



# Optimisation de séquence de tâches avec lissage des mouvements

François Keith, Nicolas Mansard, Sylvain Miossec, Abderrahmane Kheddar

► **To cite this version:**

François Keith, Nicolas Mansard, Sylvain Miossec, Abderrahmane Kheddar. Optimisation de séquence de tâches avec lissage des mouvements. JRNH'09: Journées Nationales de la Robotique Humanoïde, May 2009, Nice, France. 2009, <<http://jnrh09.irccyn.ec-nantes.fr/>>. <lirmm-00798824>

**HAL Id: lirmm-00798824**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00798824>**

Submitted on 10 Mar 2013

**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.

# Optimisation de séquence de tâches avec lissage des mouvements

François Keith, Nicolas Mansard, Sylvain Miossec, Abderrahmane Kheddar

**Abstract**—Plutôt que d'utiliser le trio classique planification symbolique - raffinement par planification de trajectoire - suivi, nous proposons d'utiliser le cadre de la fonction de tâches pour unifier le raisonnement et l'exécution. Nous utilisons alors un procédé d'optimisation couplé à une expérimentation complète pour concrétiser le plan de tâches et tirer parti de toute la redondance du robot. Les résultats sont illustrés par une simulation sur le robot Hrp-2.

**Index Terms**—Optimisation, Contrôle

## I. INTRODUCTION

Lorsque l'environnement est entièrement ou partiellement structuré, la plupart des missions robotiques peuvent être décomposées hiérarchiquement. Chacune des opérations est alors décomposée en un ensemble d'actions de base (*i.e.* fonctions sensori motrices) et doit ensuite être exécutée par les lois de commande du robot.

Plusieurs méthodes ont été proposées pour calculer de telles séquences à partir d'une mission donnée et d'un ensemble de paradigmes [1], [3]. Ils produisent généralement un plan symbolique, où les seules précisions numériques correspondent aux données temporelles du plan de tâches. L'application du plan par le robot nécessite de préciser la séquence temporelle, typiquement grâce à un planificateur de trajectoire [5]. Ces méthodes basées sur la trajectoire sont peu robustes aux changements ou aux incertitudes de l'environnement et peuvent nécessiter de recalculer la trajectoire plusieurs fois pour pouvoir réaliser la mission. Cependant la signification du plan symbolique est perdue lors du passage à la trajectoire globale et il est donc difficile (et souvent spécifiquement "hard codé") d'augmenter les trajectoires avec les infos symboliques qui permettraient d'en diminuer le coût de recalcul.

D'un autre côté, une tâche (*i.e.* une fonction de tâche [8]) peut être directement reliée aux symboles sur lesquels le plan de tâches est défini (*i.e.* atteindre un objet à saisir est une tâche qui nécessite que le bras du robot soit disponible, et dont la post-condition est d'avoir la main sur l'objet - c'est aussi décrit par une fonction sensori motrice directement applicable par exemple par cinématique inverse [6]). La mission ainsi décomposée est exécutable par une boucle fermée, qui garantit une bonne robustesse et évite le calcul explicite de trajectoire non nécessaire.

Travailler avec des tâches exclusives aboutit en général à des mouvements saccadés. D'un autre côté, produire un

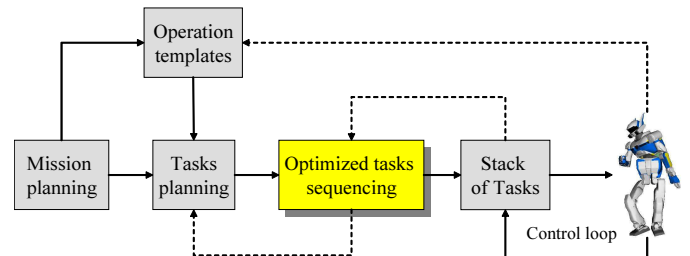


Fig. 1. Représentation schématique des différents composants nécessaires à la réalisation d'une tâche robotique autonome.

plan avec superposition des tâches se heurte rapidement aux limites physiques du robot (par exemple, aux obstacles ou aux contraintes dynamiques d'un humanoïde). Un tel problème n'ayant pas une forme discrète, les techniques fondées sur du raisonnement symbolique sont difficiles à appliquer directement. Des techniques d'optimisation semi-infinie [7] peuvent être utilisées pour générer des trajectoires bas niveau pour l'entière solution, en tenant compte de ces contraintes. Mais on retombe alors dans les problèmes posés par le calcul explicite de trajectoire (en particulier, manque de réactivité).

Nous proposons d'utiliser la notion de tâche à la fois pour le raisonnement symbolique et pour le contrôle du robot. Entre ces deux modules (Fig. 1), on utilise une optimisation semi-infinie afin de préciser le plan de tâche symbolique et de prendre en compte les contraintes numériques. Étant donné un ensemble de tâches élémentaires permettant de réaliser la mission, notre système retourne pour chaque tâche ses temps optimaux de début et de fin et ses paramètres optimaux d'exécution (*i.e.* ses gains). De plus, cette méthode réalise un lissage de mouvements (*i.e.* un enchaînement continu des tâches grâce à leur superposition).

## II. OPTIMISATION DE SÉQUENCE DE TÂCHES

Chaque tâche est décrite par trois paramètres temporels : les temps de début  $t^I$ , de fin  $t^F$  et de réalisation  $t^R$  - temps où l'erreur attachée à la tâche est considérée comme régulée. L'optimisation de la séquence de tâches est un problème défini par :

- un critère : le temps de fin de la séquence,
- des contraintes, liées aux tâches (cohérence dans l'ordre des tâches, réalisation des tâches) ou liées à l'environnement et en particulier au robot (limites articulaires, limites de vitesse articulaires, évitement des collisions),
- des paramètres : les temps  $t^I$  et  $t^F$  ainsi que les gains de chaque tâche.

F. Keith et A. Kheddar sont membres du CNRS-UM2 LIRMM, Montpellier, et du CNRS-AIST JRL, UMI3218/CRT, Tsukuba, Japon

N. Mansard est membre de Gepetto/Ria, CNRS-LAAS, Toulouse

S. Miossec est membre de SRI/PRISME, université d'Orléans, Bourges

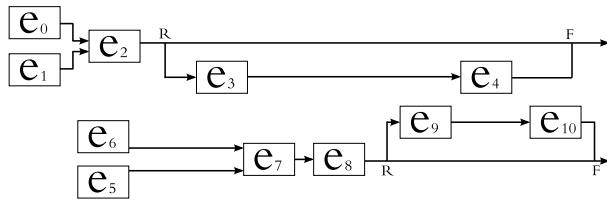


Fig. 2. Plan de tâches symbolique correspondant au Hrp-2 allant chercher une canette dans le frigo. Ce plan est le template formel (typiquement calculé par un logiciel de raisonnement) qui contraint la cohérence dans l'ordre de réalisation des tâches.

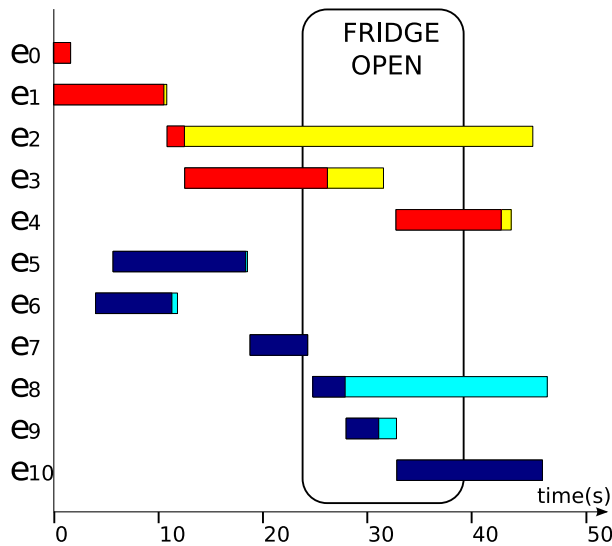


Fig. 3. Résultats de l'optimisation de la séquence de tâches. La partie sombre (rouge ou bleu marine) correspond à la phase de régulation de la tâche  $[t^I, t^R]$ . La partie claire (jaune ou bleu clair) correspond à la phase durant laquelle la tâche est régulée et conservée  $[t^R, t^F]$ .

Les détails du problème d'optimisation sont précisés dans [4].

### III. RÉSULTATS

La séquence de tâches suivante décrit le mouvement d'un robot sortant une canette d'un frigo. Les tâches sont :

- $e_0$  : Ouvrir la pince droite
- $e_1$  : Diriger la pince droite vers la poignée du frigo
- $e_2$  : Fermer la pince droite
- $e_3$  : Ouvrir le frigo
- $e_4$  : Fermer le frigo
- $e_5$  : Ouvrir la pince gauche
- $e_6$  : Rapprocher la pince gauche du frigo
- $e_7$  : Positionner la pince gauche sur la canette
- $e_8$  : Fermer la pince gauche
- $e_9$  : Soulever la canette
- $e_{10}$  : Sortir la canette du frigo

Cette séquence est centrée sur le frigo. Une façon d'assurer le bon déroulement de cette mission est de contraindre le mouvement du bras gauche à la période d'ouverture du frigo par l'ajout de contraintes temporelles. Cependant, afin de permettre une superposition des tâches, nous avons choisi de considérer la contrainte de collision entre le bras gauche et le frigo comme seule contrainte limitante.

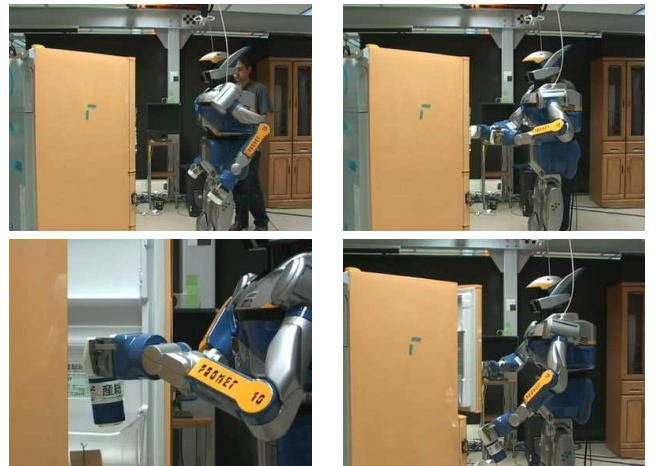


Fig. 4. Hrp-2 sortant une canette du frigo.

L'optimisation a été réalisée sur un PC tournant sous Windows, avec l'algorithme SQP de l'optimiseur MATLAB. La simulation a été réalisée avec le framework AMELIF[2]. La séquence trouvée est décrite sur la figure 3. Deux chevauchement de tâches sont observables : d'une part, le mouvement de la main gauche commence alors que le frigo n'est pas complètement ouvert et d'autre part, la fermeture de la porte commence alors que la canette n'est pas complètement sortie. La séquence de tâches a été réalisée en 47s. Sans la superposition des tâches, elle aurait duré 71s. Cette séquence de tâche a été exécutée par le robot humanoïde Hrp-2 (figure 4). La vidéo de cette séquence est disponible sur <http://www.laas.fr/~fkeith/JNRH09.avi>

### REFERENCES

- [1] R. Dechter. *Constraint Processing*, chapter 12, Temporal Constraint Network. Morgan Kaufmann, 2003.
- [2] P. Evrard, F. Keith, J.-R. Chardonnet, and A. Kheddar. Framework for haptic interaction with virtual avatars. In *Robot and Human Interactive Communication, RO-MAN 2008. IEEE International Symposium on Robot and Human Interactive Communication*, 2008.
- [3] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kauffmann Publishers, 2004.
- [4] F. Keith, N. Mansard, S. Miossec, and A. Kheddar. Time warping for optimal task sequencing - application to a humanoid robot. submitted to ICRA 2009.
- [5] S. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [6] N. Mansard, O. Stasse, F. Chaumette, and K. Yokoi. Visually-guided grasping while walking on a humanoid robot. In *IEEE Int. Conf. Robot. Autom. (ICRA'07)*, pages 3041-3047, Roma, Italia, Apr. 2007.
- [7] S. Miossec, K. Yokoi, and A. Kheddar. Development of a software for motion optimization of robots- application to the kick motion of the HRP-2 robot. In *IEEE International Conference on Robotics and Biomimetics*, Kunming, China, 2006. Robio2006.
- [8] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.