HAL
open science

Calcul automatique de trajectoires pour l'assemblage d'objets déformables

Olivier Roussel, Michel Taïx, Benoit Michel, Etienne Ferré

To cite this version:

Olivier Roussel, Michel Taïx, Benoit Michel, Etienne Ferré. Calcul automatique de trajectoires pour l'assemblage d'objets déformables. Colloque AIP-Primeca, Mar 2015, La Plagne, France. hal-01128943

HAL Id: hal-01128943

https://hal.science/hal-01128943v1

Submitted on 10 Mar 2015

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.

Calcul automatique de trajectoires pour l'assemblage d'objets déformables

Olivier Roussel & Michel Taïx
CNRS, LAAS,
7 av. du colonel Roche, F-31400 Toulouse
Université de Toulouse, UPS, LAAS,
F-31400 Toulouse, France
prénom.nom@laas.fr

Benoit Michel & Etienne Ferré
Siemens Industry Software SAS
Digital Factory Division, Kineo R&D
Miniparc 2, 478 rue de la Découverte
31670 Labège, France
prénom.nom @siemens.com

Résumé— Ces travaux présentent des approches algorithmiques pour résoudre le problème de la planification de trajectoire d'objets déformables dans le cadre d'application d'assemblage /désassemblage. Nous montrons qu'il est nécessaire de trouver un compromis entre précision du modèle et temps de calcul au sein des algorithmes de planification pour résoudre de manière réaliste le problème. Des exemples de simulation illustrent les différentes approches.

Mots-clés : planification de trajectoire, déformable, assemblage, maquette numérique.

I. Introduction

Les éditeurs de logiciels de PLM proposent aux industriels (automobile, aéronautique…) un ensemble d'outils pour simuler les processus de production par modèles numériques. Un de ces outils est la simulation de montage et démontage d'une pièce, pour valider le processus d'assemblage final ou la maintenabilité d'un produit complexe tel une voiture ou un avion. Dans ce contexte il faut pouvoir déterminer automatiquement une trajectoire sans collision pour l'objet en simulant son démontage dans le modèle numérique.

Les algorithmes de planification automatique de trajectoire, « Motion Planning », *MP*, initialement développés dans le cadre de la robotique, sont de nos jours largement utilisés dans les bureaux d'études de conception et permettent de participer à la conception collaborative entre équipes de développement de nouveaux produits.

A travers le projet ANR-AMSI (Algorithmique du Mouvement et Simulation Interactive), il est apparu que de nombreux problèmes pratiques d'assemblage ou de désassemblage d'objet ne peuvent pas être résolus en considérant uniquement des objets rigides (figure 1). Par exemple, l'insertion du toit de l'habitacle d'une voiture s'effectue par une insertion en effort après une déformation locale, la pièce est ``clipsée'' en force. Si on considère le placement de câbles ou de tuyaux dans un immeuble, nous devons considérer une structure d'objet qui autorise des déformations pour pouvoir planifier les différents réseaux. Il est donc indispensable de tenir compte de la non-collision entre l'objet manipulé et l'environnement mais aussi de certaines propriétés physiques de déformation de l'objet.

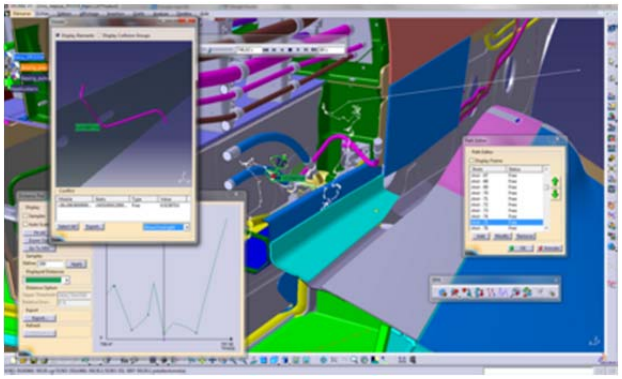

FIGURE 1: EXEMPLE D'ENVIRONNEMENT AVEC DEFORMABLES (COURTESY OF AIRBUS)

Il existe de nombreux résultats de recherches dans le cadre du *MP* pour les objets rigides [1] mais il n'en est pas de même si on considère le cas de déformables. Bien qu'il existe de nombreux travaux cherchant à planifier la trajectoire pour un objet et/ou un environnement déformable, [2, 3, 4 5, 6, 7, 8] le problème n'est pas encore résolu de manière satisfaisante pour envisager un transfert sur des logiciel commerciaux de type PLM. Les objets déformables sont plus complexes à traiter pour les algorithmes de *MP* car:

- la déformation induit des mouvements différents suivant les particules ce qui implique l'exploration d'un espace de très grande dimension,
- la déformation peut provoquer des auto-collisions qui devront être détectées,
- la physique des objets déformables qui décrit les propriétés de la matière (élasticité, raideur,...) est coûteuse en temps de calcul.

L'objectif du papier n'est pas de faire de la simulation physique de déformation d'objet, mais de proposer une algorithmique permettant d'intégrer efficacement les contraintes des déformables dans le calcul des trajectoires.

II. Algorithme de Motion Planning

Les algorithmes probabilistes de type Rapidly-Exploring Random Tree (RRT), [9], permettent de résoudre de manière

efficace le problème de *MP* pour des objets mobiles rigides. Pour relier une configuration initiale, q_i, à une configuration but, q_b, de l'objet mobile, l'algorithme développe un graphe de configurations atteignables par une trajectoire sans collision depuis q_i. L'extension consiste à sélectionner une configuration atteignable, choisir une direction d'extension, et utiliser un algorithme de planification locale pour déterminer une extension pertinente dans la direction choisie. La configuration valide ainsi atteinte après cette extension pertinente est alors ajoutée au graphe. Ce processus se poursuit jusqu'à ce que l'un des sommets, atteints de cette manière, se relie à q_b.

L'idée est d'étendre l'utilisation du RRT au cas des déformables en considérant les cas où, soit le mobile est déformable, soit l'environnement est déformable. Il faut coupler au RRT un modèle de déformation qui respecte les contraintes mécaniques de l'objet et qui soit calculable en temps réel.

III. Mobile rigide dans un environnement deformable

Dans le cas d'un mobile considéré comme un objet rigide qui se déplace dans un environnement déformable, le RRT peut être modifié par une simple adaptation de la planification locale. La méthode classique consiste à choisir une extension en ligne droite jusqu'au premier obstacle rencontré. L'ajout, lorsque cet obstacle est déformable, est de simuler sa déformation sous l'action du mobile lorsqu'il poursuit son déplacement, et de poursuivre l'extension jusqu'à ce que la force exercée devienne trop importante. Afin de simuler la déformation nous utilisons le simulateur dynamique XDE [10]. La difficulté ici est de réduire les appels au simulateur dynamique car la planification locale est appelée un grand nombre de fois (à chaque tentative d'extension du graphe) pendant une planification globale de trajectoire. En effet, la simulation de déformation, aussi rapide soit-elle dans XDE, est trop coûteuse pour des appels aussi fréquents en *MP* (facteur de l'ordre de 100). La solution choisie consiste à chercher en priorité les trajectoires évitant les obstacles déformables aussi bien que rigides et à ne faire des déformations que si nécessaires.

Cette approche donne des solutions en temps raisonnable (inférieur à 1 heure sur un cas industriel) mais le gain est discutable en le comparant avec une planification rapide qui ne considère pas l'objet déformable (temps de résolution de quelques secondes). Il faut toutefois noter que modéliser la flexibilité permet d'obtenir des informations supplémentaires. En particulier ici, la valeur maximale de la force exercée sur le flexible au cours du mouvement est connue avec les paramètres numériques employés pour le câble.

La trajectoire solution ainsi obtenue n'est pas directement exploitable car le tirage pour étendre le graphe est aléatoire. La trajectoire obtenue est donc en général extrêmement complexe, empruntant de larges détours autour des obstacles et de nombreuses portions inutiles. C'est pourquoi il est nécessaire dans un second temps de l'optimiser en retirant les points de passages et de contournement inutiles.

Pour optimiser la trajectoire, un algorithme pragmatique a été mis en place. Il consiste à synthétiser en une famille à un paramètre l'ensemble des déformations subies par le flexible au cours du mouvement du mobile. Le paramètre choisi (s sur la figure 2) est le paramètre d'avancement du mobile le long de sa trajectoire. Le problème d'optimisation est ainsi réduit à un problème à 7 degrés de liberté (6 pour le mouvement du solide et 1 pour les déformations). En discrétisant la famille des déformations, il a été possible de l'intégrer, aux algorithmes d'optimisation déjà existants, comme un mode de mouvement à 1 degré de liberté. Le principe de l'algorithme est schématisé sur la Figure 2.

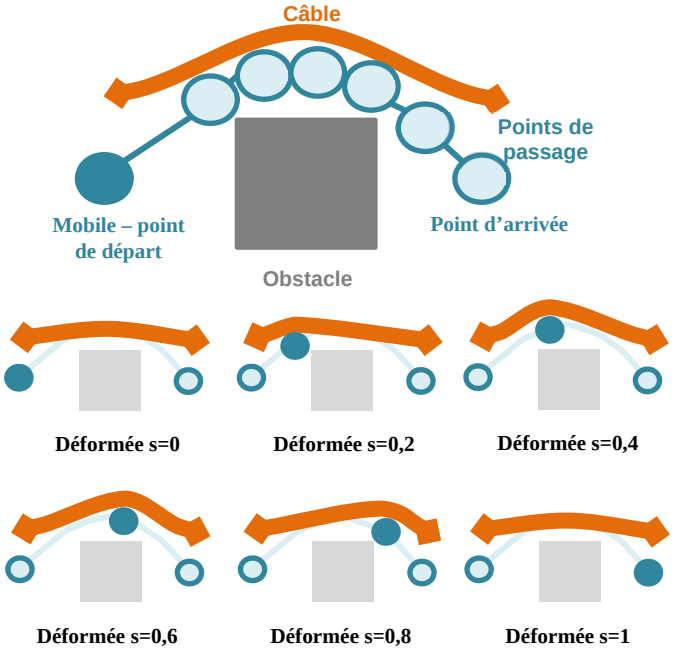

FIGURE 2: EXEMPLE DE DEFORMATION

On dispose d'une trajectoire au cours de laquelle un mobile, rigide, déforme un objet flexible : la succession des déformations subies par le flexible est collectée comme un mouvement à un degré de liberté.

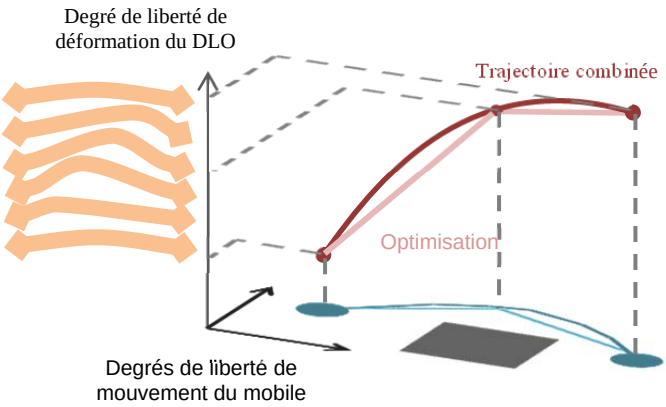

FIGURE 3: ESPACE DEPLACEMENT-DEFORMATION

Ce degré de liberté est ajouté au système formé par le mobile, on obtient alors un système combiné entre le déplacement et la déformation. La trajectoire initiale du mobile se transforme alors en un chemin de déplacement-déformation dans son espace de configuration augmenté (figure 3).
Les algorithmes d'optimisation de chemin classiques peuvent alors être employés pour optimiser ce chemin, c'est-à-dire le raccourcir tout en évitant les collisions.

Bien sûr, cet algorithme s'appuie sur une heuristique pragmatique : en formant l'espace de configuration combiné par le déplacement et la déformation, on découple les deux mouvements (déplacement et déformation). Pour minimiser la distance du mobile on se base sur les déformations que subit l'obstacle déformable lors du mouvement original. Les déformations de la trajectoire minimisée ne seront donc pas les mêmes qu'initialement. L'idée de l'algorithme est de conserver l'information obtenue au cours de l'étape précédente de recherche de trajectoire (la déformation du flexible pour laisser le passage au mobile) sans toutefois refaire appel à des simulations de déformation qui sont trop coûteuses en temps de calcul pour être utilisées à chaque tentative de minimisation. Il faut donc bien évidemment tester la trajectoire obtenue par cette méthode en faisant appel à une véritable simulation des déformations, pour s'assurer que celles qu'elle engendre sont effectivement admissibles.
L'algorithme a été intégré dans les outils de Siemens-Kineo et testé sur des cas utilisateurs. Le nombre de nœuds de la trajectoire solution se réduit d'un facteur 10 pour des temps réalistes (quelques minutes). Ce temps d'optimisation se répartit à peu près uniformément entre la phase de discrétisation, la phase d'optimisation et la validation finale.

IV. MOBILE DEFORMABLE DANS UN ENVIRONNEMENT RIGIDE

L'objectif est de planifier automatiquement la trajectoire pour un objet mobile déformable de type câble que nous noterons DLO (Deformable Linear Object) dans un environnement constitué d'objets rigides. Nous considérons que la déformation du DLO [11, 12, 13,14] doit respecter des contraintes physiques réalistes ce qui implique aussi un nombre de nœuds de modélisation élevé (de l'ordre de la centaine). La difficulté de planifier la trajectoire d'un DLO provient de la dimension du système. L'extension des méthodes classiques de planification n'est pas triviale car, à la différence des objets rigides, l'espace des configurations décrivant le système est de dimension infinie et ajouter des degrés de liberté (d.d.l.) conduit souvent à explorer un espace de haute dimension avec une paramétrisation du DLO inefficace. Dans un premier temps nous avons intégré directement l'utilisation du simulateur physique XDE lors des étapes d'extension du graphe au sein d'une approche probabiliste type RRT. Cette approche s'est montrée trop couteuse à cause du temps de simulation du DLO à chaque appel d'extension du RRT lors du calcul des chemins locaux.

Notre idée est alors d'utiliser un autre modèle de déformation de DLO afin de trouver une solution au problème de la planification de trajectoire pour ensuite guider le simulateur XDE dans une phase de vérification.

A. *Planification sans contact*

L'utilisation des résultats récents de [15] permet de calculer rapidement une solution analytique des configurations statique sans collision ni contact (espace libre) pour un DLO de type Kirchoff. L'expression de ces configurations statiquement stables dans l'espace libre est obtenue par résolution d'un problème de contrôle optimal. Le point intéressant pour la planification de trajectoire est qu'il est suffisant d'explorer un espace de dimension 6, correspondant aux forces et couples appliqués à la base du DLO, pour décrire l'ensemble des configurations stables du DLO ce qui correspond à calculer le Modèle Géométrique Direct (MGD). Connaissant les 6 paramètres a_i à la base du DLO, il suffit d'intégrer numériquement la solution sur l'ensemble des nœuds du DLO pour obtenir le MGD. Les paramètres a_i définissent donc l'espace des configurations dans lequel nous devons planifier la recherche de trajectoire, ce qui revient à planifier dans l'espace des déformations.
Les résultats de simulation (implémentation en C++) de ce résultat montrent l'efficacité du calcul de ce MGD. Le seul inconvénient est que ce calcul a une complexité linéaire en nombre de nœuds du DLO. Comme l'appel à ce calcul va être très fréquent lors de l'extension du graphe RRT il faut pourvoir en diminuer le coût.
Nous avons montré [16] qu'il était possible d'utiliser la jacobienne de la géométrie du DLO pour approximer le MGD. Cette approximation nous permet d'avoir un gain de l'ordre de 10 en temps de calcul sur le MGD (figure 4).

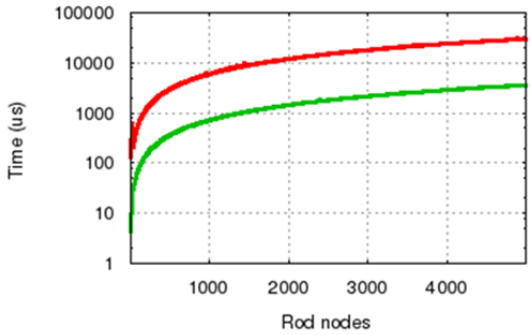

FIGURE 4 : GAIN POUR LE CALCUL DU MGD AVEC (VERT) ET SANS (ROUGE) L'UTILISATION DE L'APPROXIMATION DE LA JACOBIENNE

L'inconvénient de cette approximation est que le chemin local calculé lors de l'extension du RRT ne garantit plus que la solution globale soit libre de collision car l'approximation de la géométrie du DLO peut induire une non détection de collision. Aussi nous devons vérifier que :
- l'erreur d'approximation est faible.
- l'algorithme de planification détecte et corrige efficacement ces chemins locaux invalides générés par l'approximation.

1. Erreur d'approximation

Comme nous voulons limiter les appels au calcul du MGD exact il est nécessaire de pouvoir utiliser au minimum le calcul exact. En utilisant une métrique correspondant à l'erreur

maximale sur l'ensemble des nœuds du DLO et en bornant l'erreur maximale voulue en distance il est possible de calculer les variations maximales dans l'espace des configurations [16]. Nous en déduisons ainsi la taille du voisinage pour lequel l'approximation du MGD est valide.

2. Algorithme de planification

Nous proposons une extension du RRT, Fast Forward Geometry-RRT (FFG-RRT) pour prendre en compte ce calcul approché du MGD. Notre algorithmique, similaire à celle d'un Lazy-RRT [17], utilise à la fois les modèles exact et approché du MGD pour calculer les chemins locaux qui permettent d'étendre la roadmap du RRT. Nous limitons les appels au MGD exact en utilisant le pas maximal d'interpolation le long d'un chemin local qui est directement lié à la taille du voisinage valide de l'approximation.

Lorsqu'un chemin solution est trouvé, il est nécessaire de tester si cette solution est valide sans collision, puisque nous utilisons une approximation du MGD. S'il y a collision, il est nécessaire d'invalider le chemin local qui provoque la collision est de continuer à propager le graphe jusqu'à trouver un chemin global solution.

Cette méthode n'est efficace que si les appels de « re-planification », suite à des collisions dans le chemin global solution, sont peu nombreux. La non-collision du chemin global solution est directement liée à la qualité de notre approximation du MGD et donc cette approche ne peut fonctionner qu'avec une bonne approximation du MGD.

Les différents tests effectués montrent que le chemin global solution approximé est rarement en collision et confirme bien la qualité de notre approximation.

Résultats :

Nous avons dans un premier temps validé nos résultats sur des cas de test académiques (Figure 5) puis industriels (Figure 6). La configuration q_i est représentée en vert et q_b en rouge.

Nous constatons que notre approche par approximation de la jacobienne permet de résoudre les problèmes avec un gain de l'ordre de trois. Le très faible nombre de chemin global solution en collision confirme la qualité de l'approximation. Une étude détaillée de ces cas montre que l'invalidation des chemins locaux ne provient pas des collisions mais de problème d'instabilité des configurations qui ne respectent pas les conditions suffisantes d'équilibre statique.

Tous les tests sont effectués sur un PC classique et sont résolus en quelques minutes pour les cas industriels les plus complexes (figure 6). Vidéo accessibles sur http://projects.laas.fr/gepetto/index.php/Members/OlivierRoussel

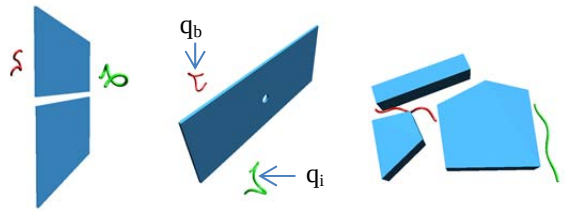

FIGURE 5: EXEMPLES DE CAS DE TESTS

En conclusion cette méthode apporte une solution mais en considérant seulement des configurations statiques. Elle offre l'avantage de pouvoir s'intégrer aux outils de Siemens-Kineo dans différents algorithmes de planification avec de bonnes performances dès que le calcul du MGD devient critique.

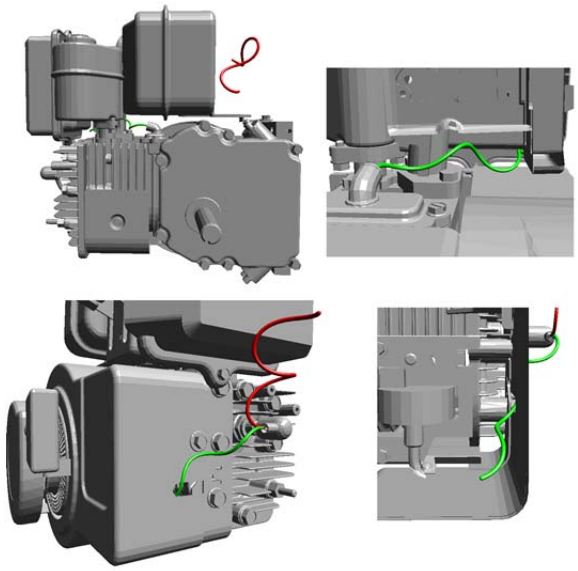

FIGURE 6: EXEMPLES DE DESASSEMBLAGE INDUSTRIEL. LES FIGURES DU HAUT CONSIDERENT LE CAS DU DLO LIBRE, CELLES DU BAS CELUI OU ON DOIT EXTRAIRE LE DLO AVEC UNE EXTREMITE FIXE. LES FIGURES DE DROITES MONTRENT LA CONFIGURATION INITIALE DU DLO DANS CHAQUE CAS.

Tout comme la planification d'objet rigide parmi des objets déformables ne fait appel à XDE que lorsqu'une collision avec un objet déformable est détectée afin d'en limiter les appels, nous pouvons utiliser la solution obtenue par notre algorithme comme une entrée du simulateur XDE afin d'en limiter les appels. Pour cela il suffit de choisir convenablement les paramètres d'élasticité pour faire correspondre les deux modèles. Si la simulation XDE invalide la solution il est nécessaire de planifier une nouvelle trajectoire ce qui accroît notablement les temps de calcul et nous a amené à considérer la prise en compte du contact.

B. Planification avec contact

Vouloir planifier dans l'espace des configurations libres en couplant une approche RRT avec XDE est trop couteux pour une utilisation autre que sur des cas simples. Une étude plus fine de cette approche montre que vouloir restreindre la planification seulement à l'espace libre de collision pénalise fortement les étapes d'extension du graphe du RRT lors de la planification locale avec XDE. La principale raison est que lorsqu'on manipule le DLO, par une extrémité par exemple, l'ensemble du DLO bouge ce qui implique une très grande probabilité de collision avec l'environnement. Par conséquence, l'extension par la méthode locale est réduite et le graphe du RRT se diffuse de manière très lente ce qui induit un nombre élevé d'appel à XDE, appels qui sont très couteux.

Si on regarde un opérateur humain qui manipule un DLO dans un environnement encombré, il va se servir des contacts pour faire glisser le DLO et guider sa progression en

ressentant les efforts qu'il exerce sur les extrémités du DLO. Aussi vouloir planifier la trajectoire d'un DLO sans contact semble peu réaliste. Dans ce contexte, il est nécessaire de prendre en compte les glissements du DLO le long des objets donc d'utiliser les contacts. Pour cela nous ne pouvons plus utiliser un modèle de configuration statique pour planifier une trajectoire, une simulation dynamique est indispensable ce qui implique de faire de la planification kinodynamique [18].
Nous nous plaçons donc maintenant dans le contexte suivant :

- Prise en contact des contacts DLO/environnement
- Simulation dynamique du DLO
- Manipulation du DLO par une de ses extrémités

La prise en compte des contacts est rarement intégrée dans les algorithmes de *MP* car il faut à fois détecter le contact très rapidement (calcul de distance exacte à la place du test binaire de collision) et ensuite calculer les efforts de contact en tenant compte des lois de Signorini (non interpénétration, unilatéralité de la force de contact et complémentarité entre la distance et la force de contact) et de Coulomb (cône de friction fonction des matériaux). XDE détecte les contacts lors d'une simulation et retourne les points et les forces de contact. Nous avons donc débuté ce travail en ne prenant en compte que des contacts glissants.
Faire de la planification kinodynamique impose de travailler dans l'espace d'état X et non dans l'espace des configurations Q. Pour un DLO décrit par n nœuds, un état X est décrit par les 12.n paramètres (situations et vitesses des nœuds),
$X = (q_1,\ \dot{q}_1, \ldots, q_n,\ \dot{q}_n)$
L'évolution de l'état du DLO X dépend des actions/commandes U appliqués au DLO, $X_{(k+1)}=f(X_k, U_k)$.
Trouver une solution à notre problème revient donc à trouver une séquence de contrôles { U_k } qui amène le DLO à l'état final. Les algorithmes de planification kinodynamique planifient alors directement dans l'espace des contrôles en respectant les contraintes dynamiques du système pour construire un graphe RRT dans l'espace des configurations. Etendre le graphe dans Q alors que le tirage aléatoire se fait dans l'espace des commandes ne donne pas toujours de bons résultats et cette approche est très sensible à plusieurs paramètres (choix de la métrique par exemple).
Aussi à la place de travailler dans l'espace des commandes nous choisissons d'utiliser les fonctionnalités d'interaction entre les objets qui sont fournis par XDE. Nous définissions des objets aux deux extrémités du DLO (appelée *gripper*) qui sont les prises de l'objet et nous remplaçons le contrôle U_k par une situation à atteindre par un *gripper*. Nous ajoutons un degré de liberté discret permettant de sélectionner un des deux *gripper* du DLO. Notre nouvel espace de contrôle est donc de dimension 7. Nous avons couplé cette méthode de planification avec le moteur physique XDE afin de récupérer les points et forces de contact. Le contrôle appliqué U_k représente un effort externe appliqué sur un des *grippers*. Le pseudo code de notre algorithme peut se résumer ainsi :

- générer aléatoirement un état pour un des nœuds extrémité
- chercher le plus proche voisin dans l'arbre d'état
- choisir le *gripper*
- initialiser le simulateur
- faire appel au simulateur XDE
- mettre à jour l'arbre

La métrique utilisé pour la fonction de voisinage est basée sur la distance euclidienne sur la position du *gripper* considéré. Le simulateur XDE utilise les erreurs en position et vitesse sur l'état courant du *gripper* comme entrée. Afin de ne pas tomber dans un minimum local, la simulation ne s'effectue que sur un temps limité.

Résultats :
Nos premiers résultats sur un cas simple (figure 7) montrent que les temps de planifications sont beaucoup plus rapides que dans une approche en espace libre.

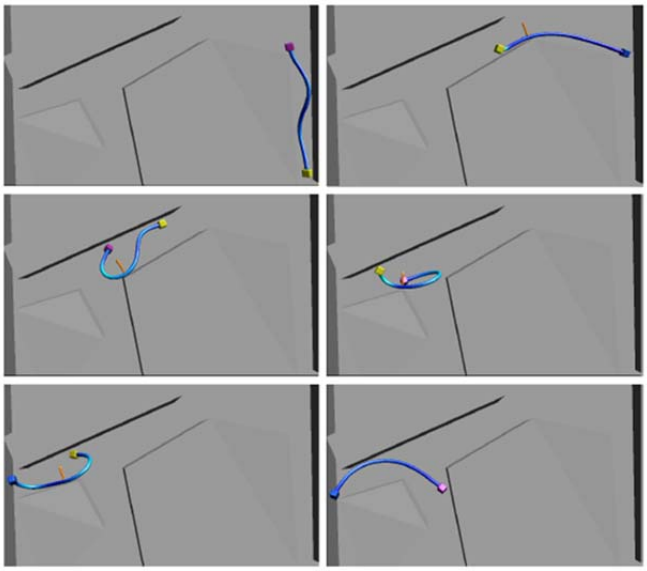

FIGURE 7: EXEMPLE DE TRAJECTOIRE SOLUTION AVEC CONTACT

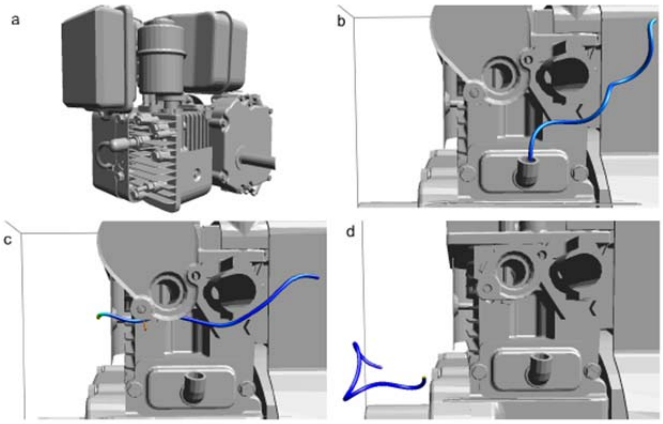

FIGURE 8 : TRAJECTOIRE SOLUTION DANS LE CAS D'UN MOTEUR (DE B A D). AFIN DE VISUALISER LE DLO DANS LE MOTEUR, DES PLAN DE COUPE SONT EFFECTUES SUR LES FIGURES B, C ET D.

Nous avons repris le même cas d'étude industriel (figure 8) et nous obtenons une solution satisfaisante dans un temps réaliste (quelques minutes).

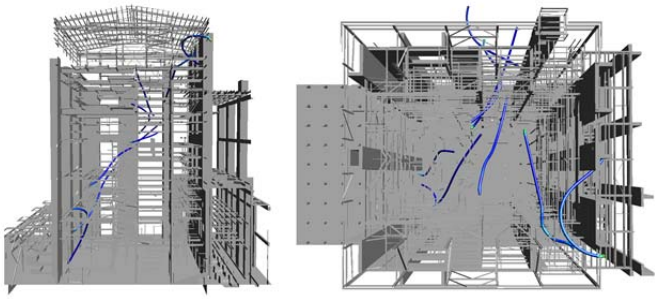

FIGURE 9 : TRAJECTOIRES SOLUTIONS EN UTILISANT LE CONTACT

V. CONCLUSIONS ET FUTURS TRAVAUX[1]

Dans ce papier nous montrons qu'il est possible de résoudre le problème de *MP* en considérant que soit le mobile ou l'environnement est déformable. La principale difficulté consiste à trouver le bon compromis entre précision du modèle, qui est utilisé lors de chaque appel au simulateur physique et temps de calcul au sein des algorithmes de *MP*. Dans le cas du mobile rigide dans un environnement déformable nous résolvons le problème en limitant les appels au simulateur physique XDE. Dans le cas de câble parmi des rigides nous utilisons une autre modélisation physique de câble permettant d'avoir une première solution qu'il convient dans un deuxième temps d'affiner en utilisant XDE au moyen de petites perturbations.

Suite à ces différents résultats, il apparait que la planification kinodynamique peut apporter des résultats satisfaisants. L'idée est de développer une méthode qui utilise à la fois de la planification dans l'espace des configurations pour utiliser le calcul rapide des configurations statique dans l'espace libre et qui utilise la planification kinodynamique seulement pour les extensions du graphe RRT. Ainsi nous planifions à la fois dans l'espace d'état du câble et aussi dans son espace des configurations. En considérant que le câble est manipulé par ses extrémités et qu'il peut glisser le long des obstacles rigides, nous planifions la séquence d'efforts à appliquer. Les erreurs en position et vitesse sur l'état courant des extrémités du câble sont les entrées pour XDE qui fournit en sortie le chemin local qui permet d'étendre le graphe. Les premiers résultats encourageants (voir figure 9) doivent être consolidés par des benchmarks plus nombreux

La prise en compte des efforts de contacts fournis par XDE est une piste prometteuse qui sera étudiée dans le futur. Pour cela il faudra utiliser les forces de contacts pour planifier non plus seulement dans l'espace libre mais dans l'espace admissible (libre et contact). L'intégration du modèle complet de contact au sein de XDE est nécessaire car nous voudrions prendre en compte dans ce travail les efforts exercés au niveau du DLO afin de ne planifier que des trajectoires réalisables. Lorsque nous aurons des résultats probants, il faudra coupler les deux approches pour considérer à la fois le mobile et l'environnement comme des déformables.

[1] Ce travail est financé par l'ANR dans le projet FLECTO

VI. REFERENCES

[1] S. M. LAVALLE. "Planning Algorithms". Cambridge University Press, Cambridge, U.K., 2006.

[2] O. B. BAYAZIT, J.-M. LIEN and N. M. AMATO. "Probabilistic roadmap motion planning for deformable objects". IEEE Int. Conf. on Robotics and Automation, pages 2126–2133. 2002.

[3] R. GAYLE, P. SEGARS, M. C. LIN, and D. MANOCHA. "Path planning for deformable robots in complex environments". Robotics: Science and Systems, pages 225–232. The MIT Press, 2005.

[4] I. KABUL, R. GAYLE and M. C. LIN. "Cable route planning in complex environments using constrained sampling". ACM Symposium on Solid and Physical Modeling, pages 395–402. 2007.

[5] F. LAMIRAUX and L. E. KAVRAKI. "Planning paths for elastic objects under manipulation constraints". Int. Jour. Robotic Research, 20(3):188–208, 2001.

[6] A. MAHONEY, J. BROSS and D. JOHNSON. "Deformable robot motion planning in a reduced-dimension configuration space". IEEE Int. Conf. on Robotics and Automation, pages 5133–5138. 2010.

[7] M. MOLL and L. E. KAVRAKI. "Path planning for deformable linear objects". IEEE Trans. on Robotics, 22:625–636, 2006.

[8] S. RODRIGUEZ, J.M. LIEN and N. M. AMATO. "Planning motion in completely deformable environments". IEEE Int. Conf. on Robotics and Automation, pages 2466–2471. 2006.

[9] S. M. LAVALLE and J. J. KUFFNER. "Rapidly-exploring random trees : Progress and prospects". Workshop on the Algorithmic Foundations of Robotics (WAFR'00), 2000.

[10] X. MERLHIOT, J. Le GARREC, G. SAUPIN and C. ANDRIOT. "The xde mechanical kernel: Efficient and robust simulation of multibody dynamics with intermittent nonsmooth contacts". Second Joint Int. Conf. on Multibody System Dynamics. 2012.

[11] S. S ANTMAN. "Kirchhoff's problem for nonlinearly elastic rods(for axial extension and cross section shear)". Quarterly of applied mathematics, 32:221–240, 1974.

[12] H. WAKAMATSU and S. HIRAI. "Static modeling of linear object deformation based on differential geometry". Int. Jour. Robotic Research. 23(3):293–311, 2004.

[13] J.C. SIMO. "A finite strain beam formulation. the three-dimensional dynamic problem. part i. Computer methods". Applied mechanics and engineering, 49(1):55–70, 1985.

[14] M. BERGOU, M. WARDETZKY, S. ROBINSON, B. AUDOLY and E. GRINSPUN. "Discrete Elastic Rods". ACM, Transactions on Graphics, 27(3):63:1–12, 2008.

[15] T. BRETL and Z. McCARTHY. "Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations". Int. Jour. Robotic Research. 33(1):48–68, 2014.

[16] O. ROUSSEL, M. TAÏX, and T. BRETL. "Efficient motion planning for quasi-static elastic rods using geometry neighborhood approximation". IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics. 2014.

[17] R. BOHLIN and L. E. KAVRAKI. "Path planning using lazy prm". IEEE Int. Conf. on Robotics and Automation, pages 521–528. 2000.

[18] S. M. LAVALLE and J. J. KUFFNER. "Randomized kinodynamic planning". Int. Jour. of Robotics Research, 20(5):378–40. 2001.