en fait formée d'une succession de courbes exponentielles issues du terme :

$$\left(\frac{dad_3 - dad_3^*}{dad_3^*} \right)^2$$

Les minimums correspondent à la situation d'égalité entre la consigne et la variable dimensionnelle ($dad_3^* = dad_3$). À chaque fois que le module de calcul renvoie un nouveau DTLM, nous obtenons un changement de courbe de la fonction dépassement d_3 et un point de rebroussement. Nous avons donc affaire à une fonction *non-continuellement différentiable*. Ceci est de nature à perturber le processus de recherche de l'algorithme de Powell puisque ces brusques variations se retrouvent logiquement dans l'expression de l'objectif global. Ce phénomène est amplifié lorsque les directions conjuguées conduisent à faire varier plusieurs consignes simultanément, entraînant le changement simultané de plusieurs DTLM.

Par ailleurs, l'exemple qui nous sert de support se caractérise par un faible nombre de solutions locales réalisables (trois) pour la liaison A (cf. tableau 4.2). De plus nous devons compter avec des variations de dimensions significatives entre chacune de ces solutions. Celà, bien sûr, n'est pas de nature à faciliter la tâche de notre algorithme de résolution.

5.3.7 Conclusions.

D'autres essais, non présentés ici, s'appuyant sur des cahiers des charges différents, ont été testés. Ils ont en majorité débouchés sur des solutions proposant un bon objectif global mais où souvent subsistent des conflits. En conclusion, l'algorithme de Powell permet une amélioration significative des solutions non optimales produites par la synthèse découplée, mais échoue dans sa recherche d'une solution réalisable.

5.4 Processus spécifique de synthèse coordonnée.

La procédure de synthèse coordonnée diffère de la synthèse découplée par la prise en charge des contraintes de montage et d'encombrement. L'algorithme que nous avons construit a pour objectif de guider les modules de calcul dans leur choix d'une solution qui soit non conflictuelle et optimale au sens global. Rappelons que cette procédure doit utiliser les composants logiciels (modules de calcul) ainsi que les cahiers des charges déjà disponibles.

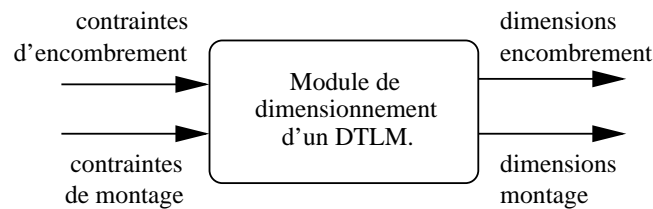


FIG. 5.15 – Paramètres d'entrée et de sortie d'un module de dimensionnement de DTLM.

5.4.1 Présentation de la méthode.

Nous avons cherché à mettre en place une méthode plus simple et plus intuitive, mais donc moins générale et plus proche du problème particulier qui nous intéresse (synthèse des DTLM). Nous avons élaboré une heuristique en nous inspirant :

- des méthodes de relaxation ;
- de la manière intuitive qu'adopte le concepteur lorsqu'il guide le processus de synthèse en agissant sur les contraintes. Cette démarche est basée sur le principe de la synthèse découplée avec la prise en charge supplémentaire des contraintes de couplage.

Nous considérons que la synthèse coordonnée ne doit pas s'appuyer sur d'hypothétiques règles issues de l'observation de l'évolution des DTLM. Une telle démarche conduirait immanquablement à élaborer autant de règles que de cas de figures envisageables.

L'heuristique retenue s'appuie sur la formulation du problème présentée sur la figure 5.16. Le niveau système a pour objectif la minimisation de la fonction objectif f sous les contraintes

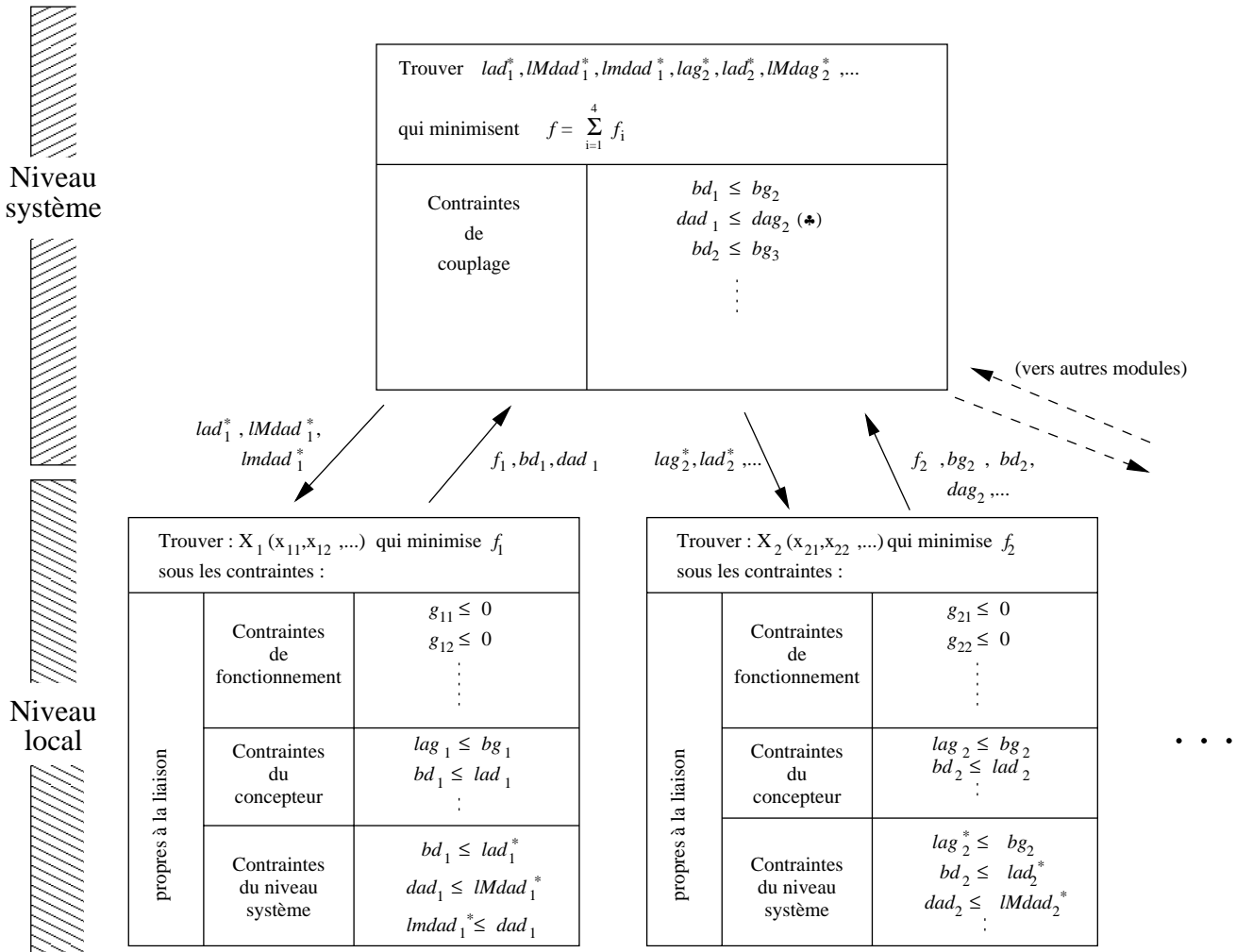


FIG. 5.16 – Découpage hiérarchique du problème pour la méthode de relaxation sélective.

de couplage. Nous observons dans ce découpage, que les variables du niveau système deviennent des limites sur les paramètres de couplage. Ces limites servent à construire un jeu de contraintes supplémentaires pour chacun des modules locaux. Ces contraintes permettent de contrôler le travail des modules locaux afin d'assurer le couplage. Remarquons que ces contraintes sont de même nature que les contraintes imposées par le concepteur. Elles se superposent aux contraintes locales existantes et peuvent ainsi, en pratique, être regroupées sous la forme :

$$\left\| \begin{array}{l} bd_1 \leq \min(lad_1, lad_1^*) \\ dad_1 \leq \min(lMdad_1, lMdad_1^*) \\ dad_1 \geq \max(lmdad_1, lmdad_1^*) \end{array} \right. \quad \text{exemple pour le module du DTLM 1}$$

5.4.2 Description de l’algorithme de relaxation sélective des contraintes.

Notre démarche consiste en la recherche d’un point de départ réalisable, puis en une optimisation de cette solution.

5.4.2.1 Point de départ du processus.

Bien que le principe du point de départ réalisable soit séduisant, la probabilité pour qu’une telle solution émerge naturellement du cahier des charges est faible. En effet, les liaisons entre les différents DTLM sont réalisées par des contraintes qui lient les variables de conception entre-elles. Par exemple, les contraintes de montage imposent des inégalités entre les diamètres de tronçon d’arbre consécutifs tandis que les contraintes d’encombrement imposent également des contraintes inégalités entre les bornes axiales de chaque DTLM voisin (cf. §2.2.2). Nous avons cherché quelle pouvait être la démarche qui fournirait, *à coup sûr*, une solution de départ réalisable, si tant est qu’il en existe une. L’obtention d’une telle solution de départ peut se faire en lançant un premier calcul pour chacun des DTLM, *de façon indépendante* ! Nous nous plaçons volontairement dans le contexte suivant en début de synthèse coordonnée :

- le concepteur a choisi les types de DTLM à mettre en œuvre ;
- les CdC global et les CdC locaux sont renseignés ; le concepteur a éventuellement fixé des limites axiales et radiales d’encombrement mais celles-ci peuvent être suffisamment lâches pour conduire à des conflits (cf. §4.5.1) ;
- les objectifs locaux sont fixés ;
- les règles de montage sont connues ;
- nous disposons d’un premier jeu de DTLM provenant de la synthèse découplée.

5.4.2.2 Recherche d’une solution réalisable.

La prise en compte simultanée des contraintes de montage et d’encombrement est à l’origine du problème de synthèse coordonnée. Nous avons vu que ces contraintes peuvent être antagonistes, compte tenu des lois d’évolution dimensionnelle de certains DTLM. La première phase de l’algorithme de relaxation a pour objectif de générer une première proposition présentant le moins de conflits possibles (violations de contraintes) puis d’améliorer ce premier jet pour arriver à une solution réalisable. C’est de cette solution réalisable que l’on partira pour tendre vers une solution réalisable optimale.

1^{re} étape : Création d’une première proposition. Dans cette étape, nous ne tenons pas compte des contraintes de couplage (i.e. encombrement dus à la présence des DTLM voisins et montage). Les calculs sont réalisés avec des objectifs locaux qui permettent d’obtenir une solution générant un minimum de conflit. En l’occurrence, nous avons choisi une optimisation de l’encombrement axial pour tous les DTLM. Cette démarche nous permet d’éliminer tout conflit axial. En effet, si à l’issue de ce calcul, il existe des conflits axiaux, alors nous pouvons affirmer avec certitude qu’il n’existe aucune solution à notre problème. Ceci se comprend aisément en observant que si les DTLM, au minimum de leur encombrement axial et sans tenir compte des contraintes de montage, sont déjà en conflit axial, il ne saurait en être autrement avec des objectifs d’optimisation différents et des contraintes supplémentaires. La figure 5.17 illustre le

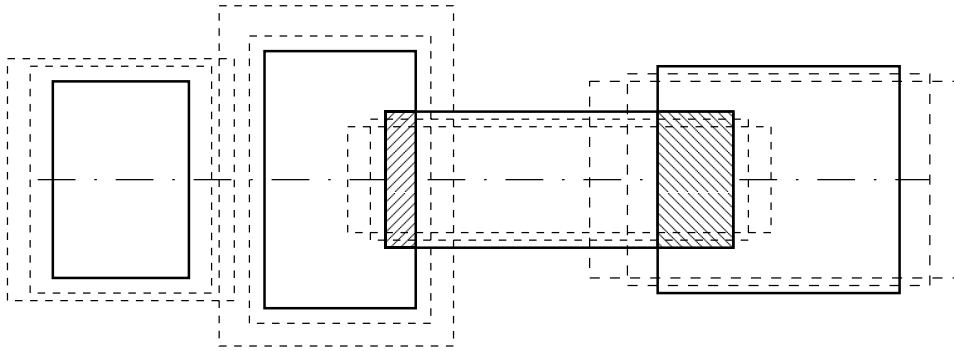


FIG. 5.17 – *Conflit axial à l'issue de la première étape malgré une optimisation de l'encombrement axial.*

cas où le DTLM 3 est en conflit avec ses voisins. Dans le cas contraire, rien n'indique cependant qu'il existe une solution réalisable. Il nous faut maintenant introduire les contraintes de montage. Si ces dernières ne provoquent pas de conflit, nous avons doré et déjà une solution réalisable. Nous pouvons alors passer à la phase suivante, c'est-à-dire l'optimisation de la solution. Sinon, il nous faut encore faire évoluer cette proposition vers une solution réalisable. Ces alternatives sont illustrées par l'algorithme partiel de la figure 5.18. Précisons que l'écriture des algorithmes a été limitée à quatre DTLM dans un souci volontaire de clarté.

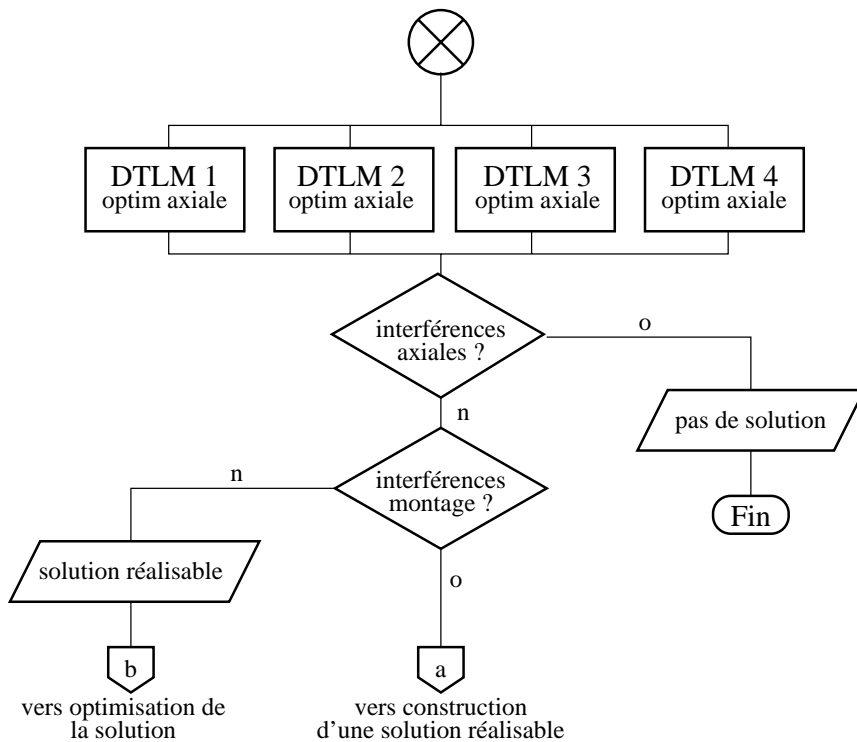


FIG. 5.18 – *Algorithme de création d'une première proposition.*

2nd étape : Construction d'une solution réalisable. Nous avons mis en avant, précédemment, la nature du problème de dimensionnement du DTLM. Nous pouvons le considérer comme bidimensionnel puisque les dimensions axiales et radiales sont simultanément impliquées. Nous utilisons cette propriété pour mener la recherche d'une solution réalisable de la façon suivante. La

première étape a fait apparaître un espace entre chacun des DTLM. En reprenant la numérotation des DTLM évoquée ci-avant, nous définissons ces intervalles de la façon suivante :

$$\begin{aligned}\Delta_{ij} &= (X_j - eg_j) - (X_i + ed_i) \\ &= bg_j - bd_i\end{aligned}$$

correspondant à l'espace entre les frontières axiales des DTLM i et j . Nous pouvons exprimer la contrainte de non interférence axiale sous la forme :

$$bd_i \leq bg_{i+1}$$

ceci respectant la formulation du problème de niveau système (cf. figure 5.16). Ces contraintes doivent venir s'ajouter aux contraintes d'encombrement déjà évoquées en §2.2.2 pour chacun des DTLM. Nous relevons ainsi trois marqueurs Δ_{12} , Δ_{23} et Δ_{34} comme représenté sur la figure 5.19. Ces espaces doivent être considérés comme un *degré de liberté* sur les dimensions axiales.

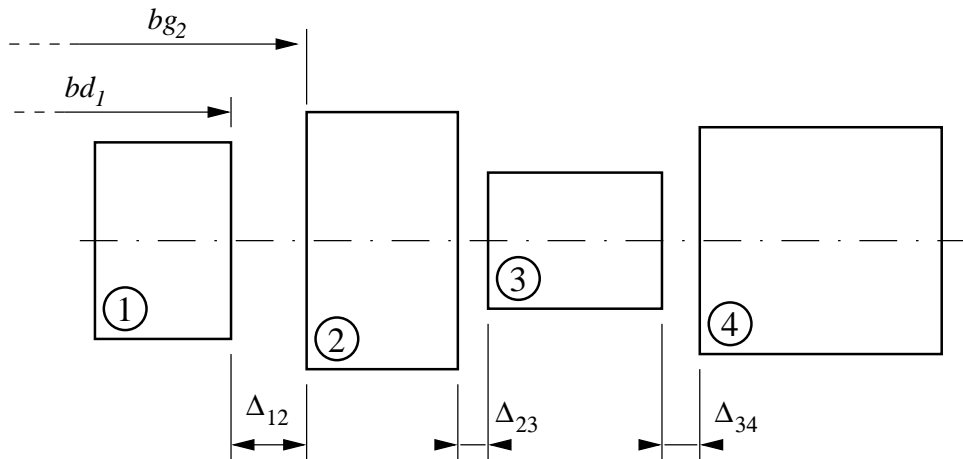


FIG. 5.19 – *Espaces axiaux inter-DTLM.*

Nous allons « jouer » avec ces espaces, que nous nommons *reliquats*, pour tenter d'éliminer les conflits de montage. Pour cela, l'algorithme doit s'assurer de la meilleure redistribution de ces espaces vers les DTLM pour permettre une modification dimensionnelle réduisant les conflits de montage. Considérons le cas représenté sur la figure 5.20. Les deux tronçons d'arbre épaulés sont caractéristiques d'un montage par la droite. Les deux DTLM tels que représentés en trait continu présentent donc un conflit de montage (Il faut atteindre une solution où dad_1 est supérieur ou égal à dag_2). Nous voyons apparaître un nouveau paramètre nommé ε_{12} (ou ε_{ij} sous sa forme générale) correspondant au demi-écart entre les diamètres de tronçons d'arbre consécutifs :

$$\varepsilon_{12} = \frac{dad_1 - dag_2}{2}$$

Notons que, contrairement aux contraintes de non interférence, l'écriture de la contrainte de montage doit tenir compte des règles de montage imposées par le concepteur. Ceci explique pourquoi nous ne pouvons donner une formulation générale de cette contrainte. Nous obtenons pour le cas présenté :

$$\varepsilon_{12} \geq 0 \quad \text{ce qui équivaut également à} \quad dag_2 \leq dad_1$$

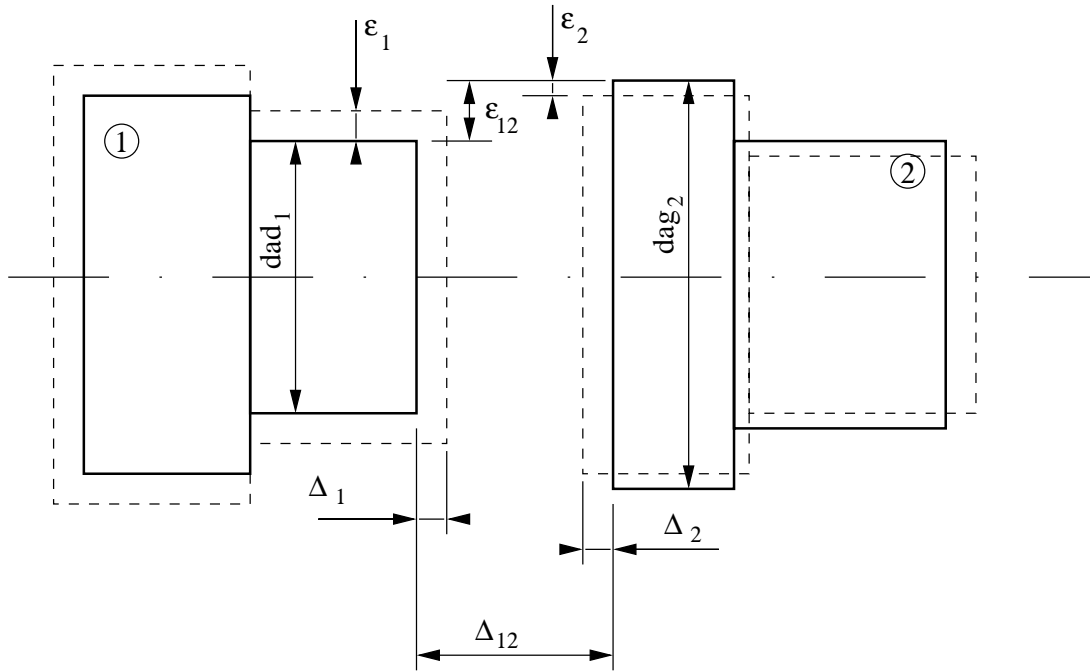


FIG. 5.20 – Évolution du conflit radial.

Remarquons enfin sur la figure 5.20 les paramètres Δ_i et ε_i : ils représentent les pas de variation dimensionnelle des DTLM selon le sens axial et radial. Ces pas sont les plus petites variations dimensionnelles causées par un changement de DTLM. Nous précisons leur utilisation ci-après.

Plusieurs méthodes sont envisageables pour redistribuer l'espace axial entre chaque DTLM (les Δ_{ij}). L'une d'elles, s'apparentant à de la dichotomie, consiste à découper les espaces en parts, égales ou non, que l'on ajoute à l'espace alloué au DTLM. Nous avons relevé un problème relatif à ces méthodes. Compte tenu de la nature discrète d'un grand nombre de paramètres dimensionnels des DTLM, ces pratiques risquent d'être pénalisantes car elles ne tiennent pas compte des pas de variation des dimensions des DTLM (qui sont parfois très différents).

La démarche que nous avons choisie consiste à faire évoluer les DTLM en utilisant pour cela le *pas minimal de variation* de chacun des DTLM. Cela revient à parcourir l'espace des solutions de chacun des DTLM en passant de proche en proche. La technique consiste à relancer les calculs pour chacun des DTLM en interdisant la dernière solution fournie. Ceci impose au module de calcul de fournir une nouvelle proposition de DTLM allant dans le sens imposé par l'algorithme. En pratique, on modifie le CdC liaison pour exclure les dernières dimensions de diamètre de tronçons d'arbre obtenues. Ceci est rendu possible par l'utilisation d'un jeu de contraintes inégalités supplémentaire dont les valeurs limites sont prises égales aux dernières dimensions obtenues. Ainsi, nous forçons les modules de calcul à fournir une nouvelle solution. Nous avons alors la certitude d'obtenir la solution immédiatement suivante (en terme de diamètre d'arbre).

Chaque Δ_{ij} et ε_{ij} est évalué au début de l'itération. Le sens de montage impose le sens de la variation de diamètre que doit suivre chaque DTLM. L'établissement d'un tableau de variation montre que pour les DTLM centraux (i.e. encadrés de part et d'autre), certains cas de figure imposent des sens de variation antagonistes sur les diamètres d'arbre. La figure 5.21 montre l'établissement de tels tableaux pour deux cas d'étude dont seule la règle de montage diffère (symbolisé par le profil triangulaire). Tandis que le cas noté (a) ne présente aucune ambiguïté sur l'évolution attendue des DTLM, le cas (b) impose au DTLM 2 une évolution conflictuelle. Le

DTLM 1 tend à imposer¹⁰ une augmentation du diamètre de tronçon d'arbre gauche du DTLM 2 tandis que le DTLM 3 impose une diminution du diamètre de tronçon d'arbre droit du DTLM 2. Or nous avons vu au chapitre 2 que les diamètres de tronçons d'arbre gauche et droit sont liés. Imposer une évolution de diamètre sur l'un d'eux revient à imposer une évolution de diamètre sur le DTLM complet. Ceci à une conséquence directe sur le processus de construction d'une

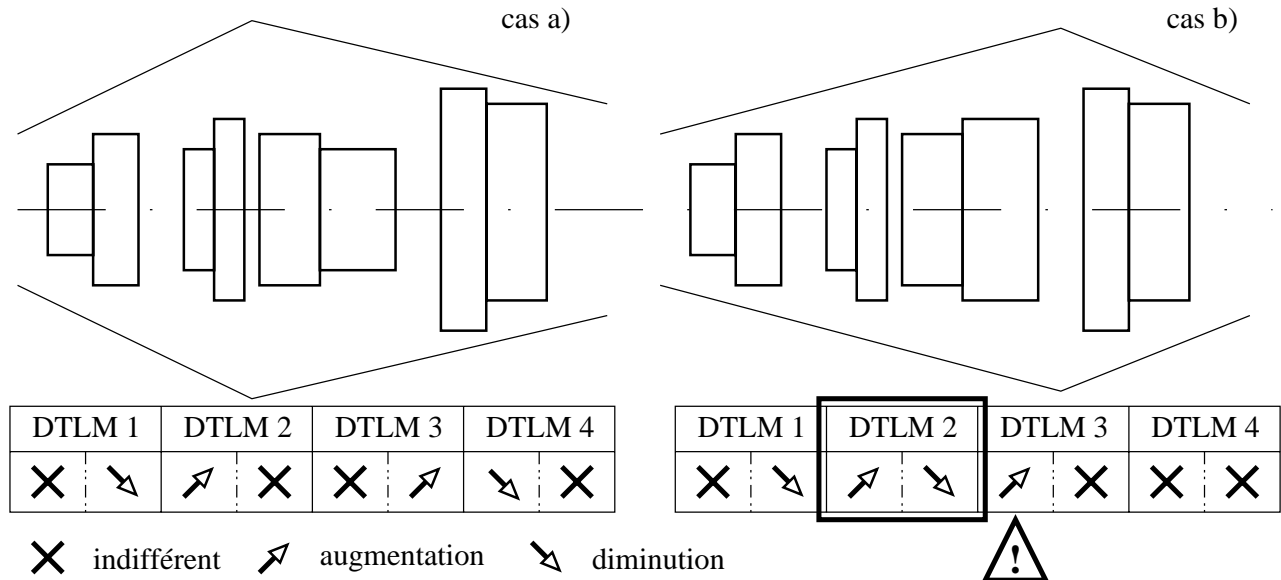


FIG. 5.21 – Tableaux variationnels des DTLM.

solution réalisable. Nous devons tester les deux sens d'évolution afin de savoir lequel procure la meilleure évolution. Ainsi, nous devons prévoir à chaque itération un calcul pour les DTLM situés aux extrémités et deux calculs pour les DTLM centraux. Le nombre de relance des modules de calcul à chaque itération est donc :

$$N_{calc} = (2 * n) - 2 \quad \text{avec } n = \text{nombre de DTLM}$$

soit ici six appels aux modules de calcul.

A chaque nouveau calcul de DTLM, nous affectons un *indice de capacité d'évolution*. Cet indice est constitué d'un rapport entre la variation du diamètre d'arbre et la variation de longueur du DTLM. Ce rapport témoigne de la bonne utilisation du reliquat d'espace axial sur la progression radiale. Considérons l'exemple de la figure 5.22. Le DTLM 1 est en conflit radial avec le DTLM 2. Nous avons représenté ce que pourraient être les étapes d'évolution de chacun des DTLM vers une résolution du conflit (augmentation du diamètre pour le DTLM 1 et diminution pour le DTLM 2). Nous constatons que l'évolution du DTLM 1 permet d'accéder à une résolution du conflit tandis que l'évolution du DTLM 2 ne le permet pas. En favorisant la diminution la plus rapide de l'interférence radiale, on risque de favoriser le type de comportement observé sur la partie droite de la figure 5.22. La mise en place de l'indice permet d'observer que l'évolution (1 ⇒ 1') présente un meilleur rapport $\frac{\text{espace axial perdu}}{\text{interférence axiale gagnée}}$ que l'évolution (2 ⇒ 2'). L'indice de capacité peut s'exprimer de façon générale :

$$Q_i = \left\| \frac{\varepsilon_i}{\Delta_i} \right\| \quad Q_i \text{ indice de capacité du DTLM } i$$

10. En réalité, c'est la résolution du conflit de montage qui implique les réductions ou augmentations de diamètres d'arbre!

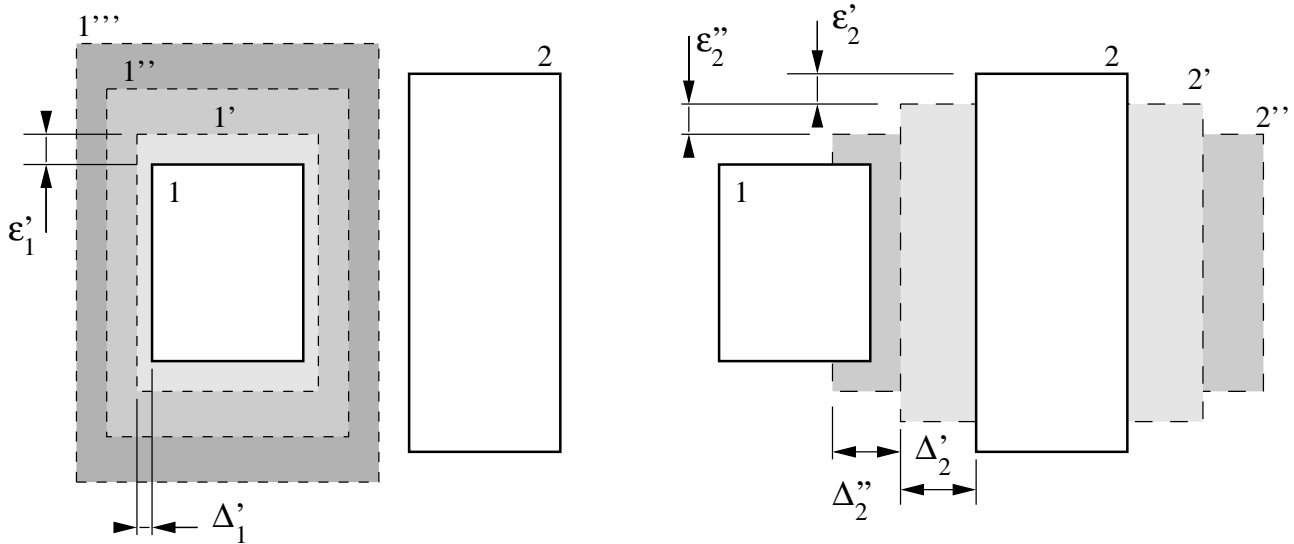


FIG. 5.22 – Influence de la vitesse d'évolution d'un DTLM sur la résolution des conflits radiaux.

Cet indice étant réévalué à chaque nouveau calcul de DTLM, il peut donner lieu à une évolution alternée des DTLM. Cette démarche est rendue indispensable par la nature discrète du comportement des DTLM.

A chaque nouvelle proposition de DTLM, l'algorithme évalue la nouvelle somme des interférences radiales et la compare à la valeur initiale (début d'itération). La modification retenue est celle qui :

- conduit à une diminution de la somme des interférences radiales ;
- ne viole pas les contraintes d'encombrement axiale ;
- présente le meilleur indice de capacité d'évolution.

Lorsqu'on a identifié un DTLM permettant une amélioration du mécanisme, il est intégré définitivement et l'on peut repartir pour une nouvelle itération.

La boucle se termine lorsqu'il n'existe plus aucune violation de contraintes, tant radiales qu'axiales, ou bien si l'un des modules de calcul ne peut plus fournir de solution (domaines vides). L'algorithme de construction d'une solution réalisable est récapitulé sur la figure 5.23. Nous n'avons pas détaillé les séquences de mise à jour des CdC et d'évaluation des modifications car elles ne présentent pas d'intérêt particulier.

5.4.2.3 Optimisation d'une solution réalisable.

A partir de la solution réalisable obtenue qui sert de point de départ, nous déclenchons une dernière phase qui a pour objectif d'améliorer (optimiser de façon globale) cette solution en restant dans le domaine des solutions réalisables. Cette dernière condition est indispensable pour pouvoir fournir à tout moment, particulièrement en cas de blocage de l'algorithme, une solution exploitable. Nous devons pour cela revenir au critère d'optimisation global. La première opération consiste à redimensionner tous les DTLM avec les objectifs locaux initiaux. Les CdC locaux de chacun des DTLM sont ainsi mis à jour en précisant les nouveaux volumes locaux suivant le voisinage des DTLM. La figure 5.24 illustre cette mise à jour des contraintes de proche en proche. Le DTLM noté i est entouré de deux DTLM qui lui imposent des limites sur les diamètres de tronçons

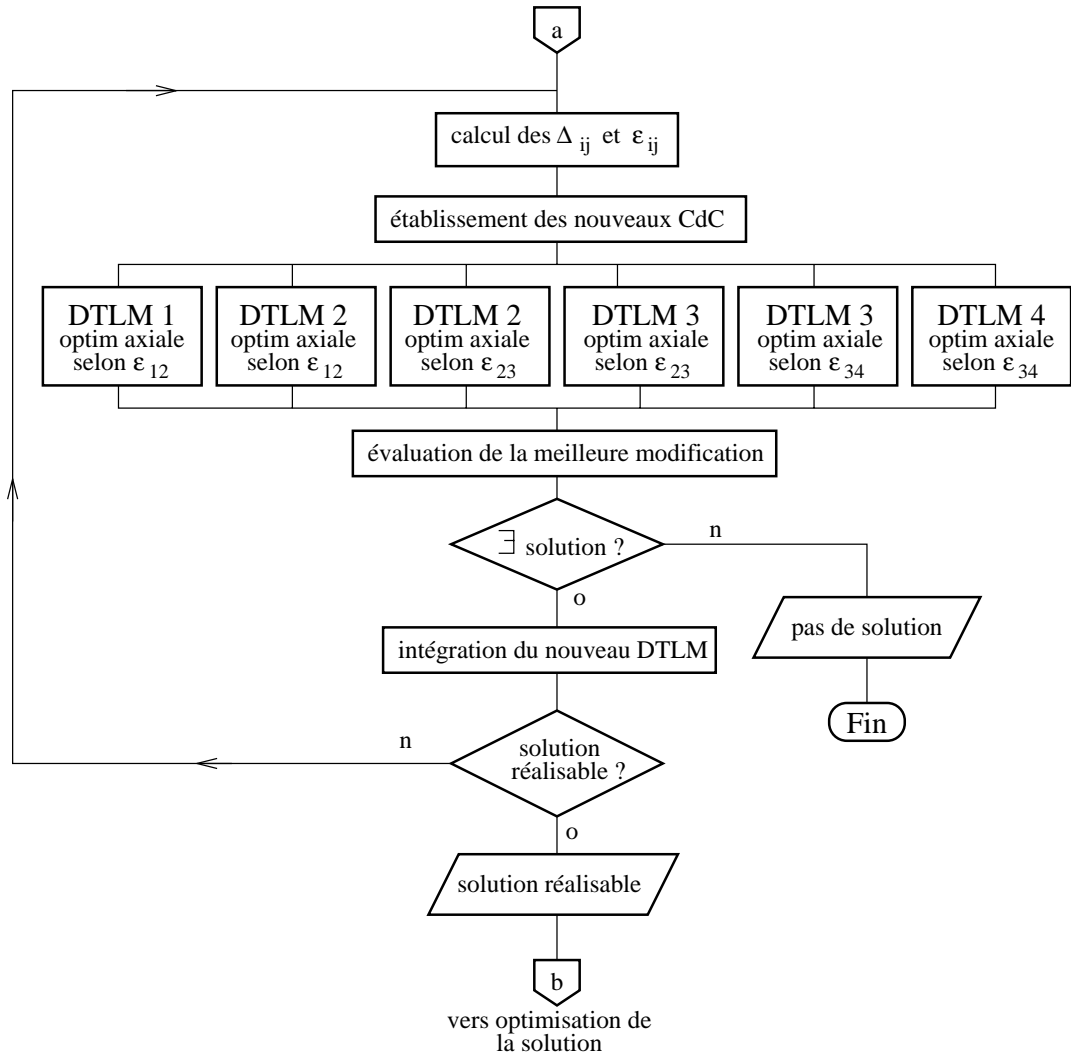


FIG. 5.23 – Algorithme de construction d'une solution réalisable.

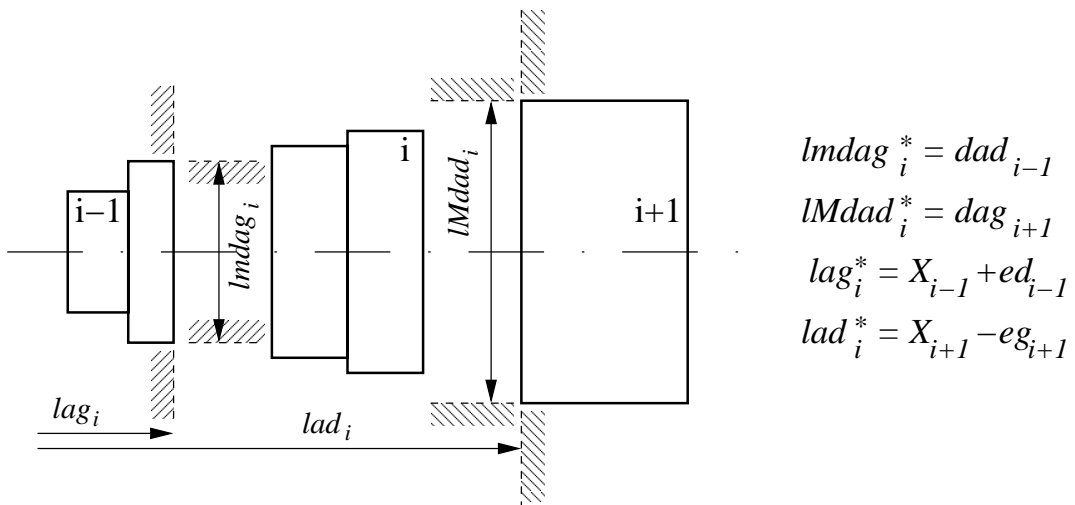


FIG. 5.24 – Principe de la mise à jour des contraintes.

d'arbre à droite et à gauche ainsi que sur l'extension axiale. Ces contraintes viennent s'ajouter aux éventuelles contraintes imposées par le concepteur dans le CdC liaison. L'algorithme établit ainsi les nouveaux CdC de tous les DTLM puis lance les modules de calcul associés. Il évalue alors les gains qui peuvent être obtenus, en terme d'objectif global, en faisant évoluer la géométrie des DTLM. Si une amélioration existe, alors l'algorithme procède à la substitution définitive du DTLM qui a permis cette amélioration. Après cette intégration du nouveau DTLM, on itère de nouveau en actualisant les jeux de contraintes.

L'algorithme sort de la boucle lorsqu'il ne produit plus de solution améliorant l'objectif global.

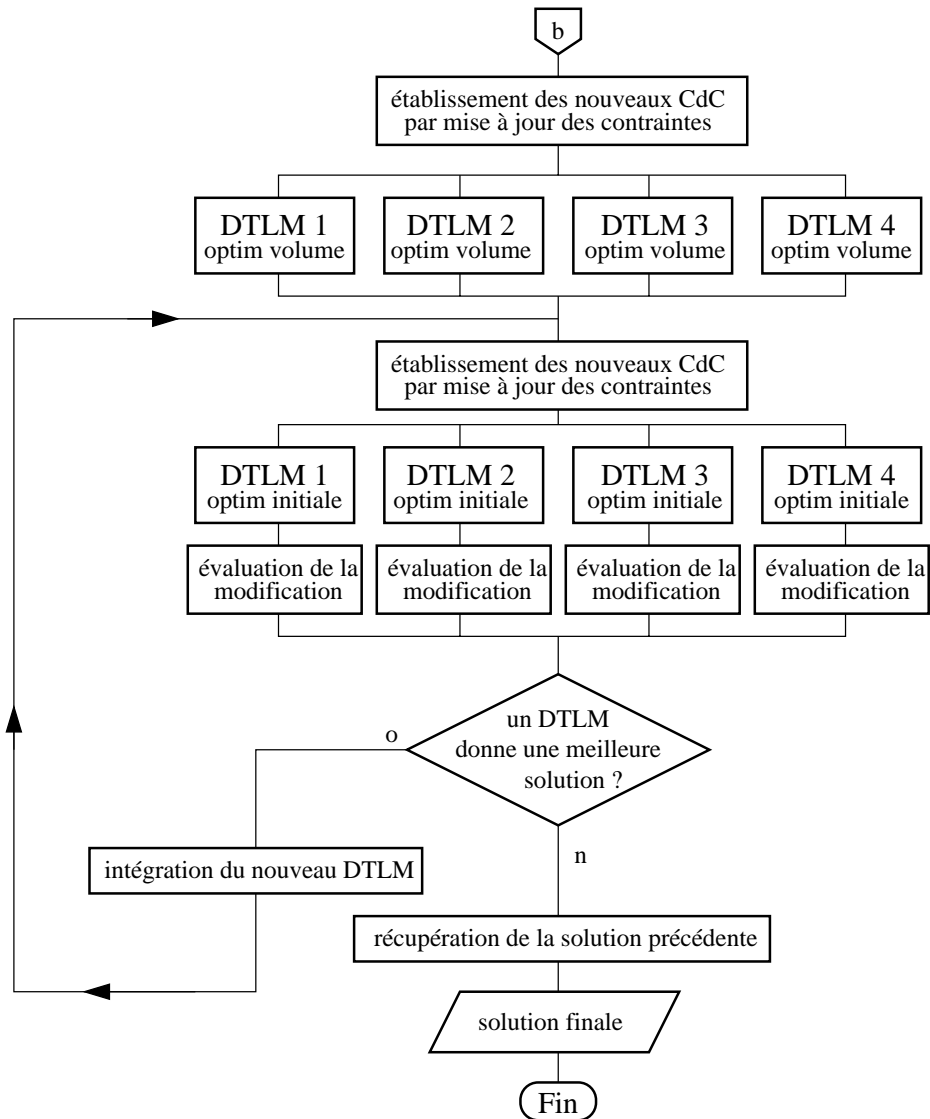


FIG. 5.25 – *Algorithme d'optimisation.*

La figure 5.25 récapitule l'algorithme relatif à l'optimisation d'une solution réalisable.

5.4.2.4 Remarques

Plusieurs remarques peuvent être formulées. Le choix d'un processus itératif est généralement synonyme de temps de traitement important. Nous verrons dans la section suivante qu'il n'en est rien (le traitement en parallèle des processus de dimensionnement des DTLM pourrait

réduire significativement les temps de calcul). Le choix de ne faire évoluer qu'un DTLM à la fois génère une évolution séquentielle de l'algorithme. Ceci nous permet de garder plus facilement le contrôle du processus en limitant le nombre de contraintes modifiées. Il existe cependant un risque de s'enfermer vers un optimum local. C'est pourquoi nous avons mis en place un dispositif permettant d'évaluer ce qui nous a semblé être la meilleure direction d'évolution (grâce à l'indice de capacité d'évolution).

La méthode de progression choisie permet un balayage complet du domaine des solutions. En effet, le principe de rejet du dernier DTLM proposé oblige les modules de calcul à fournir le DTLM solution immédiatement le plus proche. Nous imposons pour cela une très faible modification des valeurs limites sur les contraintes que nous voulons voir évoluer. Cette méthode n'est en rien perturbée par la nature discrète des DTLM. Elle présente toutefois l'inconvénient de multiplier le nombre de cas à tester en diminuant le pas de progression. En conséquence, nous nous interdisons le saut de plusieurs solutions qui permettrait une progression plus rapide de l'algorithme.

Nous présentons ci-après les résultats obtenus par l'application de l'algorithme de relaxation sélective avec un objectif d'optimisation multicritère. Nous discutons ensuite du choix d'un critère d'optimisation homogène. Nous donnons quelques exemples de sa mise en œuvre et comparons la qualité des résultats à ceux produits par la méthode énumérative. Des conclusions sont proposées à l'issue de l'analyse des résultats.

5.4.3 Relaxation sélective avec un objectif multicritère.

L'exemple présenté ci-dessous reprend le cahier des charges du problème traité en §5.3.5. L'objectif global d'optimisation est un objectif multicritère reprenant les critères locaux du cahier des charges (sa formulation est identique à l'équation 5.1). Rappelons que les critères locaux de ce problème ne sont pas homogènes.

Les résultats obtenus (cf. tableau 5.2) sont satisfaisant en ce sens que la solution produite est réalisable et s'approche de la solution optimale (obtenue par la méthode énumérative). Cependant, nous devons déplorer une certaine dégradation des objectifs locaux, et donc de l'objectif global. Ce dernier se situe quasiment à 400% de l'objectif global de la solution optimale. Ce résultat place la solution au 2210^{me} rang parmi les 92046 combinaisons possibles. Ceci peut s'ex-

	DTLM 1	DTLM 2	DTLM 3	DTLM 4	DTLM 1	DTLM 2	DTLM 3	DTLM 4
bg_i	74	107.6	179	246.7	71	105.85	173.5	242
bd_i	107.5	170.3	214.25	307.3	105	172.15	218.5	311.95
dag_i	14	37	57	48.6	25	30	45	34.7
dad_i	17	43.7	50	42	30	36.7	35	28
F_i	33.5	37	50	60.6	34	30	35	69.9
f	2.429				0.61			
	processus spécifique (relaxation)				méthode énumérative			

TAB. 5.2 – Solution obtenue par l'algorithme de relaxation sélective.

pliquer de la façon suivante. La première phase de la méthode de relaxation sélective (*recherche d'une solution réalisable*) se déroule correctement et produit rapidement (en une itération) une solution réalisable. Rappelons que cette opération débute par une optimisation en largeur de tous les DTLM. Or, lorsqu'on réintroduit les critères d'optimisation de départ (pour la phase d'*optimisation de la solution réalisable*), les DTLM centraux (DTLM 2 et 3) sont de nouveaux optimisés en diamètre tandis que les DTLM extrêmes conservent le critère d'optimisation en largeur.

Ils sont donc déjà dans un état d'optimum local. Nous nous retrouvons ainsi dans le cas typique illustré par la figure 5.21 où les DTLM présentent deux à deux des sens de variation antagonistes. Comme les DTLM extrêmes sont déjà optimisés, ils ne peuvent subir d'amélioration supplémentaire. Ils bloquent donc la progression des DTLM centraux. Précisons cependant que ce cas est un des cas les plus difficiles en matière de sens de progression antagonistes, et que malgré cela, les résultats sont satisfaisants. Une amélioration envisageable consisterait, de manière générale, à pouvoir dégrader l'objectif local de certains DTLM pour permettre l'évolution de ses voisins.

Cette observation nous a conduit à étudier l'intérêt d'une optimisation par la méthode de relaxation sélective en utilisant un objectif homogène. Ceci devrait éviter les problèmes inhérents aux changements partiels de critères d'optimisation locaux. La section suivante présente notre réflexion sur ce sujet.

5.4.4 Relaxation sélective avec des objectifs homogènes.

Le concepteur peut, durant la phase de synthèse découplée, mener une recherche de solution globale par des synthèses individuelles de DTLM présentant des objectifs d'optimisation différents. Il a le loisir de favoriser dynamiquement tel ou tel DTLM au fur et à mesure de l'obtention des résultats individuels. Nous avons montré l'intérêt d'une telle démarche. Implicitement, le concepteur est amené à y mettre en œuvre une logique de pondération des objectifs locaux lorsque la construction d'une solution d'ensemble l'amène à faire des arbitrages (résolution de conflits). Dans le cas où la recherche d'une solution globale est confiée à un algorithme, on trouve nécessairement le besoin de disposer d'un critère global afin de départager les solutions concurrentes. Une des possibilités, bien connue en optimisation, est de construire cet objectif global en faisant une somme pondérée des objectifs locaux *adimensionnés*. La principale difficulté de ce type d'optimisation multicritère est l'évaluation, par le concepteur, des coefficients de pondération.

Le maintien des objectifs locaux oblige le concepteur à préciser, dès le début de la phase de synthèse, les coefficients de pondération. On ne peut alors concevoir une possibilité de modification de ces coefficients avant la fin du processus automatique sans l'interrompre. Cette interruption serait bien entendu préjudiciable à tout processus de recherche automatique.

La présence d'objectifs locaux différents, voire antagonistes, constitue également un obstacle au processus automatique en rendant invisible des évolutions antagonistes de DTLM. Ceci a pour conséquence de complexifier la recherche d'une « bonne direction » d'optimisation.

Dans le cadre d'une procédure de synthèse coordonnée, nous avons cherché à savoir si le processus était plus robuste et plus efficace lorsqu'on utilisait un seul et même objectif d'optimisation.

Nous avons choisi d'optimiser le volume de l'arbre. La détermination du volume de l'arbre est faite en additionnant les volumes de tronçons d'arbre de chaque DTLM.

5.4.4.1 Caractérisation des problèmes.

Nous illustrons ci-dessous les différentes configurations de problèmes que peut rencontrer le concepteur lors d'une synthèse coordonnée.

Problème surcontraint par le concepteur. Ce que nous entendons par problème surcontraint par le concepteur est le cas d'étude où ce dernier a généré des CdC liaisons dont les limites forcent la production de DTLM non conflictuels. Les DTLM sont suffisamment espacés pour dégager des espaces d'encombrements axiaux larges et ne présentant pas de

zones de chevauchement. Les valeurs limites de diamètre d'arbre sont également réglées de telle façon que les DTLM produits respecteront les règles de montage. Bien entendu, ce genre de situation est idéale et ne peut survenir que lorsque le concepteur ne recherche pas une solution compacte. Le processus de synthèse coordonnée n'apporte en ce cas aucun avantage par rapport à une synthèse découplée, si ce n'est l'évaluation de la solution globale. C'est typiquement le cas où le nombre de combinaisons de DTLM à tester est le plus faible. Nous ne détaillons pas de cas de ce type qui ne présentent aucun intérêt puisque le concepteur a déjà réglé les conflits.

Problème semi-contraint. Le concepteur a établi des frontières axiales qui empêchent tout chevauchement (cf. figure 5.27). La synthèse coordonnée a pour but de gérer les conflits radiaux dus aux contraintes de montage. La rédaction de CdC menant à des problèmes semi-contraints peut présenter un inconvénient majeur. En effet, lorsque le concepteur fixe des limites d'encombrement axial, il restreint volontairement le domaine des solutions. Ce faisant, il peut éventuellement se barrer l'accès à la solution optimale si la frontière axiale entre deux DTLM voisins n'est pas judicieusement placée (cf. figure 5.26). La solution à ce travers consiste alors à relaxer les contraintes axiales, c'est-à-dire autoriser le chevauchement des DTLM pour obtenir un plus grand échantillon de solutions. Nous parlons alors de problème sous-contraint.

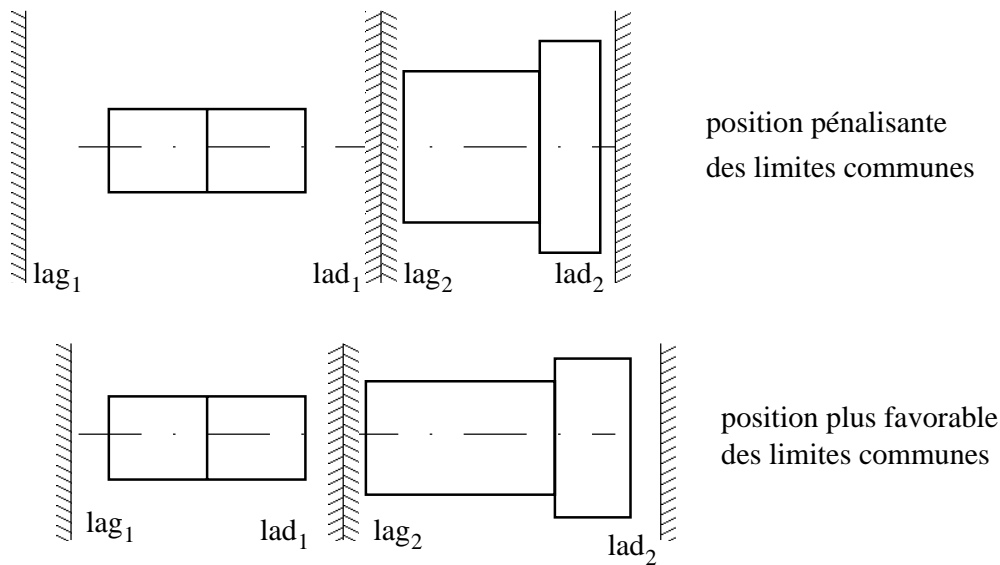


FIG. 5.26 – Influence du positionnement des limites axiales en problème semi-contraint.

Problème sous-contraint. Le concepteur a neutralisé toutes les valeurs limites d'encombrement axial et de diamètres d'arbre. Il laisse alors au processus de synthèse coordonnée le soin de gérer l'intégralité des conflits. Comme dans le cas du problème semi-contraint, le processus doit gérer les conflits de montage et la distribution du reliquat d'espace inter-DTLM axial. On doit cependant noter que l'absence de frontières imposées entre chaque DTLM peut considérablement augmenter le nombre de DTLM solutions. Ceci est possible grâce à la disponibilité totale de l'espace inter-DTLM axial, contrairement au problème semi-contraint où les limites axiales imposées par le concepteur partagent cet espace, et limite donc les possibilités d'extension des DTLM. C'est dans le cas de problèmes sous-contraints que le nombre de combinaisons s'avère le plus élevé.

Les exemples présentés ci-dessous illustrent les cas de problèmes semi-contraints et sous-contraints. Nous n'avons pas repris le cahier des charges du problème utilisé précédemment car, les objectifs locaux étant changés, il aurait été impossible de mener une quelconque comparaison. Nous avons donc préféré partir sur de nouveaux cahiers des charges typiques des cas de problèmes semi-contraints et sous-contraints.

5.4.4.2 Cas des problèmes semi-contraints.

Considérons le premier exemple présenté ci-dessous. Ce problème est un problème de type semi-contraint. Les limites axiales imposées à chacun des DTLM empêchent tout chevauchement comme le laisse entrevoir la figure 5.27. Si les modules de calcul renvoient tous un DTLM solution,

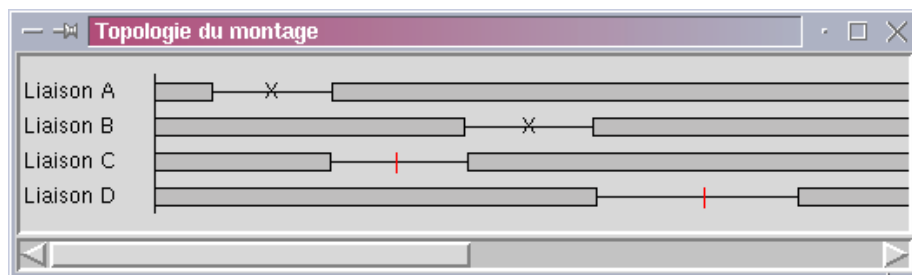


FIG. 5.27 – Cas typique de limites axiales d'un problème semi-contraint.

on a alors l'assurance de pouvoir passer la première étape de la phase de recherche d'une solution réalisable.

Le CdC du problème que nous traitons génère le volume de DTLM solutions suivant :

DTLM	Nb. de solutions
A	27
B	75
C	31
D	53

La combinaison de toutes ces solutions par la méthode énumérative conduit à évaluer 3327075 combinaisons possibles (ce qui est réalisé en un peu plus de 6 secondes).

Le lancement individuel de chacun des modules de calcul donne la solution conflictuelle nommée « proposition initiale » dans le tableau 5.4. Ce résultat est obtenu en appliquant le même critère d'optimisation en volume de l'arbre à tous les DTLM. Nous avons alors engagé une procédure de synthèse coordonnée par relaxation sélective avec un critère objectif homogène sur ce problème.

La première proposition générée par l'étape de création d'une première proposition (le critère d'optimisation vise alors l'encombrement axial) donne un résultat présentant des interférences de montage entre les DTLM C et D. Notons que le point haut est situé entre les deux DTLM centraux et qu'il ne peut donc exister de conflit de montage en cet endroit. La figure 5.28 représente cette première proposition. Seuls les tronçons d'arbre sont visualisés afin de faciliter la lecture des interférences de montage, et les espacements axiaux ont été réduits pour augmenter la compacité de l'image. On constate aisément que le montage de composants sur le tronçon d'arbre placé en 3^{me} position est impossible. L'étape de construction d'une solution réalisable tente ensuite d'éliminer ces interférences. L'analyse de l'historique révèle qu'elle en vient à bout en 4 itérations. La solution ainsi obtenue présente un volume de 70,37 cm³. Survient enfin la phase

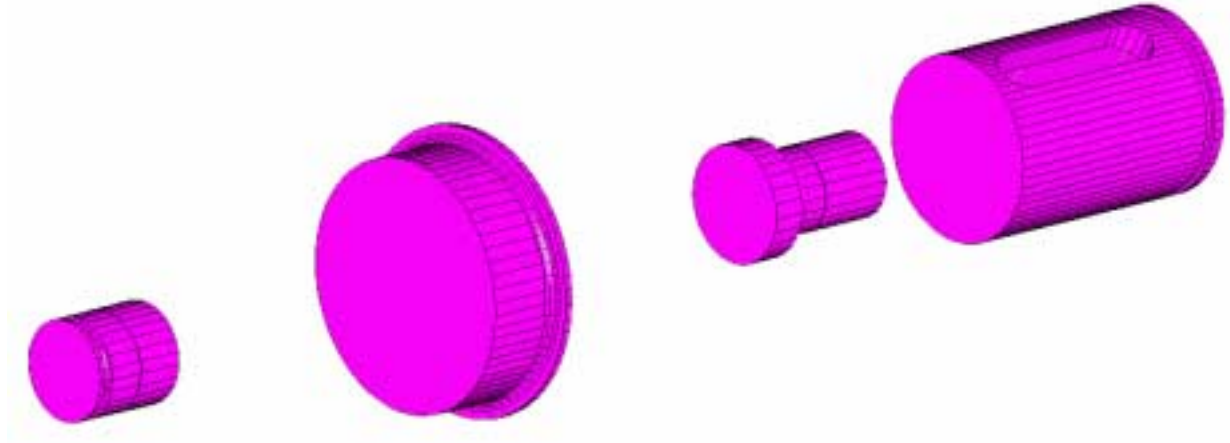


FIG. 5.28 – *Première proposition du problème semi-contraint.*

d'optimisation au terme de laquelle on obtient une solution présentant un volume de $15,12 \text{ cm}^3$. Cette solution est visible sur la figure 5.29. Le tableau 5.3 donne les cinq meilleures combinaisons

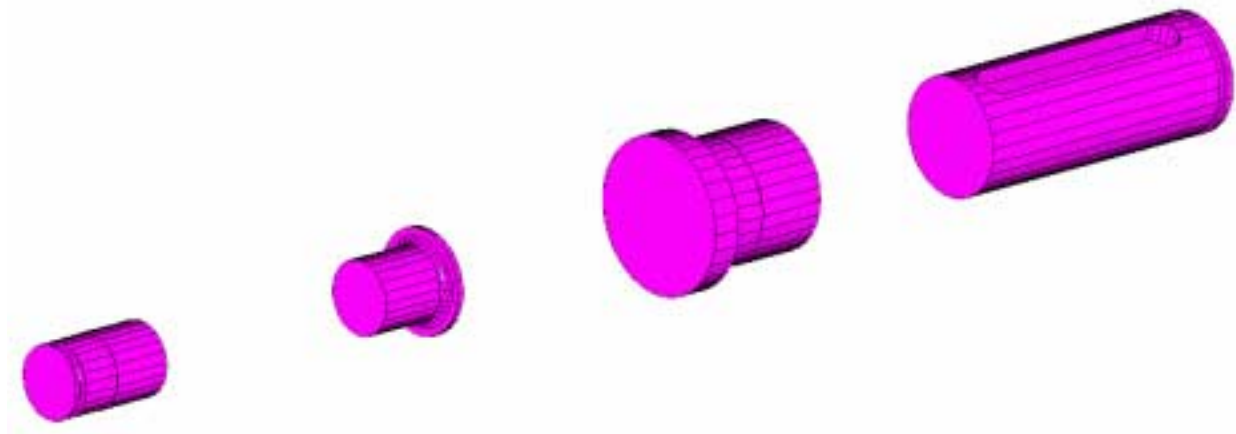


FIG. 5.29 – *Solution finale du problème semi-contraint.*

obtenues par la méthode énumérative. On constate que la solution proposée par l'algorithme de synthèse se classe en 1^{re} position! La solution finale complète est représentée sur la figure 5.30. Son obtention (hors procédure d'affichage du modèle 3D) a pris moins de 4 secondes. Le tableau 5.4 récapitule les caractéristiques de la première proposition et de la solution finale. Ce tableau fournit, à titre indicatif, l'objectif global ainsi que les objectifs locaux.

5.4.4.3 Cas des problèmes sous-contraints.

Considérons maintenant un problème de type sous-contraint présentant un cahier des charges différent du problème précédent. Nous avons augmenté la difficulté en relaxant les contraintes axiales sur les DTLM et en rapprochant les DTLM (figure 5.31). De plus, nous avons situé le point haut de l'arbre à l'une de ses extrémités (droite), ce qui augmente la probabilité d'occurrence de conflits de montage comme nous l'avons expliqué précédemment. Ces interférences peuvent apparaître sur des DTLM encadrés de part et d'autre lorsque les directions d'évolution imposées par les DTLM voisins sont antagonistes. Les CdC utilisés conduisent aux combinaisons suivantes :

Rang de la solution	Volume en cm ³
1	15,12
2	15,28
3	15,38
4	15,38
5	15,43

TAB. 5.3 – Classement des dix meilleures solutions obtenues par la méthode énumérative.

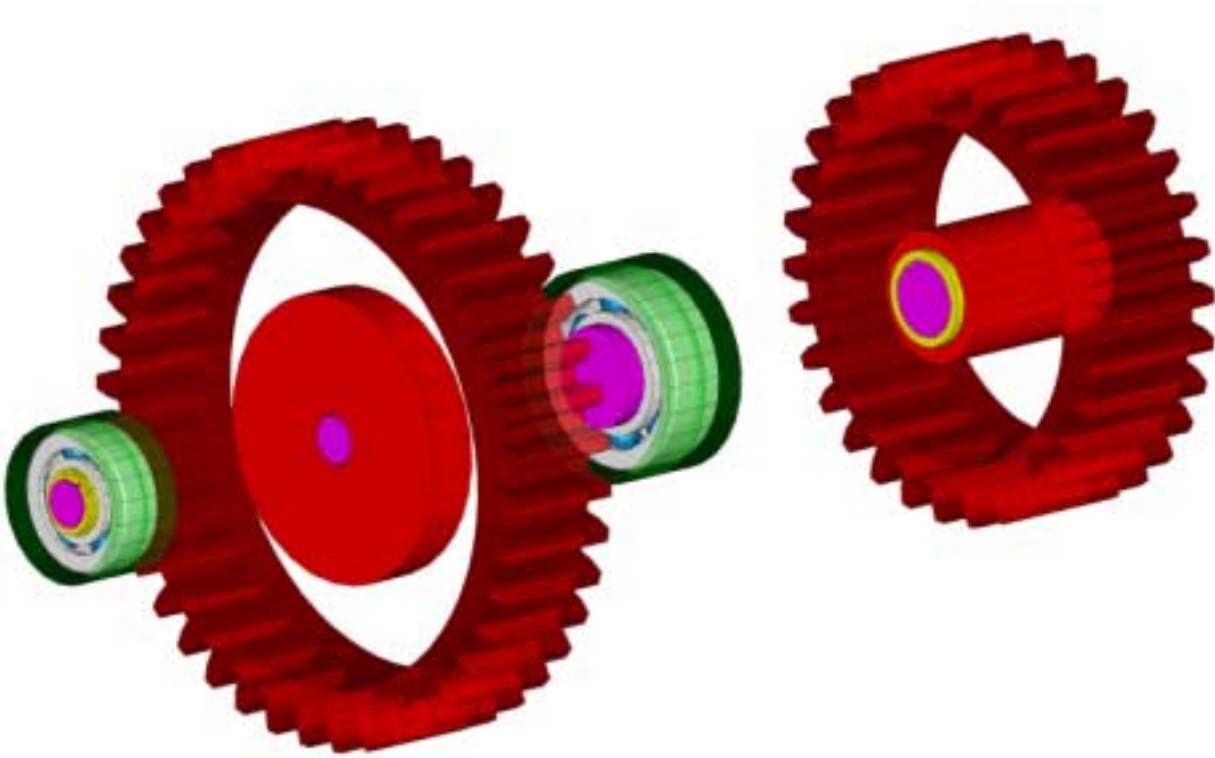


FIG. 5.30 – Modèle 3D complet de la solution optimisée.

	DTLM 1	DTLM 2	DTLM 3	DTLM 4	DTLM 1	DTLM 2	DTLM 3	DTLM 4
bg_i	58,5	133,3	205,5	295,3	58,5	133,3	209	295,3
bd_i	75,5	144,7	224,5	336,7	75,5	144,7	221	336,7
dag_i	10	10	14	16	10	10	22	16
dad_i	10	14,2	10	16	10	14,2	20	16
F_i	1,33	1,31	2,15	8,32	1,33	1,31	4,16	8,32
f	13,11				15,12			
	proposition « initiale »				solution finale			

TAB. 5.4 – Tableau récapitulatif du problème semi-contraint.

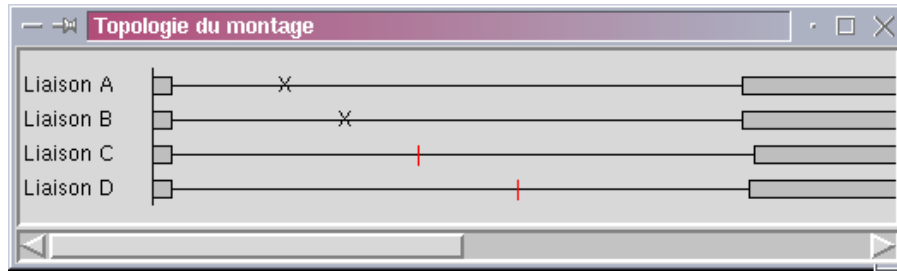


FIG. 5.31 – Cas typique de limites axiales d'un problème sous-contraint.

DTLM	Nb. de solutions
A	15
B	67
C	128
D	179

Ceci autorise 23026560 combinaisons que la méthode énumérative évalue en un peu plus de 3 secondes. La figure 5.32 représente une copie d'écran de la fenêtre de visualisation des volumes de l'ensemble des DTLM solutions (cf. §4.5.2 pour la description de cet outil). Nous constatons que l'association d'un certain nombre de ces DTLM est susceptible de générer des conflits. Examinons

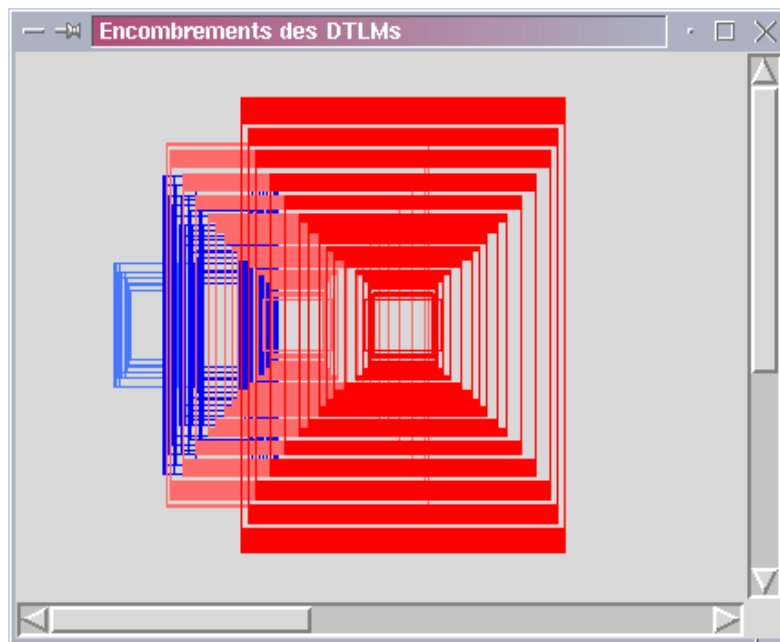


FIG. 5.32 – Ensemble des DTLM solutions d'un problème sous-contraint.

la *première proposition* générée par l'algorithme de relaxation sélective. L'optimisation en largeur dégage une marge de manœuvre significative pour tenter d'éliminer les interférences radiales (cf. figure 5.33). L'observation attentive de cette proposition montre un cas de DTLM encadré dont les directions d'évolution (pour la résolution des contraintes de montage) dictées par ses voisins, sont antagonistes (DTLM 2 et 3). Le processus s'acquitte pourtant fort bien de cet écueil et génère une solution réalisable en quelques itérations. La première proposition et la solution optimisée sont représentées sur la figure 5.33 (tronçons d'arbre uniquement).

La *phase d'optimisation* en volume consécutive conduit à une solution optimale dont le score

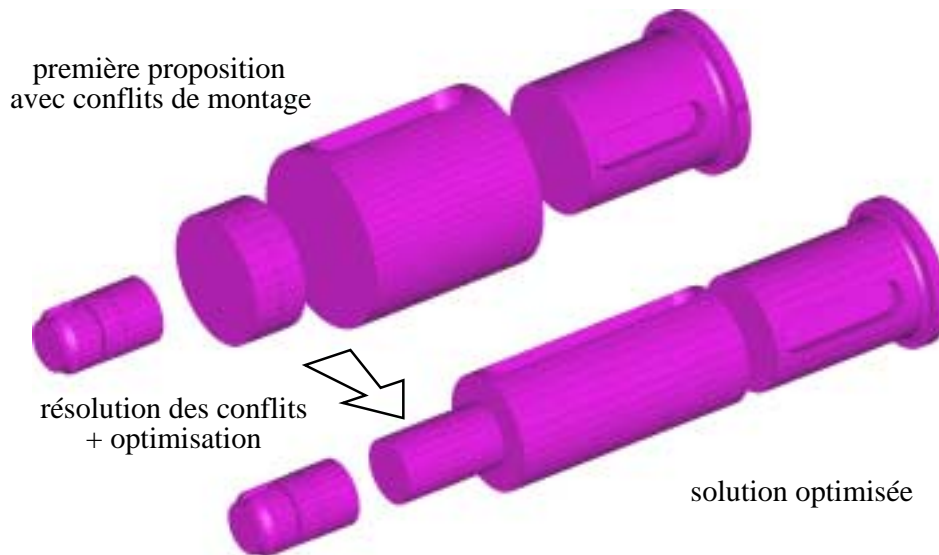


FIG. 5.33 – Première proposition et solution après optimisation.

est situé à moins de 2% de l'optimum global trouvé par la méthode énumérative ($73,54 \text{ cm}^3$ contre $72,24 \text{ cm}^3$), ce qui la place en 3^{me} position. Cette fois encore, les temps d'obtention de

	DTLM 1	DTLM 2	DTLM 3	DTLM 4	DTLM 1	DTLM 2	DTLM 3	DTLM 4
bg_i	63	99,5	125,4	192,17	63	99,5	125,4	191,67
bd_i	84,5	124,5	184,6	233,82	84,5	124,5	184,6	234,32
dag_i	12	15	27	28	12	15	27	27
dad_i	15	15	27	34,7	15	15	27	33,7
f	73,54				72,24			
	relaxation sélective (obj. homogène)				méthode énumérative			

TAB. 5.5 – Solution obtenue par optimisation collaborative.

la solution sont faibles (environ 2 secondes). Ce n'est pas tant l'écart de temps entre les deux méthodes qui doit être relevé, mais plutôt les valeurs relativement faibles des temps d'obtention de solutions. Il est délicat de comparer tous les temps d'obtention deux à deux pour des raisons d'implémentation informatique que nous expliquons en §5.4.4.4. La solution finale est représentée sur la figure 5.34.

5.4.4.4 Remarques.

Bien que le déroulement de la synthèse coordonnée par relaxation sélective soit continu, nous avons donné la possibilité au concepteur de visualiser rétrospectivement l'historique d'évolution des DTLM au cours du processus. Cet assistance, qui n'a d'intérêt que dans le cadre de la synthèse par algorithme, vise plusieurs objectifs :

- étudier l'application des règles d'évolution algorithmique au dimensionnement des DTLM,
- déceler les éventuels points de blocage du processus,
- donner des éléments d'information sur les conflits afin d'aider au choix des DTLM.

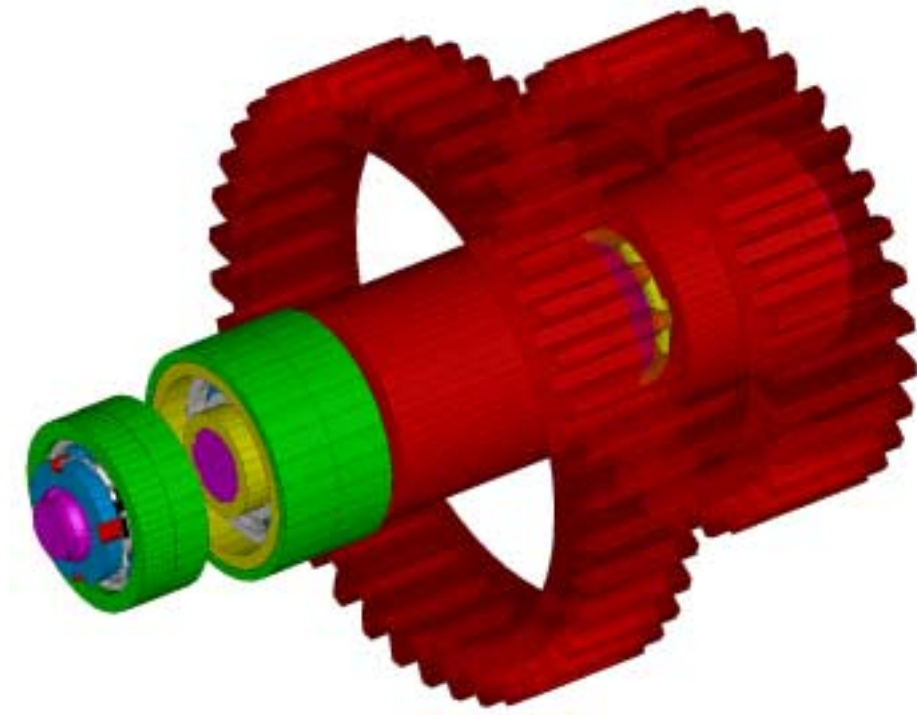


FIG. 5.34 – Représentation complète de la solution proposée au problème sous-contraint.



FIG. 5.35 – Visualisation d'une étape du processus de synthèse coordonnée par algorithme.

Nous avons ciblé la représentation des DTLM sur les tronçons d'arbre car ils sont révélateurs des conflits de montage. La figure 5.35 est un exemple de visualisation d'une étape de l'historique de synthèse coordonnée par algorithme. Les boutons de cette visionneuse permettent de sélectionner l'affichage du résultat de n'importe quelle itération.

La stricte étude comparative des résultats ne présente que peu d'intérêt. Tout d'abord parce que les temps de calcul obtenus sont indissociables de la puissance de calcul des ordinateurs utilisés. Ces valeurs sont donc vouées à une obsolescence rapide! Ensuite parce que malgré les précautions prises lors des essais (utilisation d'une machine identique, pas de charge annexe du processeur...), les méthodes de synthèse par énumération et par algorithme dépendent trop fortement des méthodes de programmation des modules de calcul pour pouvoir établir un comparatif sur les charges de calcul pures. C'est notamment le cas de la rédaction des fichiers résultats qui alourdit considérablement les temps d'exécution et pénalise ainsi les méthodes faisant massivement appel aux itérations. Nous pouvons cependant relever les caractéristiques significatives de ces méthodes.

Afin d'écourter le temps d'exécution de la méthode énumérative, nous avons employé la méthode suivante. Lorsqu'un module de calcul de DTLM est lancé, il génère une liste de DTLM solutions du CdC liaison courant. La méthode énumérative récupère l'ensemble des solutions de chacun des DTLM et tente toutes les combinaisons en y intégrant un critère d'optimisation. Cette démarche présente l'avantage de ne pas avoir à relancer les modules de calcul, mais elle n'est possible que si les modules de calcul sont aptes à fournir l'intégralité des DTLM solutions pour un CdC liaison donné! En d'autres termes, s'ils sont eux-mêmes basés sur une démarche énumérative. Ceci explique la performance de la méthode énumérative, surtout pour les grands nombres de combinaisons à tester, puisqu'on ne lance qu'une fois les modules de calcul.

La méthode de relaxation fait appel, à chaque itération, aux modules de calcul puisqu'elle modifie les CdC liaisons. Elle est bien entendu pénalisée par rapport à la méthode énumérative en ce sens qu'elle multiplie le temps de calcul par le nombre d'itérations. Ceci est principalement vrai pour les faibles nombres de combinaisons de DTLM. Par contre le nombre d'itérations tend à se stabiliser lorsque l'on augmente le nombre de combinaisons (cf. exemple du problème sous-contraint). On conçoit donc aisément que l'utilisation de modules de calculs basés sur une méthode énumérative ne soit guère favorable à la méthode algorithmique (même si au cours de l'évolution, le domaine des solutions se réduit progressivement).

L'utilisation d'un critère homogène (critère global et critères d'optimisation locaux identiques) a montré de très bons résultats (comparaison faite avec la méthode énumérative). Elle sied mieux à notre architecture algorithmique que l'utilisation d'un objectif multicritère. Nous pensons cependant qu'il serait possible d'obtenir de meilleurs résultats avec un objectif global multicritère à condition de modifier la phase d'optimisation de l'algorithme de relaxation sélective comme nous l'avons évoqué précédemment.

5.5 Conclusions.

Nous avons présenté dans ce chapitre le processus de synthèse coordonnée. Nous avons relevé le besoin d'une approche propre à la synthèse des DTLM. Cette approche tient compte des spécificités de notre démarche de conception, telles que l'indépendance vis à vis de la nature des DTLM ou de leurs méthodes de dimensionnement. Nous avons testé plusieurs méthodes algorithmiques et en avons évalué les résultats grâce à la méthode énumérative que nous avons présentée au chapitre précédent.

Nous avons constaté que, quelle que soit l'approche algorithmique employée, notre logiciel

fournissait rarement la solution optimale. En revanche l'approche énumérative s'avère être une méthode infaillible dans ce domaine, mais nous savons que l'emploi de cette seule méthode n'est pas envisageable à cause des temps de calcul dissuasifs qui en résultent (n'oublions pas que les exemples présentés ci-avant n'intégraient qu'un nombre limité de DTLM). Il nous semble donc intéressant d'envisager une approche mixte. Une méthode algorithmique, dans un premier temps, permet de valider rapidement la démarche du concepteur. Nous pouvons ensuite, éventuellement, affiner la solution en utilisant la méthode énumérative. Cette dernière sera d'autant plus rapide que le domaine des solutions aura été restreint. Le gain obtenu, en terme d'objectif, est alors de l'ordre de quelques pourcents mais le temps de calcul économisé peut s'avérer très important (notamment dans le cas des contacts obliques). L'utilisation du temps de calcul par une approche algorithmique peut se modéliser par un diagramme de Pareto tel que représenté sur la figure 5.36. Dans le cadre d'une approche mixte, il serait intéressant de conserver la première partie du processus.

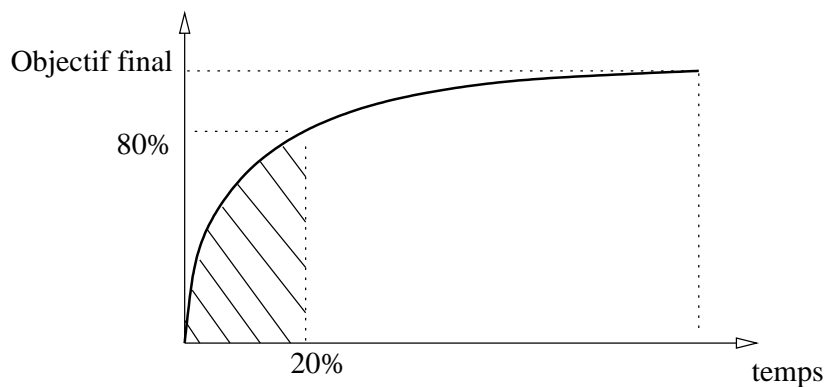


FIG. 5.36 – Diagramme de Pareto représentant la relation entre l'investissement en temps et la qualité de la solution obtenue.

En résumé, la démarche algorithmique permet d'obtenir une solution finale proche de la solution optimale dans le cas d'une liaison pivot décomposable. Elle ne nécessite pas de connaître préalablement les types de DTLM mis en jeu et autorise un fonctionnement de type «boite noire».

Le chapitre suivant traite de l'aspect Génie Logiciel du projet *Sicam2*. Nous présentons succinctement les outils informatiques employés, les raisons de leur choix et leur mise en œuvre. Plus fondamentalement, c'est l'adéquation de ce type d'outils avec des projets tels que *Sicam2* qui est discuté.

Chapitre 6

Implémentation informatique.

Résumé

LA conception d'un logiciel, et à plus forte raison d'un système informatique, n'est pas chose facile. Ce chapitre a pour objectif de présenter les grandes lignes de la réalisation du système *Sicam2*. Après une présentation synthétique des spécificités du Génie Logiciel, nous abordons l'aspect pratique de la réalisation avec l'étude de l'architecture logicielle. Cette architecture est étudiée sous l'angle des flux d'informations internes au logiciel, puis des communications hommes/machines. Ce dernier point amène une réflexion sur l'aspect ergonomique et sur l'utilisation du logiciel qui sont traités dans la dernière partie de ce chapitre. Nous terminons par quelques commentaires relatifs au logiciel¹.

Contenu du chapitre

6.1	Introduction.	172
6.2	Architecture logicielle.	172
6.2.1	Choix des langages de programmation.	173
6.2.2	Liaison des modules.	174
6.2.3	L'interface homme/machine.	177
6.2.3.1	Les menus de renseignement du système.	178
6.2.3.2	Visualisation graphique 3D des DTLM.	178
6.3	Interface graphique et ergonomie du système.	184
6.4	Commentaires sur l'interface proposée.	185

1. Dans un souci de clarté, ce chapitre comporte volontairement quelques rappels d'éléments (tels que des illustrations) présentés précédemment.

6.1 Introduction.

Au même titre que tout produit manufacturé complexe, le logiciel ne peut prétendre à une qualité totale [GMSB96]. On aimerait cependant disposer d'indicateurs de qualité, mais à défaut, on doit souvent se contenter du niveau de satisfaction des utilisateurs. Ceux-là-mêmes se sont en majorité résignés à supporter les bogues et autres désagréments d'utilisation qui ne sont pourtant pas une fatalité. La non-conformité d'un logiciel au cahier des charges initial est un tout autre problème et doit être considérée comme intolérable, surtout lorsqu'elle a des conséquences catastrophiques en termes économiques, ou pire, de vies humaines [Jau90]. M. Barthe relève un décalage entre l'efficacité d'utilisation et les fonctionnalités du logiciel. Il avance quelques causes de la sous-exploitation des capacités du logiciel telles que des fonctions sans intérêt, une mise en œuvre trop compliquée des fonctions ou bien un temps d'apprentissage trop long [Bar95]. Les conséquences les plus connues sont une démotivation des utilisateurs ainsi qu'une hausse des coûts cachés, et vraisemblablement une désaffection pour l'informatique en général. Une partie de ces problèmes est à porter au crédit du manque d'ergonomie, sujet que nous traitons dans le paragraphe 6.3.

Le logiciel, vu comme un produit industriel, est soumis à un processus de production engageant généralement de nombreuses ressources humaines, mais sa production est unitaire (la diffusion en nombre ne requiert qu'une simple opération de copie, sans aucune logistique lourde). Il se caractérise cependant par une tendance marquée au dépassement de délai dans les phases de conception (les exemples sont légion, surtout du côté des logiciels « millésimés »). La méthode de développement doit s'adapter au contexte de création et tenir compte du fait qu'un logiciel reste un produit *invisible* durant son développement. Une solution consiste à procéder à un maquetage du logiciel (ou prototypage). On distingue deux catégories de maquettes : les maquettes exploratoires — l'utilisateur précise ses attentes à l'aune de ce que lui présente le concepteur — et la maquette expérimentale — où le concepteur teste des solutions concurrentes —. Dans les deux cas, la spécification initiale des caractéristiques est essentielle à la définition du produit. La création d'un logiciel scientifique expérimental *ne respecte pas la même logique que la production d'un logiciel commercial*. Nous nous sommes cependant inspirés de certaines pratiques du génie logiciel afin de rationaliser nos efforts. L'une d'entre-elles est la modularité du logiciel, qui facilite sa maintenance, la réutilisation de ses composants et surtout la conception concourante. Elle nécessite des choix importants d'architecture, de langages, que nous détaillons dans les sections suivantes.

6.2 Architecture logicielle.

L'architecture des logiciels scientifiques et expérimentaux nécessite de pouvoir mettre au point et modifier aisément le code. Les systèmes informatiques, particulièrement dans le domaine de la CAO, ont une durée de vie beaucoup plus courte que les produits qu'ils ont aidés à concevoir ou gérer [Loy91]. Il importe donc que le logiciel soit évolutif, et portable sur de nouveaux matériels. Cela se traduira en termes de choix d'architecture logicielle et de langages. D'autres paramètres environnants sont à prendre en compte lors de la conception d'un logiciel. Considérons les acteurs du projet. La conception d'un logiciel de CAO dans le cadre d'un laboratoire de Génie Mécanique fait appel à des mécaniciens (!) qui ne sont pas toujours au faite des techniques du Génie Logiciel. Nous avons donc composé avec ces compétences. Cela se traduit en partie par des choix de langages que nous justifions en suivant.

6.2.1 Choix des langages de programmation.

Le choix du langage de programmation est un élément essentiel de la réussite d'un projet informatique. Le langage doit être bien défini syntaxiquement et sémantiquement et adapté au domaine d'application [GMSB96].

Le choix du langage C pour l'écriture des modules de calcul s'est imposé par sa pratique très répandue, ainsi que son aptitude au calcul numérique. Il n'est plus nécessaire de présenter le langage C [KR84], né dans les années 70, intronisé depuis par une norme ANSI, qui bénéficie actuellement d'une imposante collection de routines, bibliothèques de fonctions ou outils de développement, souvent gratuits grâce à l'effort de la communauté libre des informaticiens. Ce langage de programmation constitue une référence (bon nombre de systèmes d'exploitation et de langages actuels sont écrits en C!). Le C facilite également l'écriture modulaire de logiciels. Les algorithmes de calculs ont été étudiés et encodés à des dates différentes, ce qui nous a obligé à en faire des modules, dans un premier temps indépendant, que nous avons pu tester et valider individuellement.

Le langage C++ [Bro01] a été imposé par l'utilisation des A.P.I.² d'Open I-DEAS[®]. L'utilisation d'un langage objet était rendu obligatoire par l'architecture CORBA qu'utilise OPEN I-DEAS[®]. Ce choix nous a également permis de nous orienter vers une librairie mathématique orientée objet, OptSolve+, pour la phase de synthèse coordonnée [Cor02].

Afin de faire le lien entre tous ces éléments, et compte-tenu de l'hétérogénéité des langages employés, nous nous sommes tournés vers un langage script. Ce type de langage, présenté comme la révolution informatique du XXI^{ème} siècle, est souvent assimilé au ciment qui lie les briques d'une construction. Quelles sont les raisons qui nous ont poussé à choisir un langage script? Les principales que nous avons retenues sont :

- l'application doit intégrer un ensemble de composants ou d'applications existantes ;
- l'application doit intégrer une GUI³ ;
- l'application doit évoluer rapidement et facilement.

Ajoutons que ces langages sont dotés d'une portabilité hors du commun, qu'ils sont très doués pour le traitement de chaînes de caractères et qu'ils manipulent une grande variété de composants différents [SCR98]. Quatre langages scripts se partagent actuellement le plus gros du marché : Perl, Tcl, Python et PHP. Le Tcl⁴, déjà largement employé par ailleurs, a retenu notre attention ([Ous94] et [Ous98]). Sans rentrer dans le détail des petites querelles qui enrichissent les forums, nous avons choisi Tcl pour des raisons (apparentes) de simplicité d'apprentissage, pour l'extension graphique Tk⁵ et sa gratuité. Tcl est né en 1988 du besoin d'un langage de commande a raccordé à des logiciels de ... CAO ! Il s'est vu adjoindre une couche graphique en 89 nommée Tk. Ils sont tous les deux codés en C et donc facilement utilisables conjointement (Tcl peut utiliser des fonctions C, ou bien un programme C peut utiliser des bibliothèques de fonctions Tcl via un interpréteur). Tcl est un langage interprété, il n'a donc pas besoin d'être recompilé à chaque modification, ce qui le rend idéal pour l'élaboration de prototype de logiciel. La section suivante traite de l'utilisation de ce langage pour intégrer les divers composants logiciels du système.

2. Application Programming Interface.

3. Graphical User Interface, soit une interface graphique utilisateur.

4. Tool Command Language.

5. Tool Kit, l'interface graphique associée à Tcl.

6.2.2 Liaison des modules.

Nous avons adopté une structure modulaire pour la création du système. Bien que cette méthode présente de nombreux avantages, elle n'est pas naturelle pour un ordinateur qui doit partager sa mémoire entre les données et les divers programmes. Il convient donc de bien opérer le partage des opérations et c'est à l'aide d'un découpage fonctionnel du système et une approche concourante de la construction du système que nous avons architecturé le projet *Sicam 2*. Ce projet est un travail de longue haleine qui ne peut s'inscrire raisonnablement que dans la durée. La version actuelle du logiciel comporte plus de trois mille lignes de codes réparties dans une quarantaine de fichiers, et deux-cent-vingt-quatre modèles volumiques CAO. L'étalement des travaux de développement logiciel a conduit à leur confinement en modules autonomes. Cette modularité a permis la validation partielle, dans un système non finalisé, et une maintenance facilitée du système (et son évolution puisque c'est le principal souci du logiciel scientifique); rien n'est plus simple que d'isoler un module pour l'améliorer et de le réinsérer ensuite dans le système. Afin de mener à bien cette modularité, nous avons dû penser aux principaux liens qui devaient exister entre les différents modules: les échanges de données et les mécanismes d'activation. Concernant l'échange de données entre les divers modules, nous avons mis en place un double mécanisme de fichiers de données et de variables globales. L'emploi des fichiers de données en entrée et sortie de modules nous a permis de tester indépendamment les modules en bénéficiant d'une lisibilité parfaite. Ces fichiers sont rédigés en ASCII et font figurer, pour chaque donnée, une ligne descriptive de ladite donnée. Leur nom est composé selon une règle préétablie qui les distingue selon le nom du projet, le type de module concerné ainsi que le type de données contenues. Ces manipulations d'informations textuelles ont été facilitées par l'emploi d'un langage script. Le passage d'arguments par valeur entre module est une des possibilités offertes par le Tcl. Les variables sont déclarées dans des «namespace» locaux ou globaux et sont accessibles par tous les modules qui ont été lancés dans le même interpréteur (ce qui est le cas des modules annexes de *Sicam 2*). L'accès direct, en mémoire, aux variables mises à jour par les menus du module principal est accéléré par rapport à la lecture des fichiers cahiers des charges ou résultats. Cette technique a pour but de réduire les temps d'exécution.

L'ensemble du transit des données de conception est représenté sur la figure 6.1. Celle-ci mérite quelques commentaires. La première source d'informations est constituée par le *cahier des charges principal*. Il rassemble toutes les informations du niveau de définition supérieur, le niveau T3 (cf. §1.6.2.3), dont les positions des liaisons, les dentures d'engrenage... Le module principal produit des cahiers des charges à destination des modules de calcul de chaque liaison. Pour une simplification de la représentation, nous avons englobé dans les modules de calcul les différents algorithmes de dimensionnement propres à chaque DTLM. A titre d'exemple, le module calcul arbre/moyeu comprend les programmes de calcul des liaisons par clavette, par cannelures et par freinage. Chaque module de calcul est donc en relation avec un fichier *cahier des charges*, un fichier *résultats* ainsi qu'un ou plusieurs fichiers *base de données*. C'est dans ces bases de données que l'algorithme de dimensionnement recherche les caractéristiques des composants standards, les propriétés matériaux, etc. On peut constater que le module calcul de la liaison pivot engendre deux fichiers résultats dont un, le fichier nommé résultats statiques, contient les efforts dans les liaisons. Ces données sont *indispensables* au calcul de toutes les liaisons car elles déterminent les sollicitations dans l'arbre. Leur estimation est réalisée dans le module de calcul pivot une fois la décomposition de la liaison pivot connue. Si la liaison pivot est réalisée par une rotule et une linéaire annulaire, les éléments de réduction des torseurs sont alors implicitement connus, et l'on pourrait éventuellement passer au dimensionnement des liaisons arbre/moyeu sans même connaître les DTLM réalisant la liaison pivot. Par contre, l'utilisation de roulements à

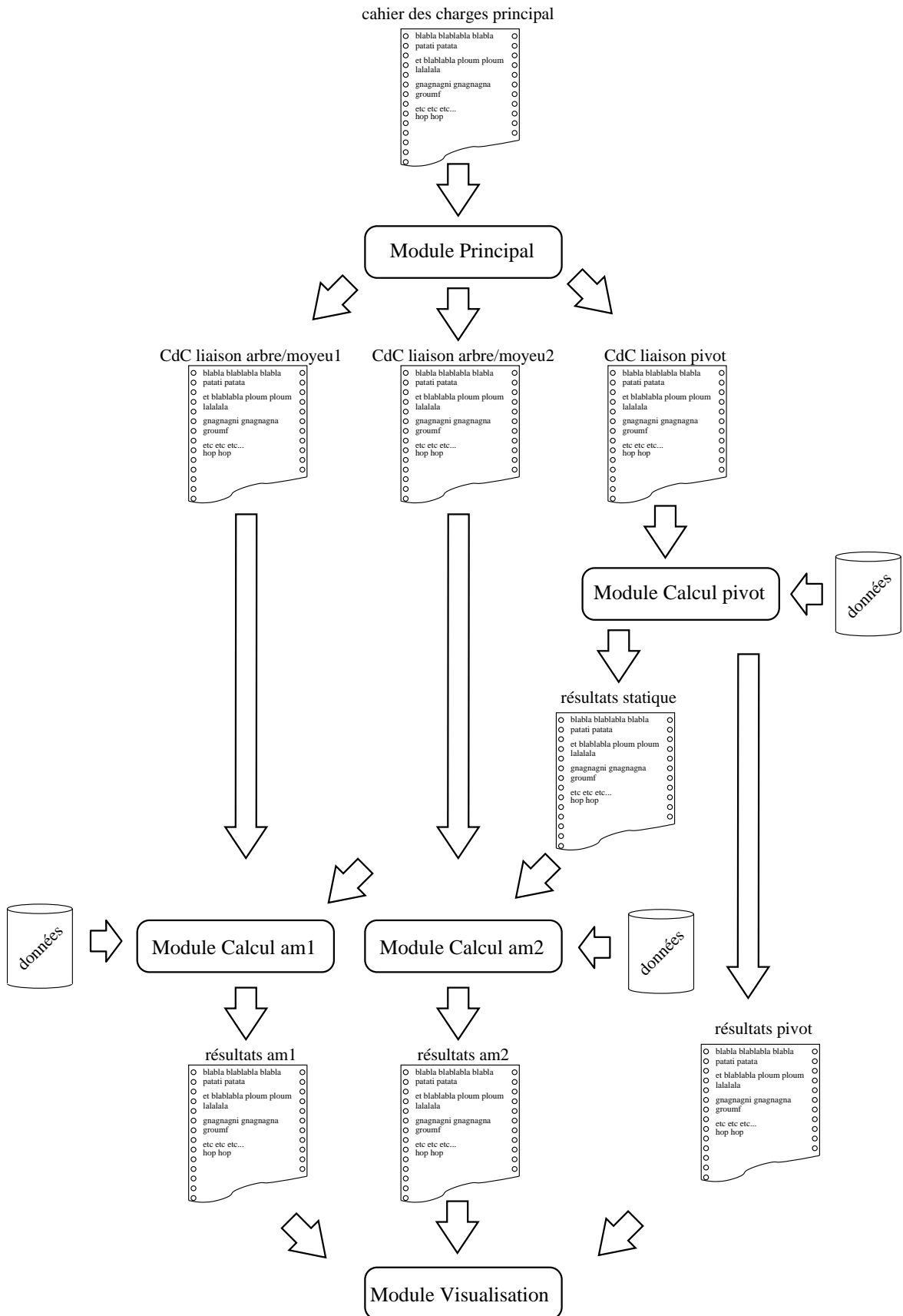


FIG. 6.1 – Datagramme du système.

contact oblique ne permet pas de déterminer les torseurs d'efforts tant qu'ils n'ont pas été choisis. Afin d'harmoniser le déroulement du processus de conception, le calcul des DTLM pivots et la détermination des efforts sont couplés dans un même module fonctionnel. Il est donc impératif de commencer la détermination des DTLM par ceux de la liaison pivot. A l'issue des calculs de DTLM, le système dispose de fichiers résultats comportant tous les éléments dimensionnels nécessaires aux modèles 3D. Le module visualisation se charge alors de transmettre ces éléments au modeleur volumique.

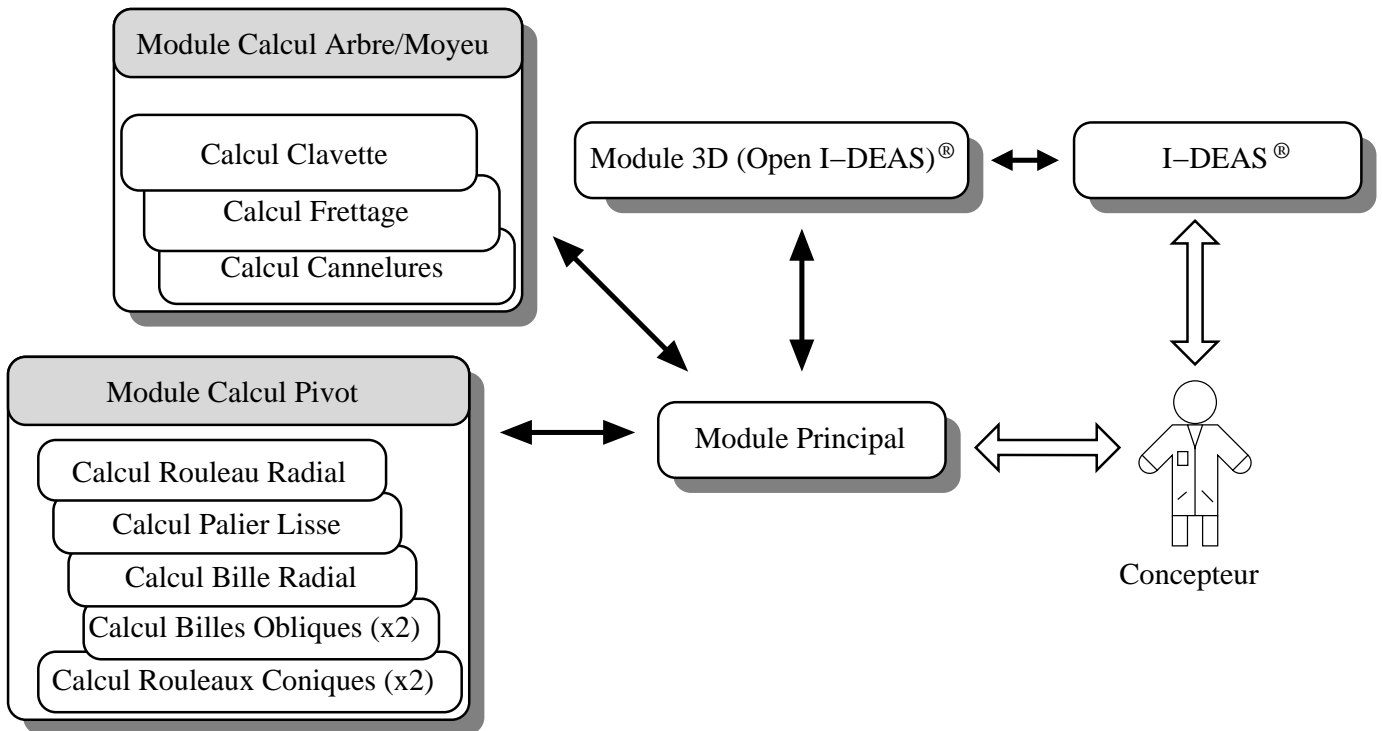


FIG. 6.2 – Actigramme du système.

L'activation des différents modules est représentée sur la figure 6.2. Le module principal est au cœur du processus de conception. Il lance les modules de calcul, le module de coordination ainsi que le module de visualisation à la demande de l'utilisateur. L'utilisateur n'a d'interactions avec le système que par le biais du module principal (donc les menus) et I-DEAS®. Le module principal est toujours actif, y compris lors du lancement des modules annexes et conserve ainsi la main en cas de problèmes rencontrés sur ceux-ci. Sans aller trop en avant dans la description d'une séquence de conception avec *Sicam*, ceci faisant l'objet des chapitres 3, 4 et 5, nous pouvons en rappeler les grandes lignes suivantes. Une certaine cohérence dans la synchronisation des actions est impérative. Le concepteur doit commencer par remplir le cahier des charges principal, puis s'intéresser à la définition de la liaison pivot qui conditionne les calculs de statiques. Alors il peut, à loisir, définir le reste des liaisons dans l'ordre qu'il désire. L'étape de coordination n'intervient qu'en dernier lieu pour trouver une solution globale (c'est-à-dire impliquant tous les DTLM). Pour bien comprendre l'intérêt d'une définition en deux temps des DTLM, nous renvoyons le lecteur aux chapitres précédents.

6.2.3 L'interface homme/machine.

Les multiples tentatives menées dans le but d'élaborer un système de conception entièrement automatique ont échoué. Ne nous leurrions pas, notre connaissance actuelle du processus cognitif de conception ne nous permet pas de telles prétentions! Plus modestement, un système de conception assistant efficacement le concepteur est un objectif louable et sage qui nous semble réalisable. *Sicam* a été conçu dans cet esprit. Aussi fait-il appel au concepteur à chaque fois qu'une étape de choix survient dans sa construction d'une solution. Deux raisons à cela :

- Le logiciel n'entend pas se substituer au concepteur dans les choix où son expertise peut (doit) faire la différence. La mécanique regorge de nombreux cas de figure où les constructions, ayant prouvé leur efficacité, n'étaient pas forcément les plus «académiques». En règle générale, le concepteur se raccroche à des solutions éprouvées et connues de tous, d'une part pour se faciliter la tâche, et d'autre part pour s'assurer de leur compréhension (qui vaudra agrément). L'originalité n'est alors pas de mise, et l'on peut passer à côté d'une architecture qui, à défaut d'être révolutionnaire, se révélerait d'une efficacité supérieure. en rendant la liberté d'investigation au concepteur, on fait la part belle à la *créativité intelligente* ⁶.
- Le système informatique doit, et c'est notre intime conviction, se doter de vertus pédagogiques. Les systèmes actuels sont trop souvent basés sur le modèle boîte-noire et presse-boutons. L'utilisateur développe une certaine méfiance vis-à-vis de ce genre de logiciels qui n'explicitent jamais les situations d'échecs. Ce constat est d'autant plus criant lorsqu'une part des connaissances est intégrée (et donc cachée) dans le code informatique. Les systèmes experts ont longtemps fait partie de cette catégorie. Ils sont en train d'évoluer vers plus de transparence, notamment en rendant accessible le déroulement du processus, les règles déclenchées, etc. *Sicam* est un outil de construction du savoir. Le concepteur l'utilise comme «calculatrice» sophistiquée afin de dimensionner les DTLM qu'il intègre au mécanisme. Quel élément a penché en faveur de tel ou tel DTLM? Est-ce la meilleure solution? Ce DTLM constituerait-il une bonne solution dans un autre contexte ou bien est-il propre à cette configuration de mécanismes? Quel concepteur ne s'est donc jamais posé ce genre de questions? En ayant la possibilité de tester rapidement différentes combinaisons de DTLM dans un contexte de mécanisme donné, *Sicam* contribue à enrichir la connaissance du concepteur. Cette démarche est un des principes d'action du groupement d'universités et d'écoles d'ingénieurs PRIMÉCA ⁷ qui a pour objet de promouvoir le développement des filières de formation d'ingénieurs et de maîtres des universités à la CMAO [Tol00]. Notre souhait est que le logiciel, au sens large du terme, soit un élément d'assistance autant que de construction du savoir. La rapidité de l'outil informatique permet d'augmenter considérablement le nombre de cas étudiés et, par conséquent, l'expertise de l'utilisateur.

Puisque l'homme doit dialoguer avec la machine, il est primordial qu'ils se comprennent sans ambiguïté. L'interface homme/machine devra donc être conçue dans le but unique d'échanger les informations entre le système et son opérateur avec une intégrité totale. L'échange bilatéral suppose des mécanismes d'envoi et d'accueil d'informations de la part du système. On distingue deux phases d'échange : la phase de renseignement du système (en amont du dimensionnement du DTLM) et la phase de retour du système (proposition dimensionnelle du DTLM). A chacune

6. Le Petit Robert[RDR02] définit l'intelligence comme la « faculté de connaître, de comprendre ».

7. Pôle de Ressources Informatique pour la MÉCANIQUE.

de ces phases est dévolue un type d'interface: menus boutons (Tk) et graphisme 3D (I-DEAS®). Les paragraphes suivants détaillent les aspects techniques des menus et de la visualisation 3D. L'aspect ergonomique sera traité en §6.3.

6.2.3.1 Les menus de renseignement du système.

Ils sont écrits en Tk et sont appelés par le module principal dès lors que l'opérateur en exprime le souhait. Ils font une large place aux éléments graphiques afin de visualiser rapidement l'information. Les variables modifiées par ces menus sont globales, c'est-à-dire qu'elles sont accessibles depuis n'importe quelle fonction. Il n'est donc pas rare de voir des variables figurer sous différentes formes, comme les positions des liaisons. La modification d'une de ces valeurs est alors immédiatement répercutée sur l'ensemble du système grâce aux capacités de traitement événementiel⁸ du Tk [Wel97]. Cela signifie que Tk intercepte tous les événements de la couche graphique du système d'exploitation —cela va du simple passage du curseur dans une zone prédéfinie à la commande de fermeture d'une fenêtre— et exécute les commandes par défaut associées à ces événements. Nous avons alors la possibilité de reprogrammer ces événements pour, par exemple, commander l'ouverture d'un fichier par un double clic sur son nom, ou bien interdire la fermeture d'une fenêtre par l'utilisateur, comme cela est le cas dans notre système. Bien entendu, ces possibilités ne doivent pas desservir le concepteur, en lui imposant un mode de fonctionnement strict. Elles doivent plutôt être vues comme une occasion de mieux contrôler la réponse du logiciel aux actions de l'opérateur, où parfois de prévenir les erreurs. Ces considérations relèvent du domaine de l'ergonomie, et nous renvoyons le lecteur en §6.3 pour plus de précisions sur l'aspect graphique de ces menus.

6.2.3.2 Visualisation graphique 3D des DTLM.

Depuis quelques années les modeleurs volumiques s'ouvrent à l'utilisateur. Dans des mesures différentes, les éditeurs de logiciel optent pour des codes informatiques en partie ou totalement réutilisables par l'utilisateur. La finalité de ces propositions est de fournir un logiciel adaptable aux besoins de chacun. Les éditeurs de logiciel ont bien compris que la demande, malgré quelques éléments communs tels que le modeleur volumique, divergeaient sur les applications métier. En proposant un logiciel dont les fonctions internes sont accessibles, l'éditeur donne la possibilité à l'utilisateur de l'ajuster à ses besoins spécifiques. C'est par ailleurs l'occasion de prendre plus durablement des parts de marché dans un contexte compétitif et très versatile (cf. §1.7.1) avec un investissement moindre dans le développement. Parmi ces éditeurs, nous trouvons MATRA (OPEN-CASCADE), AUTODESK (AUTOCAD®), PTC (Pro-ENGINEER®) et SDRC (I-DEAS®). Au moment où la décision fût prise de retenir I-DEAS® pour la modélisation 3D des DTLM, OPEN-CASCADE n'était pas encore disponible en sources libres. A l'heure actuelle, compte tenu de la plus grande liberté de conception liée à la distribution du code source, cet environnement retiendrait notre attention. L'utilisation de l'extension OPEN I-DEAS®, nous offre la possibilité d'utiliser la majorité des fonctions internes du logiciel dont celles de l'application modeleur volumique (Master Modeler). Celui-ci propose un ensemble d'A.P.I. donnant accès aux principales fonctions de manipulation des entités assemblages et pièces du logiciel grâce à CORBA.

En quelques mots, CORBA a été créé par l'OMG⁹ dont l'objectif est de promouvoir la théorie et la pratique de la technologie objet pour les applications distribuées [Gro01]. CORBA est un acronyme pour « Common Object Request Broker Architecture » et désigne une structure

8. Event-driven control.

9. Object Management Group.

d'échange d'objets par requêtes universelles. CORBA définit une méthodologie d'intégration d'applications distribuées en leur autorisant une communication bilatérale, quelles que soient leur localisation ou leur implémentation, grâce à un système de requêtes basées sur le modèle client/serveur. On l'aura compris, la technologie objet est au cœur du processus d'échange. Cette méthodologie est intégrée dans la version 7 d'I-DEAS[®] par le produit ORBIX[®], codé en langage C++, sous forme d'un serveur lancé en même temps que le logiciel. La figure 6.3 représente les interfaces et caractéristiques des composants de l'architecture CORBA [Vin97]. On y distingue

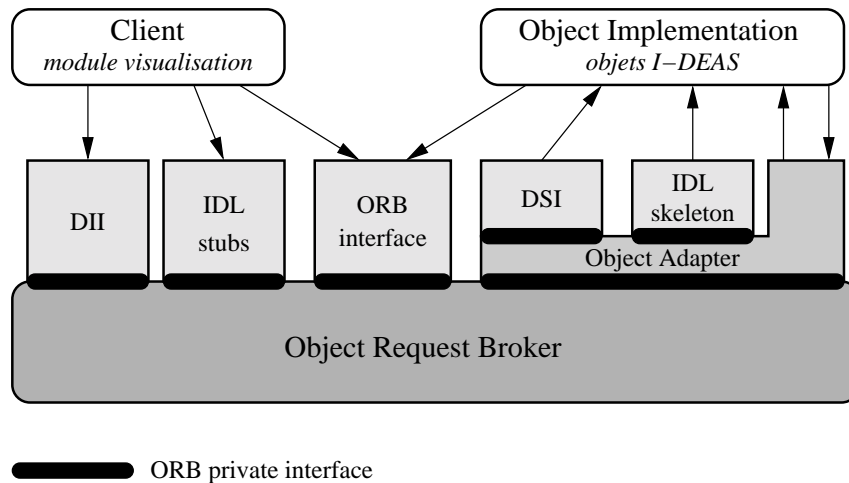


FIG. 6.3 – Architecture simplifiée de CORBA.

le client (notre application par l'entremise du module visualisation), le serveur (l'application OPEN I-DEAS[®]), l'agent d'échange (ORB) et les différentes interfaces. Les «Interface Definition Language» sont compilées en «IDL stubs» et «IDL skeleton» qui forment une sorte de traducteur vers les définitions IDL de CORBA respectivement pour le client et le serveur. Les IDL sont définies de façon tellement rigoureuses que la compatibilité est totale dans les échanges entre «IDL stubs» et «IDL skeleton», même s'ils n'ont pas été écrits dans les mêmes langages. Les «Dynamic Invocation Interface» et «Dynamic Skeleton Interface» permettent à l'utilisateur d'attaquer directement les requêtes au niveau de l'agent d'échange sans passer par les définitions IDL. Les requêtes transitent donc depuis le client vers le serveur par l'intermédiaire de l'agent d'échange. Que sont nos objets I-DEAS[®]? Ce sont toutes les entités manipulables du logiciel (surfaces, pièces, assemblages, dimensions, etc.) dont on trouve une liste exhaustive dans les IDL d'OPEN I-DEAS[®]. Les méthodes associées à ces objets sont les manipulations courantes telles que renommer, changer les attributs, déplacer, etc.

L'interface graphique du logiciel I-DEAS[®] est d'un format relativement classique. Elle se compose de quatre fenêtres (cf. figure 6.4) dont la principale occupe les trois-quarts de l'écran. Elles ont pour fonction l'impression de messages textuels à destination de l'utilisateur, la saisie de données alphanumériques, la saisie de commandes par menu d'icônes et enfin l'affichage graphique des objets modélisés. Nous avons délibérément choisi de ne pas modifier l'interface du logiciel pour des raisons d'ergonomie qui seront exposées dans la section suivante. En conséquence, l'utilisateur ne peut distinguer une version d'I-DEAS[®] classique d'une version couplée à *Sicam 2*. Au fur et à mesure que le concepteur définit les DTLM à l'aide de *Sicam 2*, il peut les modéliser dans I-DEAS[®]. La figure 6.5 représente un de ces *assemblages racines* que nous avons défini au chapitre 3.

L'algorithme que nous avons appliqué (fig. 6.6) implémente des classes d'objets DTLM auxquels nous avons incorporés les méthodes nécessaires à leur traitement. De ce fait, il s'applique

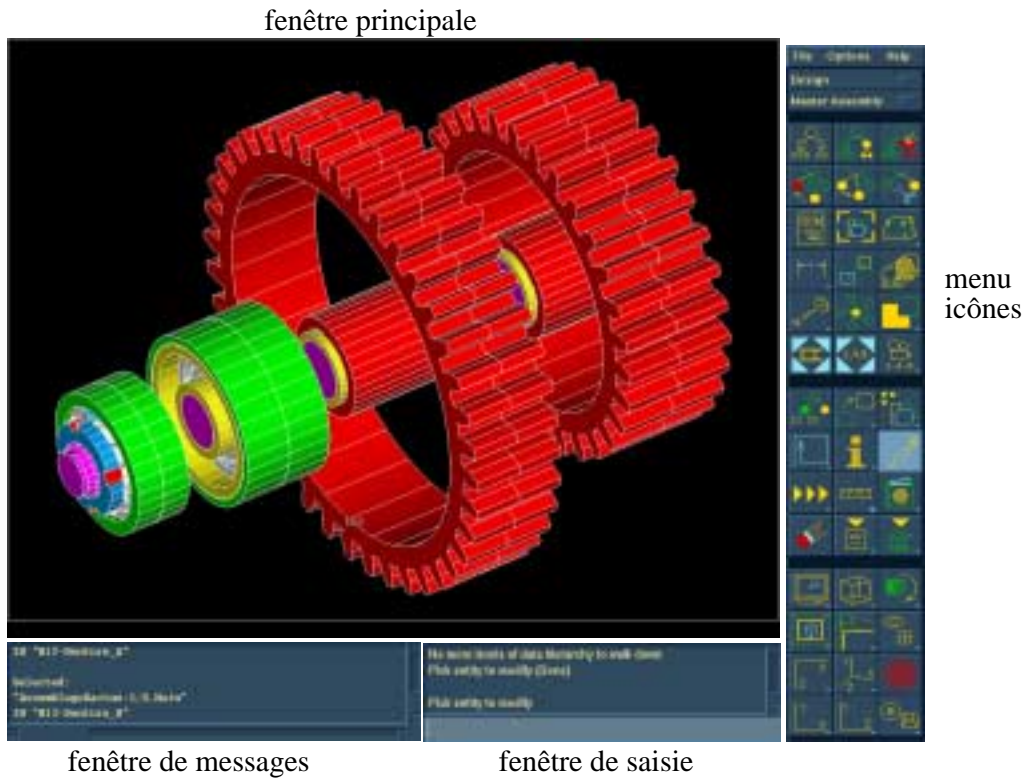


FIG. 6.4 – Interface graphique du logiciel I-DEAS®.

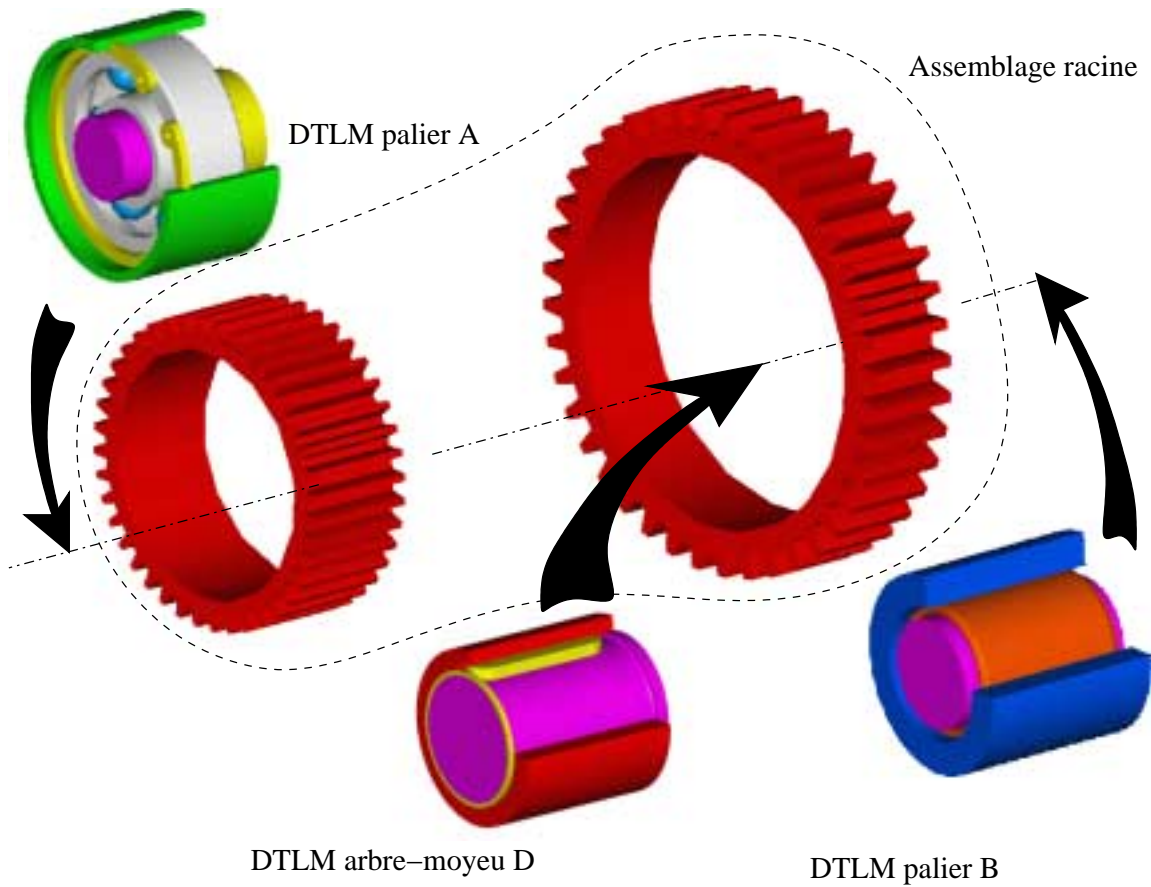


FIG. 6.5 – Composition d'un assemblage racine.

à toutes les liaisons, et donc à tous les DTLM. Il a été encodé en langage C++ puisqu'il doit utiliser les IDL d'OPEN I-DEAS®. L'algorithme présenté est extrêmement simplifié, mais il retrace les principales étapes de la visualisation d'un DTLM. L'algorithme débute tout d'abord

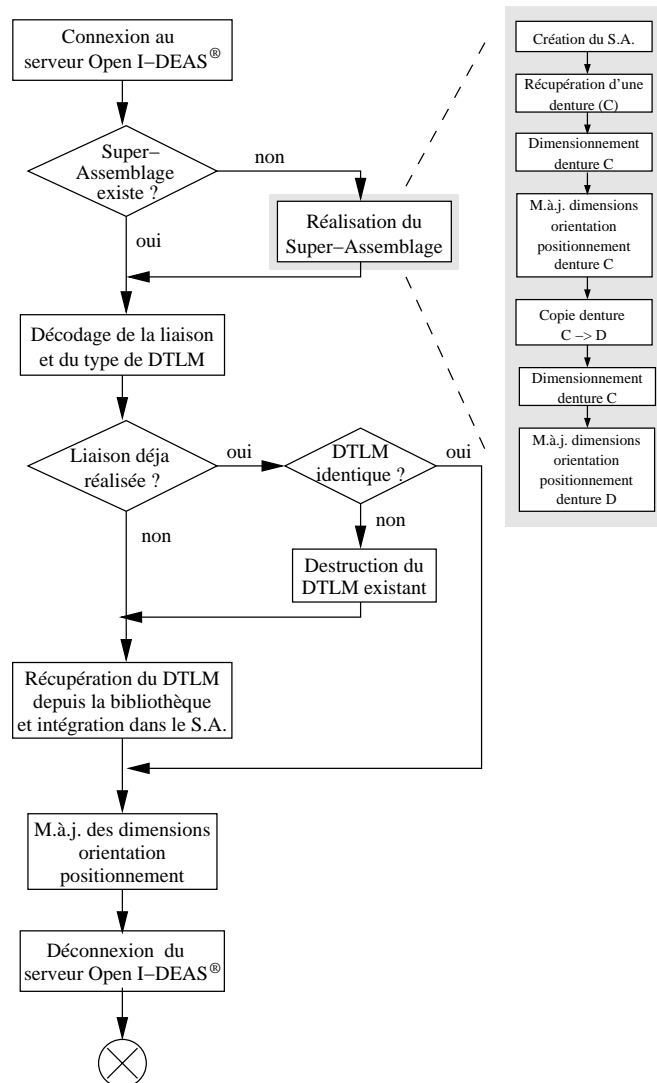


FIG. 6.6 – *Algorithme primaire du module de visualisation.*

par une connexion au serveur OPEN I-DEAS®. Il vérifie la présence de l'assemblage racine dans l'espace de travail et le crée si besoin est, en y intégrant les dentures d'engrenages. Il décode ensuite les arguments reçus du module principal qui lui indiquent quel type de liaison doit être modélisée et à l'aide de quel DTLM. Il vérifie alors que la liaison n'est pas déjà réalisée par un DTLM identique à celui demandé, auquel cas il s'agirait simplement d'un redimensionnement, et procède à l'élimination de l'éventuel prédécesseur. Il reste alors à récupérer le nouveau DTLM en bibliothèque, lire ses paramètres dimensionnels dans le fichier résultat de la liaison concernée, et redimensionner le DTLM. Les modèles volumiques des DTLM sont repérés par un codage de leurs éléments principaux. Ce code unique à chaque DTLM (figure 6.7) autorise l'identification de celui-ci dans toutes les parties du logiciel. L'orientation et le positionnement terminent la mise en place du DTLM dans le mécanisme étudié.

Le paramétrage des modèles 3D des DTLM nous a posé quelques problèmes. Nous avons détecté une lenteur certaine dans le processus de visualisation 3D (entre 1 et 3 minutes suivant

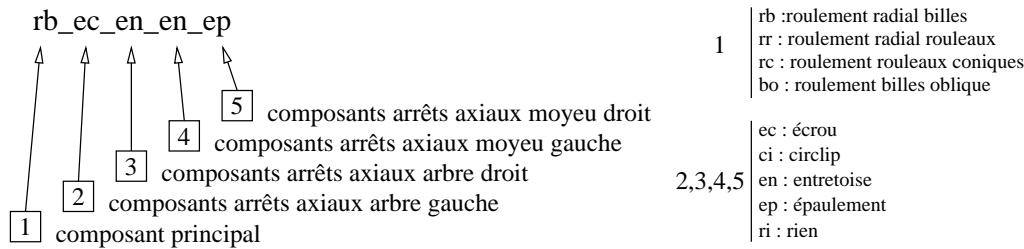


FIG. 6.7 – Exemple de codage de DTLM de liaison rotule.

le cheminement algorithmique et la complexité des modèles 3D). Cette lenteur, quasi-générale dans le module OPEN I-DEAS[®], nous semblait handicapante d'un point de vue ergonomique, car de nature à générer de l'impatience chez l'opérateur. Bien que les temps d'exécution soient restés infiniment moindres que dans le cas d'un traitement manuel, nous avons mené des investigations dans le but de découvrir quel était le maillon faible de la chaîne (i.e. l'opération la plus lente) et le motif de cette lenteur. Compte tenu de la répartition des temps d'exécution du module de visualisation 3D, nous nous sommes focalisés sur la procédure de modification des dimensions et de mise à jour¹⁰ des DTLM, ces deux actions formant le redimensionnement (figure 6.8). Les

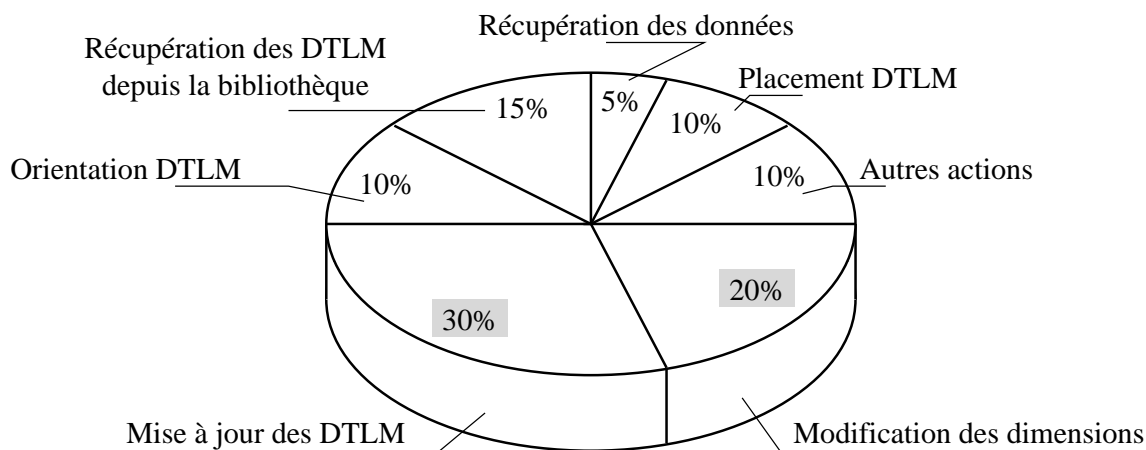


FIG. 6.8 – Répartition moyenne des temps d'actions du module de visualisation 3D.

pourcentages affichés reflètent une moyenne obtenue pour la visualisation d'un DTLM dans le cas d'un S.A. vierge (pas de liaison précédemment définie, donc pas de DTLM à effacer). Ils sont globalement indépendants de la complexité du DTLM. La plupart des opérations présentent des temps de traitement incompressibles, à l'image de la récupération des DTLM en bibliothèque (sous réserve d'une modification intégrale de la conception des DTLM pour diminuer le nombre de pièces dès l'assemblage). L'action que nous avons menée concerne le paramétrage des DTLM. Comme il a été précisé au chapitre précédent, le paramétrage du modèle 3D doit être concordant avec le paramétrage mathématique du problème d'optimisation. Ceci n'est pas toujours réalisable, car les méthodes de construction du modèle 3D peuvent répondre à une logique différente. Pour des raisons de robustesse du modèle, on peut préférer, par exemple, une construction filaire — donc à bases de rayons — qui diffère du modèle mathématique utilisant les diamètres. Il convient donc de mettre en place des *règles de conversion* qui, à partir des données mathématiques du problème, produiront toutes les dimensions du modèle 3D paramétré. Prenons l'exemple de la

10. La mise à jour des composants, sous I-DEAS[®], est l'opération qui produit le nouveau composant en résolvant, grâce au solveur interne du logiciel, les contraintes géométriques et dimensionnelles affectant celui-ci.

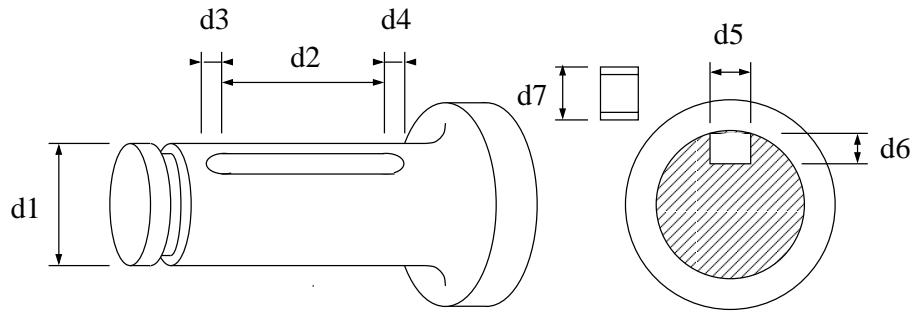


FIG. 6.9 – Paramétrage simplifié du modèle 3D de l'arbre claveté épaulé.

liaison arbre/moyeu par clavette dont le paramétrage du modèle 3D est donné figure 6.9. Les deux cas extrêmes sont les suivants :

1. Le module de calcul détermine toutes les dimensions nécessaires au paramétrage du modèle 3D. Le module de visualisation doit alors transférer toutes ces valeurs à OPEN I-DEAS® puis modifier toutes les dimensions du modèle, et enfin mettre à jour le modèle (fig. 6.10).

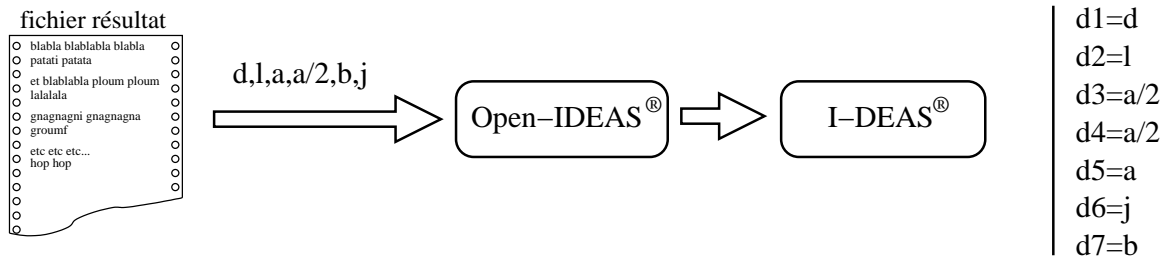


FIG. 6.10 – Transfert maximal des paramètres dimensionnels.

2. Le module de calcul détermine uniquement les paramètres de conception du DTLM. Ils sont transmis à OPEN I-DEAS® qui, lorsque l'on procède à la modification des dimensions, propage les dimensions dérivées des valeurs de base (fig. 6.11).

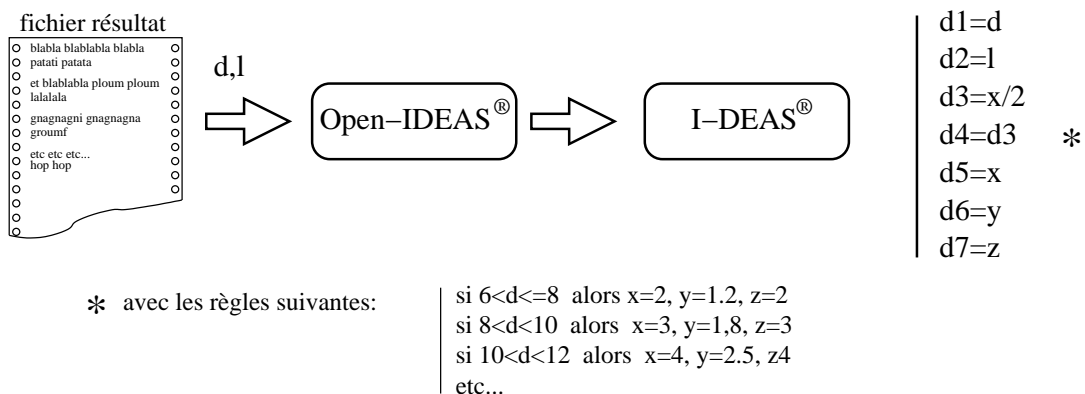


FIG. 6.11 – Transfert minimal des paramètres dimensionnels.

Quelles sont les conséquences prévisibles du choix de ces deux configurations ? Il semble logique que dans le premier cas, les temps de transfert de données soient plus importants. Nous faisons

l'hypothèse que, toutes choses égales par ailleurs, les calculs numériques sont plus rapides lorsqu'ils sont effectués avec le langage C compilé (le module de calcul), plutôt qu'en script interprété (I-DEAS®). Selon cette hypothèse, nous devrions gagner du temps de calcul dans le premier cas. Reste à savoir si le gain de temps est suffisant en comparaison du surplus engagé pour le transfert des données. Nous avons effectué une campagne d'essais pour mesurer les gains de temps éventuels entre chaque solution extrême. Pour cela, nous avons réalisé deux modélisations sur deux DTLM (de complexité différente), puis nous avons mesuré les temps d'exécutions complets des boucles de visualisation. Les résultats sont les suivants. Le gain de temps est faible, mais toujours en faveur du transfert minimal des données. La différence de gain entre un modèle 3D complexe (faisant intervenir de nombreuses pièces et dimensions) et un modèle simple est négligeable. Les explications à ces résultats semblent être les suivantes :

- le temps de transfert de données via OPEN I-DEAS® est proportionnel au nombre de valeurs à transférer ;
- toutes les dimensions du modèle 3D doivent être modifiées, que ce soit par une affectation simple (cas 1) ou par l'application de règles internes (cas 2). On ne peut donc espérer d'amélioration de ce côté.

En résumé, il faudrait engager de gros efforts de réflexion pour obtenir de faibles gains de temps. De plus, le logiciel est un produit commercial dont l'analyse de la structure nous est totalement impossible. Les mécanismes de modification de dimensions et de gestion de mémoire pourraient nous apporter des informations complémentaires, utiles dans notre démarche. Enfin, l'intégration de toutes les règles de dimensionnement alourdit le modèle 3D et pose dans certains cas un problème de stabilité du système. L'utilisation *intégrale* d'un modèle objet couplé à un modeleur 3D « libre » supprimerait ces problèmes de transfert et de conversion.

En conclusion, nous constatons que, de la structure d'un système informatique (logiciel + matériel), dépendent fortement ses performances. L'utilisation de codes libres où l'écriture complète du logiciel nous libérerait d'un certain nombre de contraintes mais au prix d'un engagement informatique qui n'est vraisemblablement pas à la portée de spécialistes de la mécanique.

6.3 Interface graphique et ergonomie du système.

Dans le milieu des années 80, est apparu un accessoire informatique qui est rapidement devenu indispensable : la souris. Cet accessoire est un des éléments clés de l'environnement graphique WIMP¹¹ (pour la postérité, c'est le Macintosh® qui restera comme le précurseur en ce domaine). La démocratisation de l'ordinateur est, en grande partie, due à l'avènement de ces nouvelles interfaces graphiques. Les fenêtres se sont progressivement garnies d'icônes, de règles à curseur, de combo-box, etc. que l'on nomme *widgets*. Cette population de petits symboles graphiques grossit régulièrement ses rangs en intégrant des représentations similaires à des dispositifs de la vie courante : potentiomètres, boutons poussoirs, interrupteurs multipositions. L'idée sous-tendue est de simuler le comportement réel de ces dispositifs. Beaucoup de *réflexes opératoires* sont associés aux représentations des widgets, et l'utilisateur s'approprie d'autant plus facilement le logiciel que l'interface lui semble familière. Par exemple, la représentation du menu déroulant déclenche quasi-instantanément chez l'utilisateur le réflexe de choix d'élément dans une liste finie de propositions. A l'inverse, une zone de capture de texte lui suggère une donnée

11. Windows, Icon, Mouse and Pull-down menu (les éléments incontournables des environnements graphiques actuels).

de valeur non prédéfinie. Cette habitude de manipulation, nous avons voulu la mettre à profit dans notre interface graphique. Notre tâche a été grandement facilitée par l'emploi de widgets propres au système sur lequel l'opérateur a l'habitude de travailler, et ce grâce à Tk qui reprend intégralement les widget du système sur lequel il est implanté. On évite ainsi l'apprentissage de nouvelles représentations à l'utilisateur et l'on favorise l'*intégration cohérente* d'outils distincts sur un même poste de travail.

L'ergonomie d'un logiciel ne saurait se réduire à l'apparat de son interface graphique (on a d'ailleurs trop souvent tendance à confondre *interface homme/machine* et *interface graphique*). L'interface graphique n'est qu'un outil du dialogue. Son élaboration n'est donc pas aussi simple et le résultat conditionne grandement la facilité de prise en main du logiciel. M. Barthe, dans [Bar95], pense que « l'ergonomie est un problème de conception et non de réalisation du logiciel ». Il dispense donc un certain nombre de conseils à suivre lors de la conception du logiciel. En premier lieu, il préconise un « tri pertinent de l'essentiel et de l'accessoire [qui] suppose une parfaite connaissance du travail des opérateurs ». Cette opinion est unanimement partagée par les acteurs du Génie Logiciel. Les besoins, et surtout la prise en compte du fonctionnement intellectuel de l'utilisateur importe plus que l'apparence de l'écran. En d'autres termes, le concepteur de logiciel doit épouser les attentes du futur utilisateur afin de lui proposer un outil avec lequel il pourra travailler en parfaite *synergie*. On déplore aussi le manque de considération des erreurs possibles de l'opérateur. On doit tenter de mettre en place des mécanismes visant à éviter les erreurs et une méthodologie de traitement de ces erreurs. Enfin, comme nous l'avons vu dans les paragraphes précédents, l'ordonnancement des opérations est aussi un point important de l'ergonomie. Le logiciel ne doit pas être à l'initiative du dialogue, mais il doit être attentif aux désirs de communication de l'opérateur. Cela se traduit par un accès facilité aux éléments de dialogue là où le concepteur peut en éprouver le besoin. A l'inverse, de nombreux logiciels proposent des structures de menu qui ne sont qu'une reproduction de la structure interne du logiciel ; on est alors dans le cas où la *logique d'utilisation* n'est pas toujours identique à la *logique de fonctionnement*.

6.4 Commentaires sur l'interface proposée.

Plusieurs commentaires s'imposent à l'issue des paragraphes précédents. Le système que nous avons mis en place tente de respecter le mode de travail du concepteur dans la phase de détermination des DTLM. L'interface graphique, qui ne représente que la part visible du logiciel, s'adapte à ce mode de travail. On retrouve, au travers de la hiérarchie des menus, une partie de la structure de données orientée objet discutée au chapitre 2. La hiérarchisation des données se traduit, par exemple, par une chronologie de renseignement du système et une différenciation des fenêtres (comme l'objectif d'optimisation qui se situe dans la fenêtre précédent celle du choix des arrêts axiaux de la liaison arbre/moyeu). Néanmoins, nous avons tenu à limiter l'explosion du nombre de fenêtres afin d'améliorer la lisibilité et le repérage du concepteur ; c'est pourquoi l'arborescence des menus ne présente pas plus de deux niveaux. Rappelons que *Sicam 2* ne présente pas de point de blocage, car le concepteur peut toujours remonter dans la hiérarchie de définition des DTLM afin de tester de nouvelles solutions. Tous ces efforts ont pour but l'amélioration de l'ergonomie du logiciel, d'un point de vue fonctionnel et esthétique. Cet objectif peut sembler secondaire dans le cadre de travaux de recherche, mais nous pensons qu'une étude de faisabilité doit montrer les possibilités d'avancement dans tous les domaines concernés. Le dialogue homme/machine est à la base du logiciel de CAO, et s'avère parfois être son talon d'Achille. De plus, dans le cas d'une campagne d'évaluation (test en grandeur réelle),

une interface homme/machine réussie constitue un élément perturbateur de moins au travail d'analyse.

Suivant ce principe, nous n'avons pas effectué de modification de l'interface d'I-DEAS®, bien que cela eut été aisément réalisable, pour ne pas perturber l'utilisateur dans son utilisation quotidienne du logiciel. Il doit pouvoir poursuivre sa conception à l'aide des éléments de mécanisme obtenus grâce à *Sicam2*, les DTLM, de façon transparente. Dans le même esprit de transparence, signalons l'homogénéité de la signalétique employée dans *Sicam2*, ainsi que l'utilisation des widgets propres au système, par le biais de Tk.

Il serait cependant prétentieux de dire que l'interface homme/machine de *Sicam2* est parfaite. Nous déplorons une lenteur certaine dans la phase de visualisation 3D qui est de nature à générer un handicap ergonomique. On doit envisager une implémentation totale en C.O.O. afin de limiter les nombreux transferts entre les modules. La phase de redimensionnement des modèles 3D est particulièrement longue, mais elle reste compétitive par rapport à ce que pourrait obtenir manuellement un opérateur (5 à 10 fois plus rapide suivant le niveau d'expérience).

Le traitement événementiel constitue une avancée considérable dans l'amélioration des relations entre l'opérateur et la machine. Il permet notamment une gestion *active* et *préventive* des erreurs en neutralisant par exemple, en temps réel, les options contraires à un contexte créé dynamiquement par l'opérateur. Nul doute que ce genre de traitement de l'information en temps réel puisse apporter une meilleure convivialité aux logiciels de CAO. Enfin, il est légitime de se questionner sur le devenir de nos observations dans le cas où les modèles employés seraient sensiblement complexifiés. A cela, nous répondrons en deux points :

- La croissance de la puissance de calcul des ordinateurs (loi de Moore) est telle que l'on ne peut plus invoquer la complexité des modèles comme frein au développement d'assistances informatiques. Certaines méthodes de programmation que l'on délaissait voilà 15 ans, comme l'énumération, sont de nouveau exploitées, avec les avantages qu'elles seules procurent. Il faut savoir adapter la complexité des modèles aux méthodes informatiques disponibles, sachant que les règles de conception du Génie Logiciel évoluent rapidement.
- Nous nous situons dans un contexte d'avant-projet, pour lequel de nombreuses données sont soumises à variations. L'emploi de modèles agrégés simples représente une nécessité. Ces modèles dirigent le concepteur vers un domaine de solutions qui doit volontairement rester large. Imaginons un instant que les modèles mis en place soient rigoureux au point de ne posséder qu'une marge d'erreur infinitésimale. On pourrait donc définir au plus juste tous les composants du mécanisme avec le risque que la moindre modification d'un des paramètres remettent en cause l'intégralité du mécanisme. En quelque sorte, les « marges d'erreur » des modèles fournissent une certaine pérennité aux solutions proposées.

L'idée de prôner l'utilisation de modèles simplificateurs, et non « simplistes », est liée à la synthèse progressive d'un mécanisme. De même que le concepteur progresse dans la structure du mécanisme, les modèles doivent gagner en précision ; cette précision doit toutefois rester à l'échelle des approximations faites sur le fonctionnement du mécanisme. La définition des DTLM n'est pas l'étape ultime de ce processus, et l'introduction à ce niveau, par exemple, de modélisations fines des accidents de formes n'aurait aucun sens compte tenu de l'imprécision du calcul des efforts dans le système.

Le prochain chapitre présente les conclusions de notre travail et dégage des perspectives de recherche dans les différents domaines que nous avons abordés.

Chapitre 7

Conclusions et perspectives.

7.1 Conclusions

Dans la première partie de ce mémoire, nous avons proposé une définition d'organisation opérationnelle de la phase de synthèse de mécanismes. Cette organisation repose sur un découpage de l'activité en tâches associées à des niveaux de définition du produit. Nous nous sommes interrogés sur le type d'assistance que pouvait attendre le concepteur durant ces étapes de la conception et la forme que cette assistance pouvait prendre. Parmi les tâches que nous avons définies nous nous sommes intéressés à celle prenant en charge le choix et le dimensionnement des dispositifs technologiques réalisant les liaisons mécaniques du système que nous avons nommés DTLM. C'est une étape stratégique pendant laquelle le concepteur doit migrer du concept abstrait (la fonction) au concret (la matière). C'est bien souvent là qu'apparaissent les problèmes (incohérence de l'architecture, inadaptation des solutions technologiques existantes, incompatibilités dimensionnelles...). Notre réflexion sur le type d'assistance à fournir nous a guidés vers l'élaboration d'une structure de données orientée objet à destination des DTLM. Nous avons ensuite développé un outil d'assistance à la synthèse dimensionnelle des DTLM en prenant soin de dissocier le niveau d'assistance en deux phases : la synthèse découplée et la synthèse coordonnée.

Le découpage de l'organisation opérationnelle que nous proposons dissocie les différentes activités liées à la synthèse d'un mécanisme. Il permet d'adapter l'assistance à la nature des entités manipulées, aux méthodes métiers traditionnellement employées... La structure de notre outil est fidèle à cette organisation. Par exemple, l'indépendance des modules de calcul des liaisons autorise l'utilisation de méthodes d'optimisation différentes adaptées à la nature des problèmes locaux (souvent discrète dans notre cas).

À partir de cette organisation, nous avons élaboré une structure de données orientées objet appliquée aux DTLM qui reflète parfaitement les classifications couramment admises de liaisons mécaniques. Cette structure fait également apparaître les spécificités des liaisons non décomposables telles que la liaison pivot par contact oblique. Un certain nombre de DTLM ont été construits afin de remplir cette structure de données.

La mise en œuvre de la phase de synthèse se fait en deux temps.

Le concepteur débute par une phase de synthèse découplée qui a pour objectif de l'assister dans le choix et le dimensionnement *individuel* des DTLM (i.e. sans prise en compte automatique des contraintes de couplage). Il dispose d'une totale liberté dans le choix des DTLM, de leur paramétrage et des objectifs d'optimisation ; il peut étudier l'influence de ses choix sur chacune des liaisons indépendamment des autres en relançant, à volonté, les modules de calcul. Il dispose également de moyens pour contrôler la production de DTLM solutions. Ceci est réalisé grâce

à un mécanisme de valeurs limites qu'il impose aux paramètres dimensionnels du DTLM. Il a ainsi la possibilité d'arbitrer les conflits entre DTLM lorsqu'il se préoccupe de leur association. Cependant, nous avons montré que l'efficacité d'une telle méthode reposait essentiellement sur l'expertise du concepteur et sur le niveau d'informations mises à sa disposition. Pour des cas simples, ce dernier peut raisonnablement trouver une solution qui, même si elle n'est pas optimale au sens global du mécanisme, peut s'avérer suffisamment satisfaisante. En revanche, lorsque les contraintes de couplage s'accumulent, il devient difficile pour le concepteur d'arbitrer les conflits. C'est pourquoi nous avons cherché à développer un outil de synthèse coordonné.

La phase de synthèse coordonnée est une étape supplémentaire vers l'obtention d'une solution globale optimale. Elle propose de gérer, de façon automatique, les contraintes de couplage. Plusieurs approches ont été testées. La première s'inspire des méthodes employées en optimisation collaborative. Elle permet de considérer le dimensionnement des DTLM comme autant de disciplines de conception qui doivent collaborer à la construction d'un mécanisme. Cette méthode respecte la totale indépendance des différents modules de calcul et conserve les objectifs d'optimisation locaux grâce à l'emploi d'un objectif global multicritère. Les résultats ont montré qu'il était possible d'obtenir par cette méthode une nette amélioration de la solution par rapport au point de départ, mais qu'il pouvait subsister des conflits irrésolus. La nature discrète des paramètres des DTLM et l'adéquation des techniques d'optimisation employées sont en cause. Nous avons donc conçu une méthode originale basée sur le principe de relaxation sélective des contraintes que nous avons pu ajuster aux spécificités de notre problème. L'utilisation de cet algorithme permet d'atteindre dans la plupart des cas une solution réalisable. Néanmoins, l'utilisation d'objectifs locaux hétérogènes (impliquant un objectif global multicritère) gêne considérablement le processus. La méthode de relaxation sélective associée à des objectifs locaux homogènes a donné de très bons résultats, même appliquée à un problème présentant une grande combinatoire de solutions. La qualité des résultats produits par ces différentes méthodes a pu être mesurée par comparaison avec ceux produits par une méthode énumérative.

La construction conjointe des modèles 3D des DTLM nous a semblé incontournable. Le concepteur visualise ainsi les résultats obtenus et en évalue la qualité, sans ambiguïté. De plus, ces modèles pourraient être exploités dans la suite du processus de conception, notamment dans la phase d'habillage des pièces. La continuité du processus de conception (dans un contexte de maquettage virtuel par exemple) serait ainsi assurée.

Plusieurs pistes de recherche sont apparues au cours de l'élaboration de *Sicam2*. Elles méritent, dans le cadre d'une poursuite de nos travaux, d'être étudiées plus en détail. Ce sont les perspectives que nous évoquons ci-après.

7.2 Perspectives

Nous avons dégagé quelques grands axes de réflexion :

Amélioration des fonctionnalités de notre outil d'assistance. Elle passe tout d'abord par l'élargissement de l'éventail des technologies de DTLM proposés au choix du concepteur pour les liaisons déjà implémentées. Elle passe également par l'addition de nouveaux types de liaisons à la gamme actuellement proposée. Enfin, il serait intéressant de mesurer les performances de notre démarche dans le cas de mécanismes présentant un plus grand nombre de liaisons. Il s'agirait notamment d'étudier l'impact d'une augmentation du nombre de liaisons sur le processus de synthèse coordonnée.

Universalisation des DTLM. *Sicam2* est une maquette logicielle destinée à valider notre approche. Notre objectif n'ayant jamais été de concevoir un logiciel à vocation commerciale, nous nous sommes appuyés sur des produits professionnels (I-DEAS® notamment). Nous avons, grâce à cette maquette, démontré que nos concepts pouvaient s'accommoder de systèmes de CAO existants. Mais les impératifs dus au contexte de réalisation de *Sicam2* l'ont quelque peu rendu dépendant de ces produits. Une implémentation totale de l'architecture objet permettrait de s'affranchir de ces servitudes. La définition d'interfaces standards doit mener à la création d'une bibliothèque universelle de DTLM implémentables dans n'importe quel système de conception intégrée.

Extension du domaine d'assistance. Dans le cadre de la recherche en conception intégrée, *Sicam2* devrait prendre place dans un système plus large comportant une assistance aux divers niveaux de notre organisation opérationnelle tels que l'habillage des pièces mécaniques ou la détermination de la nature des liaisons par exemple.

Certaines difficultés rencontrées lors de la synthèse des mécanismes avec *Sicam2* ont particulièrement retenu notre attention.

- La condition d'antériorité imposée au concepteur pour le placement des centres de liaisons peut paraître pénalisante, bien que pleinement justifiée par notre organisation opérationnelle. En effet, le concepteur se retrouve parfois dans des cas où une petite modification du placement des DTLM permettrait de débloquer une situation conflictuelle (particulièrement dans les cas d'étude de forte compacité). Il pourrait être bénéfique de pouvoir autoriser un faible déplacement de ces centres de liaison, sans pour autant aller jusqu'à proposer un placement dynamique des liaisons (ce qui remettrait en cause notre organisation opérationnelle). Cela permettrait, par exemple, de pouvoir « faire glisser » certains DTLM à qui il ne manquerait que très peu de place en encombrement axial. Le principe de placement « a priori » des centres de liaisons resterait valable tout en introduisant une certaine tolérance sur la position. Ceci entraînerait, bien entendu, la prise en compte de paramètres supplémentaires dans la phase de synthèse coordonnée qu'il serait intéressant d'étudier.
- Nous avons discuté de l'intérêt de la diversité des objectifs locaux lors de la phase de synthèse découplée. L'utilisation d'objectifs multicritères lors de la synthèse coordonnée assure la constance des intentions du concepteur mais s'avère parfois pénalisante. Nous devons travailler à la définition de ces objectifs. Nous pourrions également envisager d'autres méthodes d'optimisation mieux adaptées au traitement d'objectifs multicritère.
- La problématique des liaisons décomposables ou non a été soulevée. Dans le cas de la liaison pivot réalisée par contact oblique, nous avons pris le parti d'utiliser deux DTLM dépendants. Nous aurions pu faire le choix inverse consistant à n'utiliser qu'un seul DTLM mais présentant un paramétrage différent. Cette éventualité mérite d'être étudiée car l'extension de *Sicam2* à d'autres types de liaison fera inmanquablement ressurgir le problème.

Les pistes d'investigation sont nombreuses et témoignent du chemin restant à accomplir. Néanmoins, la base de travail que nous proposons est opérationnelle et permet d'envisager une poursuite des travaux.

Bibliographie

- [AA00] A. ANUAR et J. ATKINSON. «A software system for specifying design procedures». *Journal of engineering design*, 11(3):191–210, 2000.
- [AC98] Thierry ANDRO et Jean-Marie CHAUVET. *Objets métiers*. Eyrolles, Paris, 1998.
- [AC01] Inc. AUTOMATION CREATION. «MATWEB the Online Materials Information Resource». Page www, avril 2001. <http://www.matweb.com> visité avril 2001.
- [AF96] M. AYACHE et A. FLORY. *Approche orientée objet*. Economica, Paris, 1996.
- [AFB96] Boris ABERŠEK, Jože FLAŠKER, et Jože BALIČ. «Expert system for designing and manufacturing of a gear box». *Expert Systems with Applications*, 11(3):397–405, 1996.
- [AFN55] AFNOR. «Cannelures rectilignes à flancs en développante». Norme française, décembre 1955. NF E 22-141.
- [AFN80] AFNOR. «Assemblages frettés. Dimensions, tolérances et états de surface pour assemblages usuels». Norme française, mars 1980. NF E 22-620.
- [AFN84] AFNOR. «Assemblages frettés sur portée cylindrique. Fonction, réalisation, calcul.». Norme française, janvier 1984. NF E 22-621.
- [AFN85] AFNOR. «Cannelures cylindriques droites à flancs en développante. Règles pour le choix dimensionnel et la vérification.». Norme française, novembre 1985. NF E 22-148.
- [AFN87a] AFNOR. «Cannelures cylindriques droites à flancs en développante et centrage sur flancs. Angles de pression 30°, 37,5° et 45°». Norme française, octobre 1987. NF E 22-144.
- [AFN87b] AFNOR. «Clavettes parallèles.Dimensions». Norme française, novembre 1987. NF E 22-177.
- [AFN87c] AFNOR. «Rainures pour clavettes parallèles. Dimensions ». Norme française, novembre 1987. NF E 22-175.
- [AFN91] AFNOR. «Analyse de la valeur, analyse fonctionnelle NF X50-151 ». Rapport Technique, AFNOR, 1991.
- [Alz93] Hussam Aldin ALZEIN. *Gestion de modification de structures dans un système de C.A.O.-batiment*. Thèse de doctorat, Université de Savoie, 1993.

- [Ama98] Abdelmejid Ben AMARA. *Contribution à l'intégration de la composante calcul dans une démarche fonctionnelle intégrée*. Thèse de doctorat, Université de Valenciennes, 1998.
- [Ano91] ANONYME. « Current state of the art on multidisciplinary design optimization (MDO) – an AIAA white paper ». AIAA, Washington, DC, septembre 1991. ISBN 1-56347-021-7.
- [Art76] I. ARTOBOLEVSKI. *Les mécanismes dans la technique moderne*, volume 1-5. Éditions MIR, Moscou, 1976. Mécanismes à leviers, 710 pages, traduction française de 1976.
- [Ash00] Michael F. ASHBY. *Choix des matériaux en conception mécanique*. Techniques et Ingénieries. DUNOD, 2000.
- [Bar95] Michel BARTHE. *Ergonomie des logiciels*. Masson, 1995. collection MIPS.
- [Bas98] David BASSETTI. « Logiciel d'aide à la sélection des matériaux ». *Matériaux et techniques*, 86(5-6):21–29, 1998.
- [BD85] V. BLANCHIN et S. D'ALBERTO. « Contribution à SICAM: recherche d'une solution optimale ». Projet de fin d'études, INSA Toulouse, juin 1985.
- [Bea98] Ph. BEAUFILS. « Le savoir faire métier enrichit la CAO ». *Industries et Techniques*, 796:74–79, septembre 1998.
- [Ben93] Driss BENCHAOUINE. *Système expert en mécanique. Application au choix et au montage des éléments technologiques*. Thèse de doctorat, INSA Lyon, mars 1993.
- [BG98] R. J. BALLING et D. L. GALE. « Collaborative optimization of systems involving discrete design at the discipline level ». *Journal of Mechanical Design*, 120:32–39, mars 1998.
- [Boc98] Jean-Claude BOCQUET. « Modèles fonctionnels ». Dans *Conception de produits mécaniques*, Chapitre 16, pages 397–411. Hermes, 1998. Sous la direction de Michel Tollenaere.
- [Boo94] Grady BOOCH. *Analyse et conception orientée objet*. Addison-Wesley, Paris, 1994.
- [Bor86] M. BORRIEN. « Rayon minimal en fond de clavette ». rapport final, CETIM, 1986.
- [Bro01] Frank B. BROKKEN. *C++ Annotations*. Université de Groningen, 2001. <ftp://ftp.rug.nl/contrib/frank/documents/cplusplus.annotations>.
- [BS70] G. S. BEVRIDGE et R. S. SCHECHTER. *Optimization: Theory and Practice*. McGraw-Hill, Tokyo, 1970.
- [BS95] R. J. BALLING et J. SOBIESZCZANSKI–SOBIESKI. « An algorithm for solving the system-level problem in multilevel optimization ». *Structural optimization*, 9:168–177, 1995.
- [BS96] R. J. BALLING et J. SOBIESZCZANSKI–SOBIESKI. « Optimization of coupled systems: a critical overview of approaches ». *AIAA Journal*, 34(1):6–17, janvier 1996.

- [CdP73] J. CHABAL et R. de PREESTER. *Méthodologie de la construction mécanique*. Foucher, 1973.
- [CH95] Xiangping CHEN et Christoph M HOFFMANN. « On editability of feature-based design ». *Computer-Aided Design*, 27(12):905–914, 1995.
- [CH00] K. CASE et M. S. HOUNSELL. « Feature modelling : a validation methodology and its evaluation ». *Journal of materials processing technology*, 107:15–23, 2000.
- [Cho92] Éric CHOTIN. *Placement automatique de circuits intégrés*. Thèse de doctorat, I.N.P.Grenoble, 1992.
- [Con96] Damien CONSTANT. *Contribution à la spécification d'un modèle fonctionnel de produits pour la conception intégrée de systèmes mécanique*. Thèse de doctorat, Université Joseph Fourier – Grenoble 1, octobre 1996.
- [Cor02] Tech-X CORPORATION. « OptSolve++ ». pages www, visité en juin 2002. OptSolve++ manual
<http://www.techxhome/products/optsolve/>.
- [CW96] Susan E. CARLSON et Michael D. WHITE. « A genetic approach to catalog design ». Dans AAAI PRESS, éditeur, *AAAI Fall Symposium Technical Report*, page 154, 1996.
- [CWDH00] Graham COATES, Robert WHITFIELD, Alex DUFFY, et Bill HILLS. « Coordination approaches and systems - part II: an operational perspective ». *Research in Engineering Design*, 12:73–89, 2000.
- [CY92] Peter COAD et Edward YOURDON. *Analyse orientée objet*. Masson, Paris, 1992.
- [Dai96] Alain DAIDIÉ. *Réalisation d'un logiciel de conception automatique appliqué à la détermination des engrenages cylindriques : interfaçage avec un logiciel industriel de CAO*. Thèse de doctorat (spécialité : génie mécanique), INSA Toulouse, LGMT - Laboratoire de Génie Mécanique de Toulouse, janvier 1996.
- [Dam96] Mihai DAMIAN. *Conception automatique de liaisons pivot par roulements utilisant les méthodes de l'intelligence artificielle*. Thèse de doctorat, Université Paul Sabatier (Toulouse III), décembre 1996.
- [Def85] Yves DEFORGE. *Technologie et génétique de l'objet industriel*. Maloine, 1985.
- [Deg96] Irène DEGIRMENCYAN. *Du paradigme essai-erreur au paradigme du choix dirigé. Application à la conception mécanique. Acteur unique, acteurs multiples*. Thèse de doctorat (spécialité : automatique), École Centrale de Paris, Laboratoire d'Ingénierie Informatique et Systèmes Intégrés de l'Institut Supérieur des Matériaux et de la Construction Mécanique, janvier 1996.
- [DF95] N. DURAND et T. FONTAINE. *Réussir un projet de conception orientée objet*. Masson, Paris, 1995.
- [EM01] EADS-MATRA. « OpenCASCADE the open source 3D modeling kernel by OpenCASCADE ». Page WWW, avril 2001. <http://www.opencascade.com> visité avril 2001.

- [FA89] H. FESSLER et T. APPAVOO. « On the effect of key edge shape on keyway edge stresses in shafts in torsion ». *Journal of strain analysis*, 24(3):121–125, 1989.
- [Fau99] Jean-Christophe FAUROUX. *Conception optimale de structures cinématiques tridimensionnelles. Application aux mécanismes de transmission en rotation*. Thèse de doctorat, INSA de Toulouse, 1999.
- [FE82] H. FESSLER et M. EISSA. « Elastic stresses due to torque transmitted through the prismatic part of keyed connections – Part 2 – ». *Journal of strain analysis*, 17(4):215–222, 1982.
- [FE83] H. FESSLER et M. EISSA. « Three-dimensional elastic stress distribution in end-milled, keyed connections ». *Journal of strain analysis*, 18(2):143–149, 1983.
- [Fer91] J. FERBER. *Conception et programmation par objets*. Hermes, 1991.
- [GC84] J. GUILLOT et B. CASENAVE. « Logiciel de conception de mécanismes à partir des réalisations technologiques des liaisons ». Dans *31^e conférence européenne sur la CFAO et l'infographie*. MICAD, 1984.
- [GMR81] J. GUILLOT, J. MULLOR, et J.Y. ROUSSELOT. « SICAM: Un exemple de CAO en mécanique ». Dans *3^e journées scientifiques et techniques de la production automatisée*, pages 70–78, Toulouse, juin 1981. ADEPA.
- [GMSB96] Marie-Claude GAUDEL, Bruno MARRE, Françoise SCHLIENGER, et Gilles BERNOT. *Précis de génie logiciel*. Masson, 1996.
- [GR01] S. K. GUPTA et W. C. REGLI. « Editorial ». *Computer-Aided Design*, 33:619–620, 2001.
- [Gro01] Object Management GROUP. « Page Principale OMG ». Page Web, juillet 2001. <http://www.omg.org>.
- [Gui87] Jean GUILLOT. *Méthodologie de définition des ensembles mécaniques en CAO. Recherche des solutions optimales*. Thèse d'état, Université Paul Sabatier - INSA de Toulouse, 1987.
- [HJ98] Christoph M. HOFFMANN et Robert JOAN-ARINYO. « On user-defined features ». *Computer-Aided Design*, 30(5):321–332, 1998.
- [HW98] Wynne HSU et Irene M. Y. WOON. « Current research in the conceptual design of mechanical products ». *Computer-Aided Design*, 30(5):377–389, 1998.
- [HWW98] Q. J. HARMER, P. M. WEAVER, et K. M. WALLACE. « Design-led component selection ». *Computer-Aided Design*, 30(5):391–406, 1998.
- [Jau90] Patrick JAULENT. *Génie logiciel: les méthodes*. Armand Colin, 1990.
- [JG00] K. JHA et B. GURUMOORTHY. « Automatic propagation of feature modification across domain ». *Computer-Aided Design*, 32:691–706, 2000.
- [Joh78] Ray C. JOHNSON. *Mechanical design synthesis*. Robert E. Krieger, 2^{ème} édition, 1978.

- [Joh80] Ray C. JOHNSON. *Optimum design of mechanical elements*. Wiley-Interscience Publication, 2ème édition, 1980.
- [KR84] B. W. KERNIGHAN et M. D. RITCHIE. *Le langage C*. Masson, Paris, 2ème édition, 1984.
- [KS96] Dileep KHADILKAR et Larry STAUFFER. «An experimental evaluation of design information reuse during conceptual design». *Journal of engineering design*, 7(4):331–339, 1996.
- [KS01] S. KODIYALAM et J. SOBIESCZANSKI–SOBIESKI. «Multidisciplinary design optimization – some formal methods, framework requirements, and application to vehicle design». *International Journal of Vehicle Design*, special issue:3–22, 2001.
- [LO98] Wen-Yau LIANG et Peter O’GRADY. «Design with objects: an approach to object-oriented design». *Computer-Aided Design*, 30(12):943–956, 1998.
- [Loy91] Michel LOYER. «La CAO le DAO». Dans *Que sais-je*, numéro 2612. Presses Universitaires de France, 1991.
- [Mao95] Salah MAOUCHE. *Contribution à l’optimisation du placement de formes quelconques: application à l’industrie de la confection*. Thèse de doctorat, Université de Lille 1, 1995.
- [Mar95] P. MARCHAND. «Les bibliothèques de composants mécaniques pour la CAO». Rapport d’étude 194340, CETIM, janvier 1995.
- [Mar99] Carlos MARTINS. *Optimisation de systèmes mécaniques par décomposition en sous-systèmes simples*. Thèse de doctorat, INSA Toulouse, janvier 1999.
- [MDP01] Cédric MASCLET, Alain DAIDIÉ, et Manuel PAREDES. «Introducing CAD’s high level features: the TDMJs». Dans *2ème conférence internationale CPI’2001*, Fes, Maroc, octobre 2001.
- [Meh95] Kamel MEHDI. *Système intégré de conception des boîtes de transmission par engrenages*. Thèse de doctorat, INSA Lyon, mai 1995.
- [MGM98] Carlos MARTINS, Jean GUILLOT, et Cédric MASCLET. «Decomposition and collaborative methods in nonhierarchical optimal design». Dans B. KATALINIC, éditeur, *9th International DAAAM Symposium*, pages 293–294, Vienna, Austria, 1998. DAAAM International.
- [Min89] M. MINOUX. *Programmation mathématique. Théorie et algorithmes*. Dunod, 1989.
- [MM00] A. MEDLAND et G. MULINNEUX. «A decomposition strategy for conceptual design». *Journal of engineering design*, 11(1):3–16, 2000.
- [MSFP99] Cédric MASCLET, Marc SARTOR, Jean-Christophe FAUROUX, et Manuel PAREDES. «Définition d’une organisation pour la conception optimale de mécanismes à 1 ddl». Dans *Congrès français de mécanique*, Toulouse, septembre 1999.

- [MSP00] Cédric MASCLLET, Marc SARTOR, et Manuel PAREDES. «Object-oriented approach to the creation of a CAD tool for choosing Technological Devices Used as Mechanical Joints». Dans *3rd International conference on Integrated Design and Manufacturing in Mechanical Engineering*, Montréal, mai 2000.
- [Ous94] John K. OUSTERHOUT. *Tcl and the Tk toolkit*. Addison-Wesley Professional Computing, mai 1994. ISBN 0-20-163337-X, <http://www.scriptics.com/resource/doc/books/>.
- [Ous98] John K. OUSTERHOUT. «Scripting: Higher Level Programming for the 21st Century». *IEEE Computer*, mars 1998. Consultable en ligne à l'URL : <http://www.scriptics.com/people/john.ousterhout/scripting.html>.
- [Pam92] Jose-Alfonso PAMANES-GARCIA. *Contribution à l'étude de l'accessibilité aux tâches et à la détermination du placement optimal de robots manipulateurs*. Thèse de doctorat, Université de Poitiers, 1992.
- [PB96] Gerhard PAHL et Wolfgang BEITZ. *Engineering design. A systematic approach*. Springer-Verlag, 2ème édition, 1996.
- [Pil97] Walter D. PILKEY. *Peterson's stress concentration factors*. John Wiley & sons, 2ème édition, 1997.
- [Poi89] Jean-Pierre POITOU. *30 ans de CAO en France*. Hermes, Paris, 1989.
- [Pow64] M. J. D. POWELL. «An efficient method for finding the minimum of a function of several variables without calculating derivatives». *The Computer Journal*, 7:155–162, 1964.
- [Pre00] Jean-Francois PREVEREAUD. «Il faudrait être fou pour dépenser plus!». *Industries et Techniques*, 818, septembre 2000.
- [PTVF92] William H. PRESS, Saul A. TEUKOLSKY, William T. VETTERLING, et Brian P. FLANNERY. *Numerical recipes in C: the art of scientific computing*. cambridge university press, 2ème édition, 1992.
- [Qam98] A. Z. QAMHIYAH. «A strategy for the construction of customized design libraries for CAD». *Computer-Aided Design*, 30(11):897–904, 1998.
- [RDR02] Josette REY-DEBOVE et Alain REY. *Le nouveau petit Robert*. Le Robert, 2002.
- [Rum95] James RUMBAUGH. *OMT*. Masson Paris, London, 3ème édition, 1995.
- [SCR98] SCRIPTICS CORPORATION. «Scriptics Home Page». Page WWW, 1998. <http://www.scriptics.com>. La dernière version de Tcl/Tk y est téléchargeable.
- [SCW98] S. SZYCKMAN, J. CAGAN, et P. WEISSER. «An integrated approach to optimal three dimensional layout and routing». *Journal of Mechanical Design*, 120(3):510–512, 1998.
- [Sil94] Yves SILBERBERG. *Réalisation d'un système intelligent d'aide à la conception de mécanismes. Application à la réalisation d'un arbre de transmission*. Thèse de doctorat (spécialité: génie mécanique), Université Blaise Pascal - École doctorale des sciences pour l'ingénieur de Clermont-Ferrand, octobre 1994. 192 p.

- [SK93] T.S. SAKTHIVEL et V. KALYANARAMAN. « A KBES for integrated engineering ». *Engineering with computers, International journal for computer aided mechanical and structural engineering*, 9(1):1–16, 1993.
- [SKF01] SKF. « SKF Interactive Engineering Catalogue ». page WWW, 2001. <http://iec.skf.com> visité avril 2001.
- [SM89] J. E. SCHIGLEY et C. R. MISCHKE. *Mechanical engineering design*. Mc Graw Hill, New York, 5ème édition, 1989.
- [TF02] Sayeejee TATINENI et Georges FADEL. « Coupling through movelimits in multi-disciplinary optimization ». <http://citeseer.nj.nec.com/136283.html>, 2002. Lien vérifié en avril 2002.
- [Tic94] Serge TICHKIEWITCH. « De la CFAO à la conception intégrée ». *Revue internationale de CFAO et d'infographie*, 9:609–621, 1994.
- [Tol00] Michel TOLLENAERE. « Pôle de Ressources Informatique pour la MÉCANIQUE ». Page WWW, 2000. <http://ensgi.inpg.fr/PRIMECA>.
- [Tro99] Nadège TROUSSIER. *Contribution à l'intégration du calcul mécanique dans la conception de produits techniques: proposition méthodologique pour l'utilisation et la réutilisation*. Thèse de doctorat, Université Joseph Fourier, Grenoble 1, octobre 1999.
- [Tse99] Yuan-Jye TSENG. « A modular modeling approach by integrating feature recognition and feature-based design ». *Computers in industry*, 39:113–125, 1999.
- [Tum00] Sedar TUMKOR. « Internet-based design catalogue for the shaft and bearing ». *Research in engineering design*, 12:163–171, 2000. <http://www.mkn.itu.edu.tr/~Katalog>.
- [VAN84] Garret N. VANDERPLAATS. *Numerical optimization techniques for engineering design with applications*. McGraw-Hill, 1984. 333 p., ISBN 0-07-066964-3.
- [VDI93] VDI. « Methodik zum entwickeln und konstruieren technischer systeme und produkte ». Rapport Technique, V.D.I.-Verlag, Düsseldorf, 1993.
- [Vin97] Steve VINOSKI. « CORBA : integrating diverse applications within distributed heterogeneous environments. ». *IEEE Communications Magazine*, février 1997.
- [WCDH00] Robert WHITFIELD, Graham COATES, Alex DUFFY, et Bill HILLS. « Coordination approaches and systems - part I: a strategic perspective Coordination approaches and systems – part I: a strategic perspective ». *Research in Engineering Design*, 12:48–60, 2000.
- [Wel97] Brent WELCH. *Practical programming in Tcl and Tk*. Prentice Hall, 2ème édition, 1997. ISBN 0-13-616830-2, <http://www.beedub.com/book/>.
- [WGSF01] Jean-Christophe WAHL, Fabien GRENET, Marc SARTOR, et Jean-Christophe FAUROUX. « Assistance qualitative à la synthèse structurelle des mécanismes à base de mouvements de rotation et de translation ». Dans 7^e Colloque sur la conception intégrée, pages 251–258. AIP–Primeca, avril 2001.

- [WW96] Manjula B. WALDRON et Kenneth J. WALDRON. «The influence of the designer's expertise on the design process.». Dans M. B. WALDRON et K. J. WALDRON, éditeurs, *Mechanical design. Theory & Methodology*, Chapitre 2, pages 5–20. Springer-Verlag, 1996.
- [Zha97] Ying ZHANG. *Le paradigme du choix contraint orienté par objectifs pour l'aide à la conception mécanique*. Thèse de doctorat (spécialité: automatique), École Centrale de Paris, Laboratoire d'Ingénierie Intégrée des Systèmes Industriels de l'Institut Supérieur des Matériaux et de la Construction Mécanique, mars 1997.

Annexe A

Cahiers des charges du problème–support.

Les données du problème–support commun au chapitre 5 et à la section 4.5.3 sont fournies ci-dessous sous forme de copies d'écran des fenêtres-menu de définition. Il s'agit de la *formulation initiale* du problème avant toute modification du concepteur durant la phase de synthèse découplée.

CAHIER DES CHARGES PRINCIPAL

Definitions de la denture de gauche (C):	PARAMETRES DE TRANSMISSION	Definitions de la denture de droite (D):
Diametre primitif dp1(mm)= 120	Couple transmis (N.M)= 80	Diametre primitif dp2(mm)= 90
Module m1(mm)= 3	Vitesse de rotation (tr/min)= 2000	Module m2(mm)= 3
Inclinaison beta1(deg)= 0	materiau arbre: 35NCD16	Inclinaison beta2(deg)= 0
Largeur B1(mm)= 25		Largeur B2(mm)= 30
Resultante Rx1(N)= 0		Resultante Rx2(N)= 0
Resultante Ry1(N)= 484.2		Resultante Ry2(N)= 647.2
Resultante Rz1(N)= -1320		Resultante Rz2(N)= 1777.6
Moment Mx1(Nm)= 80		Moment Mx2(Nm)= -80
Moment My1(Nm)= 0		Moment My2(Nm)= 0
Moment Mz1(Nm)= 0		Moment Mz2(Nm)= 0

TOPOLOGIE DE LA LIAISON PIVOT

POSITIONS DES LIAISONS

position du palier gauche (A) en mm: a= 91.0
position du palier droit (B) en mm: b= 196.0
position de la denture gauche (C) en mm: c= 139.0
position de la denture droite (D) en mm: d= 277.0

FIG. A.1 – Fenêtre du cahier des charges global.

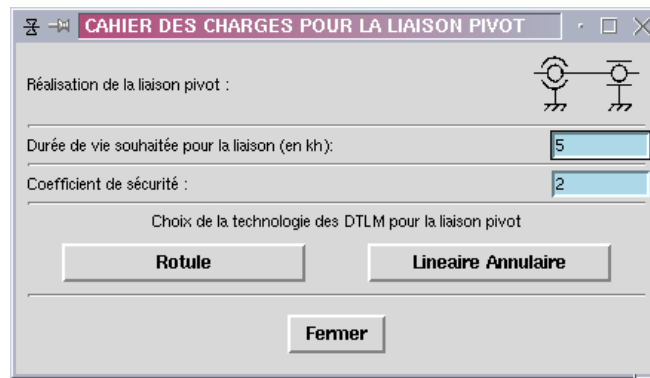


FIG. A.2 – Fenêtre de définition de la liaison pivot.

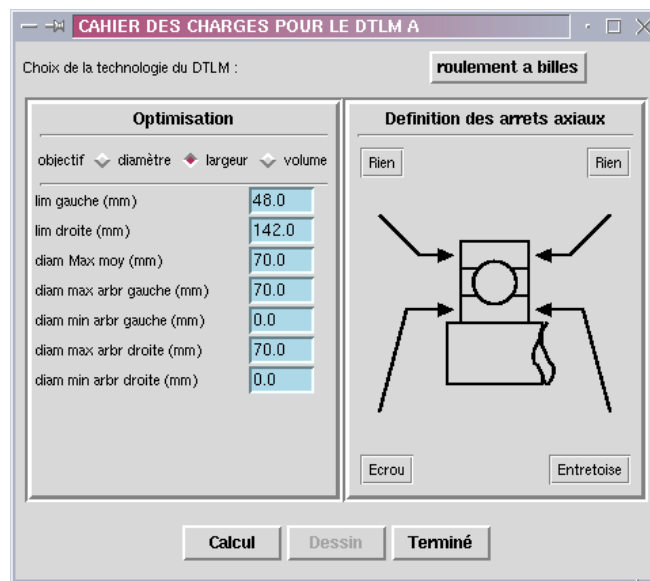


FIG. A.3 – Fenêtre de définition complémentaire de la liaison A.

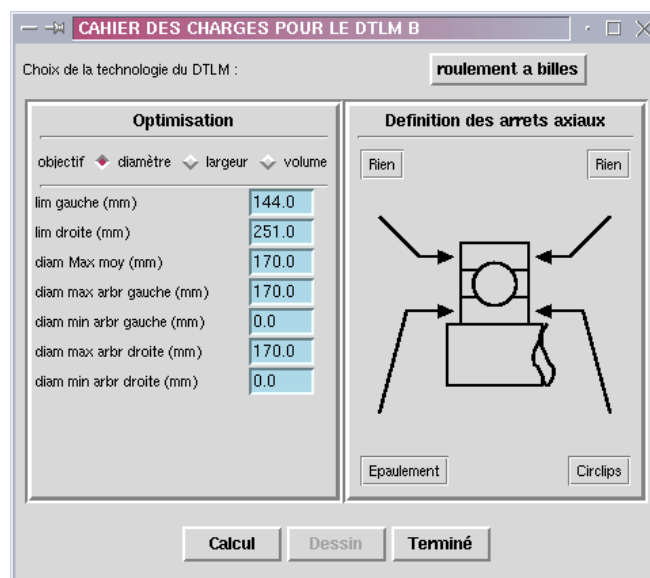


FIG. A.4 – Fenêtre de définition complémentaire de la liaison B.

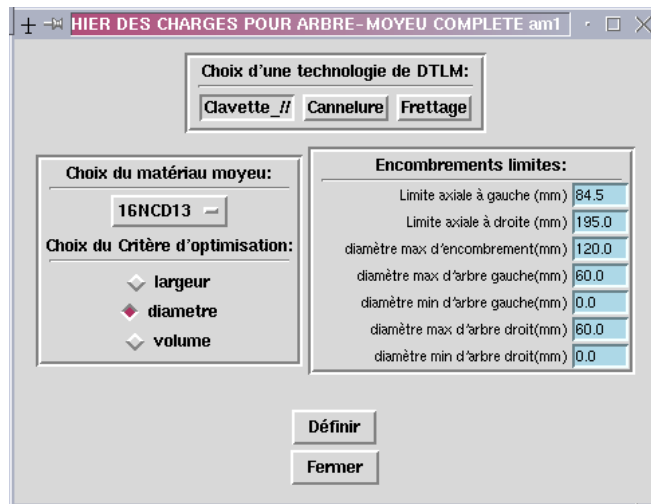


FIG. A.5 – Fenêtre du menu de définition commun de la liaison arbre-moyeu 1.

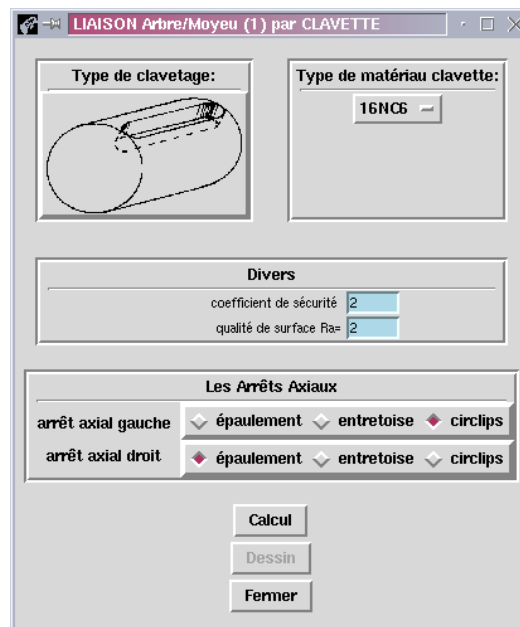


FIG. A.6 – Fenêtre du menu de définition caractéristique de la liaison arbre-moyeu 1.

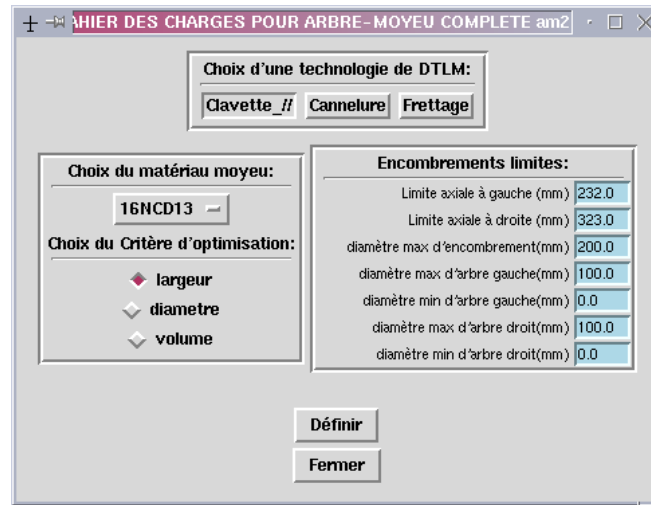


FIG. A.7 – Fenêtre du menu de définition commun de la liaison arbre-moyeu 2.

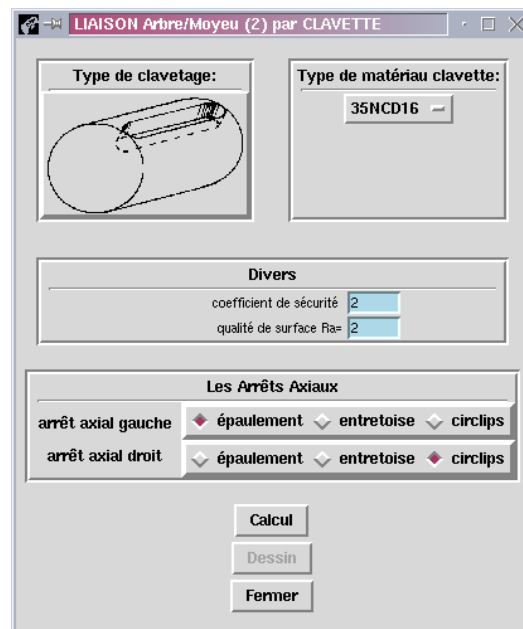


FIG. A.8 – Fenêtre du menu de définition caractéristique de la liaison arbre-moyeu 2.