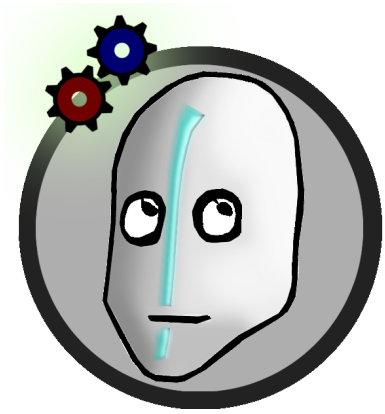


ACRONYME	ROBOERGOSUM
NOM DU PROJET	ROBOTS CONSCIENTS
REFERENCE	DECISION ANR-12-CORD-0030
NUMERO DE LA TACHE	T5
NOM DE LA TACHE	Self-Awareness and Social Awareness
NUMERO DU RAPPORT	D5.2.1
TITRE DU RAPPORT	Report on multiple system learning in human-robot interaction context
PARTENAIRES	ISIR, LAAS
DATE	T0+18

1. RESUME DU DELIVERABLE / SUMMARY

Il nous a semblé important, dans cette première phase du projet, de réfléchir à la manière dont systèmes des deux partenaires peuvent coopérer. En effet, le LAAS dispose d'une architecture robotique basé sur une architecture trois niveaux [1, 3] alors que l'ISIR propose un ensemble d'algorithmes d'apprentissage [5, 6] ne s'inscrivant pas dans ce type de modèle. La conception de ces systèmes est donc très différente, la question était donc de savoir comment les faire coopérer.
Nous avons eu plusieurs réunions de travail où les uns et les autres ont expliqué leur système et les expérimentations qui avaient déjà été réalisées avec chacun d'entre eux. A l'issue de ces réunions nous avons décidé qu'il serait intéressant d'explorer comment nos systèmes pouvaient être complémentaires à un niveau décisionnel, i.e. sur la manière de choisir la prochaine action à réaliser par le robot. D'autre part, nous avons choisi dès le départ de considérer une tâche coopérative mettant en jeu un homme et un robot. Nous sommes pour cela partis d'une tâche de manipulation d'objets pour laquelle la situation est décrite par l'image suivante :

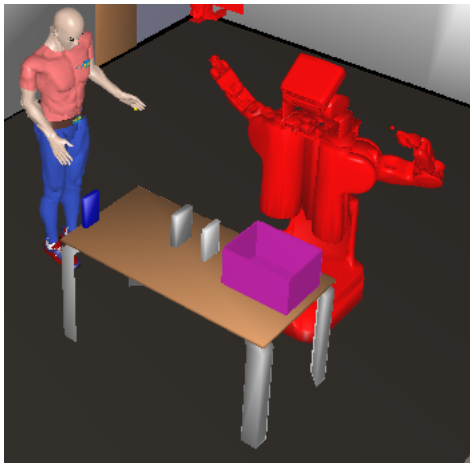


Figure 1 Tâche de manipulation d'objets.
Un ensemble de cassettes est sur la table, certaines sont accessibles à l'homme, d'autres au robot, certaines aux deux. L'homme et le robot doivent nettoyer la table et à la fin de la tâche, toutes les cassettes doivent être déposées dans une poubelle (accessible à l'un ou à l'autre des protagonistes).

Chaque équipe a fourni une brique logicielle capable de calculer la prochaine action à exécuter pour réaliser la tâche (les actions disponibles sont identiques) :

- HATP (pour le LAAS) [2] est un planificateur HTN, disposant d'un modèle du monde et capable de calculer un plan à partir de l'état courant fourni par le système (lorsque l'état initial le permet) comprenant l'ensemble des actions pour réaliser la tâche, cependant dans notre cas, HATP ne fournira que la première action du plan
- un algorithme d'apprentissage model-free (pour l'ISIR) [5] est capable également de proposer la prochaine action à exécuter

A chaque itération, le système a donc accès à deux actions possibles. Pour choisir entre ces deux actions, nous avons conçu un MetaContrôleur qui choisit entre les deux options à chaque pas d'exécution.

Notre intuition de départ était la suivante :

- HATP dispose de connaissances a priori, stockées dans son modèle qui lui permet de fournir un plan complet, il ne nécessite pas de phase d'apprentissage mais il se peut qu'il soit incapable de proposer une nouvelle action (si son modèle du monde lui indique qu'aucun plan ne permet de réaliser la tâche)
- l'algorithme d'apprentissage model-free nécessite une phase d'apprentissage qui peut être longue mais proposera toujours une action à réaliser

  => le MetaContrôleur pourrait guider l'apprentissage de l'algorithme model-free en se basant sur les actions proposées par HATP. Une fois cet apprentissage réalisé, l'algorithme model-free pourrait être privilégié par le MetaContrôleur.

Les expériences menées jusqu'à présent avec un robot en simulation (mais en faisant tourner les algorithmes d'exécution des actions) valident la connexion entre nos systèmes. Cependant, l'intégration actuelle montre également ses limites :

- Un travail de paramétrage des algorithmes et de la tâche doit être mené pour être pertinent vis-à-vis de la tâche.
- Le critère que nous avons appliqué au niveau du MetaContrôleur pour choisir la prochaine action est simple et ne permet pas d'exploiter toute l'information disponible sur le déroulement de la tâche. Un travail est actuellement en cours sur ce sujet.

Mais cela nous a également permis de mettre en exergue deux axes pour la poursuite de ces travaux:

1. Le premier axe s'inscrit dans la continuité du travail déjà mené, il nous paraît clair que l'apprentissage peut bénéficier des connaissances a priori que l'homme peut avoir sur le système (ici apporté par le système de planification HATP)
2. Le second axe consiste cette fois-ci à améliorer le planificateur HATP à l'aide de l'apprentissage. En effet, la planification HTN intègre des coûts qui permettent de choisir entre différentes solutions. Ces coûts sont aujourd'hui fixés au départ. Il nous paraît intéressant d'étudier comment l'apprentissage pourrait venir mettre à jour ces coûts en fonction des résultats des actions.

Nous détaillons les expérimentations effectuées dans la suite du deliverable. Nous spécifions tout d'abord la tâche, expliquons ensuite l'intégration effectuée pour ensuite parler des résultats et finir sur les perspectives.

2. SPECIFICATION DE LA TACHE

Comme représenté Figure 1 Tâche de manipulation d'objets, le robot et un opérateur sont autour d'une table. Sur la table, il y a trois cassettes. Une poubelle est aussi présente dans l'environnement. Le but de la tâche est de nettoyer la table, c'est-à-dire de ranger toutes les cassettes dans la poubelle. Certains cassettes ne sont atteignables que par l'opérateur, les autres uniquement par le robot, il est donc nécessaire pour les deux agents d'interagir afin d'accomplir la tâche comme on peut le voir présenté Figure 2 Description de l'environnement.

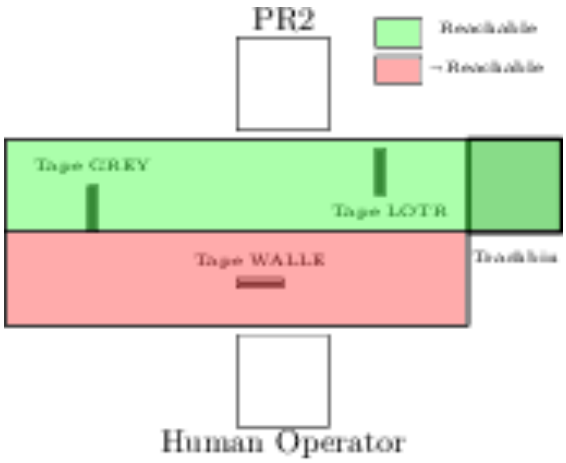


Figure 2 Description de l'environnement

Le robot manipule l'état construit à partir des faits **isReachable**, **isIn Trashbin**, **isOn Table**, **isVisible** calculés sur la base de la position géométrique de chacun des éléments et rendus disponibles par l'intermédiaire du module Spatial Reasoning and Knowledge (SPARK [4]) du LAAS. Ces faits sont vrais ou faux. Nous sommes théoriquement confrontés à 2^{13} états, mais certains faits sont mutuellement exclusifs, ce qui réduit la taille de l'espace d'état potentiel.

L'état du monde construit par le robot est le suivant :

	isReachable	**isIn Trashbin**	**isOnTable**	**isVisible**
Trashbin	C0	-	-	-
LOTR_TAPE	C1	C4	C7	C10
WALLE_TAPE	C2	C5	C8	C11
GREY_TAPE	C3	C6	C9	C12

L'ensemble des actions disponibles pour réaliser la tâche sont les suivantes :
- TakeObject(Agent A, Object O, Table T) : L'Agent A prend l'Objet O sur la Table T
- ThrowObject(Agent A, Object O, Trashbin T): L'Agent A jette l'Objet O dans la poubelle T
- GiveObject(Agent A, Object O, Agent B): L'Agent A donne l'Objet O à l'Agent B (échange « main » à « main »)
- Escape(Agent A): L'Agent A se déplace légèrement (pour réduire ses contraintes géométriques).
- Wait(Agent A): L'Agent A attend pendant une certaine durée.

Si HATP manipule des actions et leurs arguments, l'Expert habituel manipule des actions indépendamment de leur nature, mais sans variabilité. Afin que nos deux modèles puissent communiquer, nous définissons la correspondance suivante afin de permettre la traduction des actions entre les différents modules :

A0	TakeObject(PR2, GREY_TAPE, TABLE)
A1	TakeObject(PR2, LOTR_TAPE, TABLE)
A2	TakeObject(PR2, WALLE_TAPE, TABLE)
A3	ThrowObject(PR2, GREY_TAPE, Trashbin)
A4	ThrowObject(PR2, LOTR_TAPE, Trashbin)
A5	ThrowObject(PR2, WALLE_TAPE, Trashbin)

A6	GiveObject(PR2, GREY_TAPE, HUMAN)
A7	GiveObject(PR2, LOTR_TAPE, HUMAN)
A8	GiveObject(PR2, WALLE_TAPE, HUMAN)
A9	GiveObject(HUMAN, GREY_TAPE, PR2)
A10	GiveObject(HUMAN, LOTR_TAPE, PR2)
A11	GiveObject(HUMAN, WALLE_TAPE, PR2)
A12	Escape(PR2, TABLE)
A13	Wait(PR2)

Afin de permettre l'apprentissage de la tâche par l'Expert habituel, nous définissons la Fonction de Récompense qui décrit la tâche. Dans un premier temps, nous nous concentrons sur la tâche « Nettoyer la table » sans donner d'importance à l'ordre dans lequel les cassettes sont rangées. La tâche est considérée « finie » lorsque toutes les cassettes sont dans la poubelle et que le robot attend. Il reçoit alors une récompense R = 1.0. Chaque action a un coût faible devant la récompense (fixé ici à R=-0.05, mais nous pouvons envisager un coût lié par exemple à l'énergie nécessaire à accomplir une action, et dans un deuxième temps, ce coût dépendra directement de l'humain avec qui le robot interagit). Ces récompenses et coûts sont donnés par une source externe qui évalue le comportement du robot.

Nous envisageons trois cas particuliers pour lequel la complémentarité de notre système pourrait s'avérer pertinente :

- **Cas 1 :** *l'environnement ne change pas*, l'espace d'état permet au planificateur HATP de trouver un plan valide. Dans ce cas, HATP est l'Expert qui est le mieux à même de résoudre la tâche, puisque sa planification lui permet de trouver un chemin vers l'état but en exploitant ses connaissances. Le MetaContrôleur doit donc le sélectionner préférentiellement lors des premières exécutions de la tâche. L'Expert habituel apprend en parallèle des actions choisies par HATP, et prend la main lorsque son apprentissage est suffisant pour lui faire choisir les bonnes actions à un coût moindre.
- **Cas 2** : *l'environnement change*, HATP est capable de trouver un plan. On se place après apprentissage d'une habitude comportementale par l'Expert habituel, et on change les conditions dans lesquelles le robot agit (i.e. la poubelle change de côté). La tâche est la même, mais l'habitude n'est plus pertinente dans ces conditions, le MetaContrôleur doit donc redonner la main à HATP pour retrouver un plan qui permette de résoudre la tâche. En conservant ces nouvelles conditions environnementales, l'Expert habituel peut réapprendre une habitude (avec les défauts de plasticité de cet algorithme, ou éventuellement avec stockage de l'habitude précédente et apprentissage à partir de zéro).
- **Cas 3** : *HATP ne peut trouver un plan*. En partant d'un système qui n'a pas encore appris, nous plaçons le robot dans un cas pour lequel HATP ne peut trouver de plan valide. Malgré l'échec de la planification, l'approche essai-erreur de l'Expert habituel permet au robot d'agir et au planificateur de trouver un état dans lequel il peut réussir à planifier.

3. INTEGRATION

Le but de cette expérimentation est de valider un exemple de complémentarité entre les deux systèmes : un système de planification de type HTN développé au LAAS (HATP), un système d'apprentissage par renforcement développé à l'ISIR (Expert Habituel). La Figure 3 Architecture de l'expérimentation montre les différents éléments que nous avons fait communiquer et leur provenance.

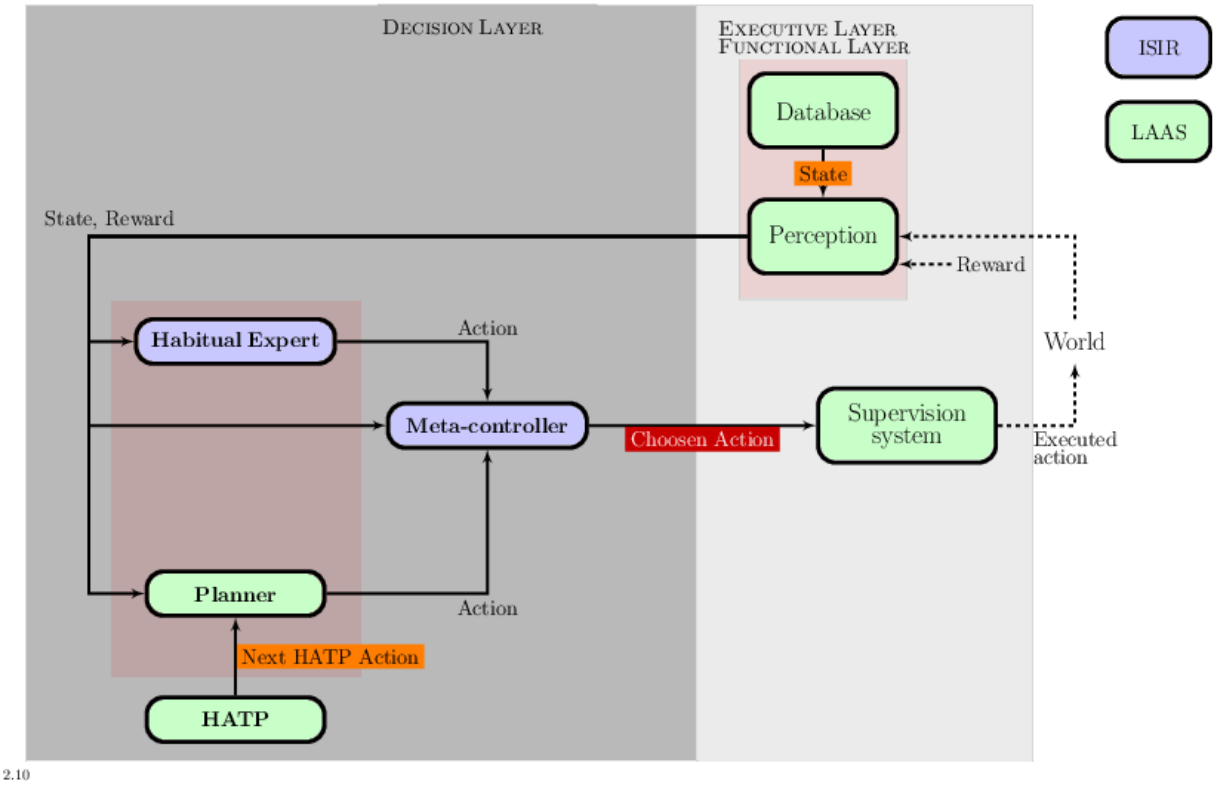


Figure 3 Architecture de l'expérimentation

Nos deux systèmes, Habitual Expert et Planner, envoient la prochaine action calculée au MetaContrôleur. Celui ci envoie l'action qu'il a sélectionnée au système de supervision qui exécute l'action. A l'issue de l'action, une récompense est calculée. Elle est fournie ainsi que le nouvel état du monde aux différents systèmes qui peuvent alors calculer une nouvelle action.

Cette tâche est répétée de manière cyclique afin que le robot puisse former une habitude, et possède une certaine variabilité, que nous pouvons exploiter pour vérifier que le robot reste capable de s'adapter à un changement de conditions tel que voulu dans l'architecture. De plus, la tâche implique un agent humain, dont le degré de coopérativité peut varier selon les cas que l'on souhaite étudier. La présence d'un individu averti du fonctionnement d'un robot permet de maîtriser les cas d'attribution de récompense, tandis qu'un individu naïf permettra d'étudier l'engagement entre l'humain et le robot.

L'ensemble du système a été implémenté et testé en simulation sur le robot PR2. La Figure 1 Tâche de manipulation d'objets montre une vue de l'environnement construit par le robot en simulation. Dans cet environnement, le robot calcule l'état du monde tel que nous l'avons défini et est capable de réaliser chacune des actions en faisant tourner les mêmes algorithmes que ceux actuellement implémenté sur le robot pour ce type de tâche de manipulation. Il nous semblait évident qu'il nous fallait tester notre système en premier lieu en simulation mais pour cette raison, nous avons bon espoir de pouvoir faire tourner cette expérimentation sur le robot réel assez rapidement.

4. RESULTATS

Sur la base de cette intégration, nous avons réalisé 7 expérimentations : les expériences 2 et 3 sont faites avec un critère – au niveau du MetaContrôleur – sur l'entropie de l'action, tandis que les expériences 4 à 8 utilisent un critère sur la récompense moyenne. Si les résultats qui suivent ne sont que préliminaires, ils valident l'intégration en commun des travaux des partenaires du projet. Certains ajustements restent nécessaires pour évaluer nos systèmes dans la tâche proposée, comme proposé ci-après.

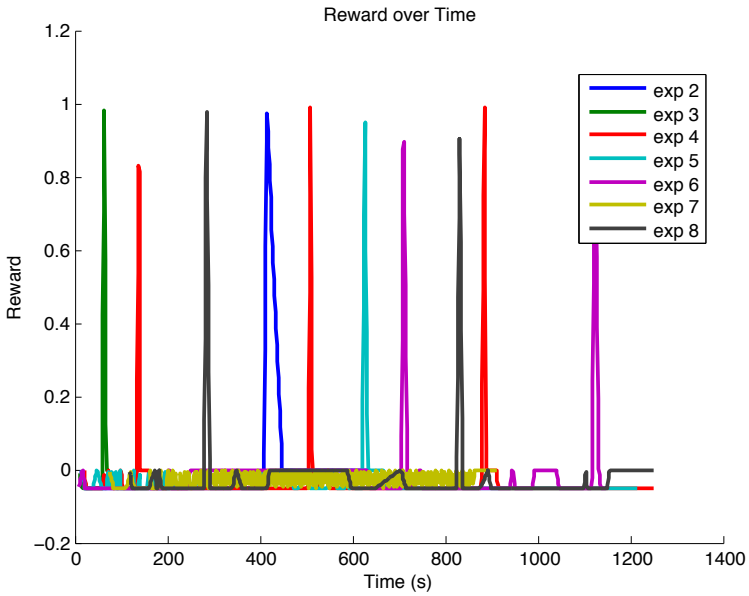


Figure 4 Récompense interpolée dans le temps

En termes de performance, c'est-à-dire de récompense obtenue au cours des expériences, on observe que les différentes configurations que nous avons testées parviennent à résoudre la tâche jusqu'à 3 fois (Figure 4 Récompense interpolée dans le temps, expérience 4, pics de récompense).

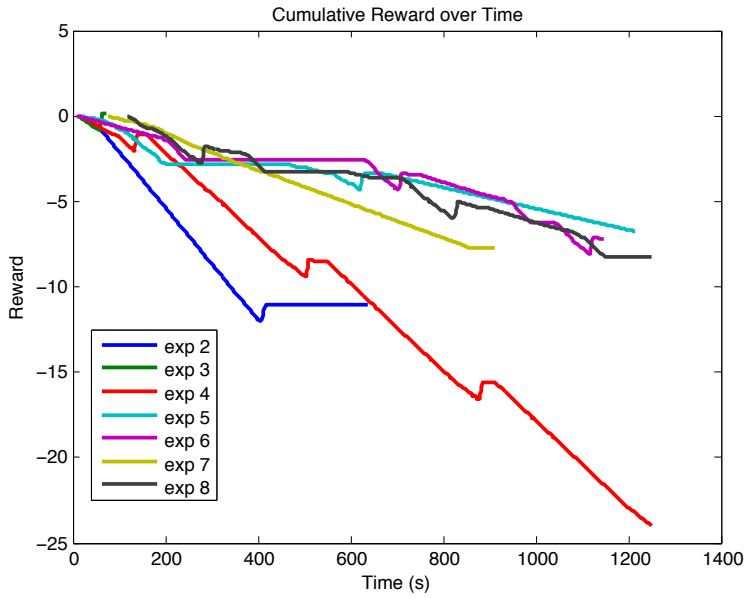
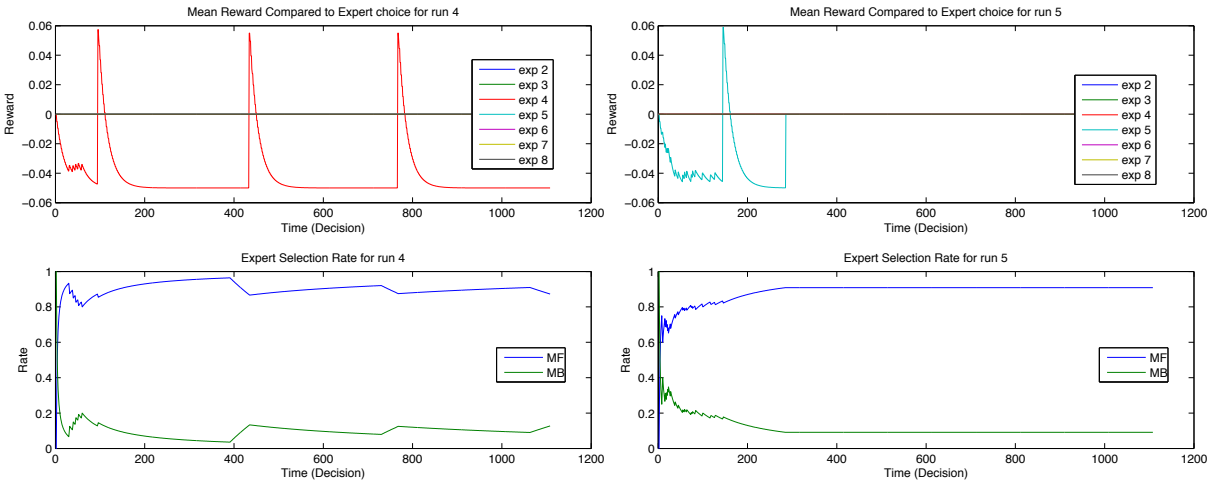


Figure 5 Récompense Cumulée dans le temps

Cependant, la récompense accumulée (cf Figure 5 Récompense Cumulée dans le temps) montre que notre choix de pénaliser les actions n'est pas judicieux car l'accomplissement de la tâche ne permet pas de compenser les coûts payés par le robot. Étant donné que définir une fonction de récompense pertinente est une tâche compliquée, il est prévu de revenir à une fonction plus simple qui ne présente que la récompense lorsque la tâche est accomplie, pour vérifier que ces limites ne sont dues qu'à une fonction mal dimensionnée dans la tâche.

Nous avons également cherché à évaluer la pertinence du critère de récompense moyenne dans ce cadre. Pour cela nous présentons sur la Figure 6 Mean Reward and Expert Selection for each run, la récompense et les experts selectionnés dans chacun des cas. MF (pour model-free) correspond à l'apprentissage tandis que MB (pour model based) correspond à HATP.



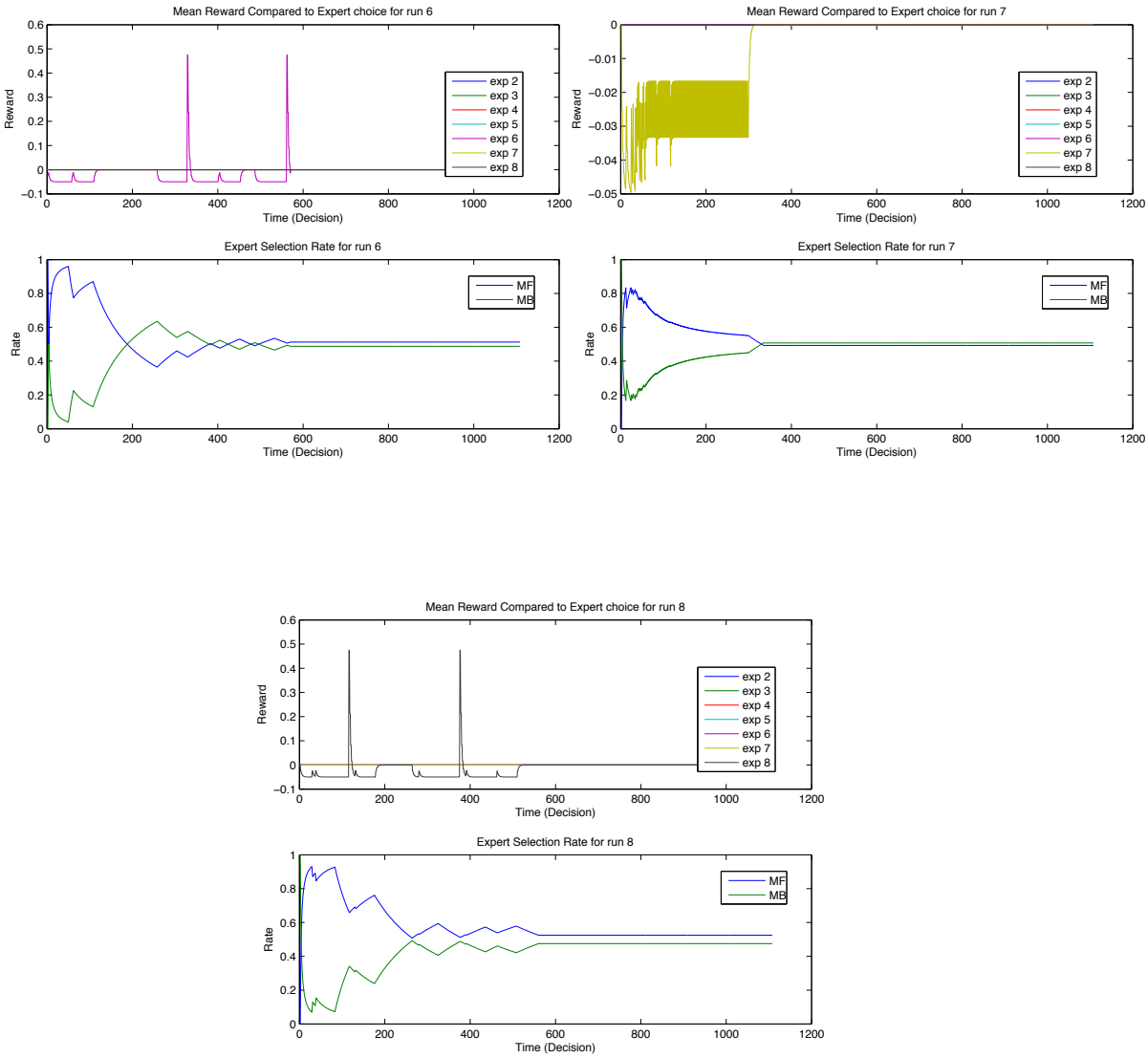


Figure 6 Mean Reward and Expert Selection for each run

Lorsque l'on met les pics de récompense reçue en parallèle de la proportion de sélection des Experts, on constate bien une augmentation de sélection de l'Expert habituel lorsque la récompense croît ou reste stable, ce qui valide le fonctionnement du critère. Cependant, ces figures montrent la limite du critère de récompense moyenne : l'Expert habituel est beaucoup sélectionné en début d'expérience, alors qu'il n'a pas encore appris. On ne bénéficie pas de HATP, dont le plan est a priori bien meilleur à ce moment. Il faut donc revoir ce critère afin de permettre un comportement qui exploite la force du planificateur en début d'apprentissage.

5. PERSPECTIVES

Vis-à-vis des éléments à mettre en œuvre dans la tâche 5 du projet, cette expérimentation nous permet d'avancer sur les points suivants :

- Vérifier que l'apprentissage d'habitudes est un principe général et utile pour un robot dans une grande variété de situations.
- Préparer l'étude de l'influence des interactions sociales au niveau de la décision sur le comportement du robot

En effet, même si les résultats montrent le besoin de retravailler et d'ajuster les différents paramètres de l'expérience, nous avons cependant :

- une architecture de contrôle capable d'apprendre des habitudes fonctionnelles, nous pensons en effet qu'il peut s'avérer important pour un homme interragissant avec un robot de retrouver des comportements plus ou moins similaires à chaque interaction
- une tâche pour évaluer notre architecture en interaction avec l'humain, qui offre un certain nombre de degrés de libertés qui permettront d'impliquer l'humain à différents niveaux (comme un professeur qui donne la récompense, pour étudier des cas variés d'apprentissage, comme une motivation où le robot cherche la satisfaction de l'humain, etc.)

Dans cette optique, nous envisageons deux axes de poursuites de ces travaux :

1. Le premier axe s'inscrit dans la continuité du travail déjà mené, il nous paraît clair que l'apprentissage peut bénéficier des connaissances à priori que l'homme peut avoir sur le système (ici apporté par le système de planification HATP), mais il faut que nous menions des expérimentations plus longues et que nous expérimentions en conditions « réelles » avec le robot réel et un homme
2. Le second axe consiste cette fois-ci à améliorer le planificateur HATP à l'aide de l'apprentissage. En effet, la planification HTN intègre des coûts qui permettent de choisir entre différentes solutions. Ces coûts sont aujourd'hui fixés au départ. Il nous paraît intéressant d'étudier comment l'apprentissage pourrait venir mettre à jour ces coûts en fonction des résultats des actions.

6. REFERENCES

[1] R. Alami, R. Chatila, S. Fleury, M. Ghallab, F. Ingrand, An architecture for autonomy, *International Journal of Robotic Research* (1998)

[2] S. Alili, V. Montreuil, R. Alami, *A Task Planner for an Autonomous Social Robot*, Distributed Autonomous Robotic Systems 8, 2009, pp 335-344

[3] M. Fiore, A. Clodic, R. Alami, On Planning and Task Achievement Modalities for Human-Robot Collaboration, *International Symposium on Experimental Robotics*, Marrakech/Essaouira, June 15–18, 2014

[4] G. Milliez, M. Warnier, A. Clodic, R. Alami, *A framework for endowing an interactive robot with reasoning capabilities about perspective-taking and belief management IEEE RO-MAN 2014*, Aug 2014, Edinburgh

[5] E. Renaudo, B. Girard, R. Chatila, M. Khamassi, *Design of a Control Architecture for Habit Learning in Robots*, Biomimetic and Biohybrid Systems - Third International Conference, Living Machines 2014, Milan, Italy, July 30 - August 1, 2014.

[6] R.S. Sutton, A.G. Barto, Reinforcement Learning I: Introduction. MIT Press, 1998.