

From motion effects to affordances: Bayesian learning of high-level actions

Stanislas Leroy

Master 2 parcours Intelligence Artificielle

2016-2017

Département Informatique

Encadrants : Stéphane Doncieux et Alexandre Coninx

Institut des Systèmes Intelligents et de Robotique

Université Claude Bernard Lyon 1, France

Mots-clés – Robotique développementale, affordance, clusterisation d'effets

Remerciements

Mon stage de 6 mois au sein de l'équipe AMAC à l'ISIR fût une formidable expérience. Je voudrai remercier Stéphane Doncieux, Alexandre Coninx et Carlos Maestre pour leur support et les nombreux conseils avisés qu'ils m'ont donnés. Bénéficier de leur expérience et savoirs a été un véritable atout pour mener à bien ce stage.

L'ISIR est un excellent laboratoire de robotique en France et avoir la possibilité d'y faire un stage a été une extraordinaire opportunité. Mon stage de 6 mois au sein de l'équipe AMAC fût une formidable expérience, notamment parce que j'ai pu travailler sur un domaine passionnant et relativement jeune, la robotique développementale. Par ailleurs, cette immersion dans un laboratoire de recherche m'a donné l'opportunité de découvrir davantage le monde de la recherche.

Reprendre des études en 2016 après avoir travaillé quelques années a été un véritable défi pour moi. En effet, cela supposait de quitter la ville où je vivais, mes amis et mon confort matériel ainsi que de revenir sur les bancs de l'université. J'allais avoir 30 ans et je me demandais si j'allais réussir dans ce Master 2.

1 Introduction

1.1 Le laboratoire

L'ISIR¹ (Institut des Systèmes Intelligents et de Robotique) est un laboratoire de recherche multi-disciplinaire créé en 2007 et co-tutellé par l'UPMC et le CNRS. Il rassemble des chercheurs de différents domaines : Sciences de l'Ingénieur et de l'Information ainsi que des Sciences du Vivant.

Les travaux de recherche à l'ISIR sont principalement centrés sur la modélisation et l'analyse des systèmes dynamiques artificiels et naturels, la conception optimale de systèmes robotiques interactifs, la commande des systèmes interactifs, la conception et le traitement du signal de systèmes perceptifs multimodaux, la modélisation des interactions homme - système, les modèles neuro-computationnels pour l'autonomie, l'apprentissage artificiel, l'adaptation bio-inspirée des systèmes et de leur commande. Dans cette perspective, le laboratoire est constitué de 4 équipes :

- AGATHE (Assistance aux Gestes et Applications THErapeutiques)
- AMAC (Architectures et Modèles pour l'Adaptation et la Cognition)
- INTERACTION
- SYROCCO (SYstèmes RObotiques COmplexes)

J'ai réalisé mon stage de fin d'études au sein de l'équipe AMAC, dirigée par Stéphane Doncieux. L'équipe est composée de 13 membres permanents et 6 membres non-permanents. Plusieurs membres sont spécialisés en neurosciences tandis que d'autres sont experts en robotique.

¹ <http://www.isir.upmc.fr>

1.2 Le projet DREAM

Mon stage a été financé dans le cadre le projet DREAM².

Une partie des membres de l'équipe AMAC travaille sur le projet DREAM (3 chercheurs, 2 post-doctorants, 3 doctorants et 4 stagiaires). Seung Su, Léni et Jonathan : tracking d'objets et création de *saliency map*.

Par exemple, plusieurs personnes travaillent sur la perception interactive. L'idée principale de la discipline est que pour segmenter des éléments (en l'occurrence des objets manipulables) dans une scène quelconque on ne peut pas se baser sur un a priori. Pour compenser cela, l'on va interagir avec l'environnement afin d'acquérir cette information. Le but d'un babilage est donc de produire une base de données. Une partie consiste à segmenter ce qui bouge et une autre doit pouvoir segmenter chaque objet. La segmentation de Leni me permet d'initialiser des hypothèses d'objets (ensemble de supervoxels) avec lesquels le robot interagit afin de valider ou non l'appartenance des supervoxels à un unique objet. La base de données qui est construite contient pour chaque objet les exemples de supervoxels qui lui ont été attribué.

Carlos : Apprentissage d'affordance Pierre : reconnaissance de formes ?

DREAM (Deferred Restructuring of Experience in Autonomous Machines) soit Restructuration différée de l'expérience dans les machines autonomes est un projet financé par le programme de recherche et d'innovation H2020 de l'Union Européenne. Ce projet a pour objectif d'incorporer un processus de sommeil et l'équivalent de rêves au sein d'une architecture cognitive, dans le but de consolider leur expérience et donc améliorer leur capacité à apprendre et à s'adapter.

Les différents partenaires académiques disposent de 5 robots pour mener à bien ce projet : le robot Meka, PR2 de Willow Garage, plusieurs Thymio-2, le bras robotique CrustCrawler ainsi que le robot Baxter de Rethink Robotics. La majorité des robots est présente au sein de l'ISIR.

Accumuler de la connaissance nécessite des processus de consolidation afin d'éviter d'être submergé par les informations. Le sommeil crucial pour la restructuration des représentations, maintenir l'intégration et la cohérence de la connaissance, amélioration de l'apprentissage et la formation d'abstraction.

Permettre aux robots d'acquérir une représentation ouverte du monde. Alterner périodes d'expériences et de sommeil.

Jusqu'à présent, le rôle du sommeil a été négligé dans l'IA et la robotique.

Selon DREAM, le cerveau de compose de 3 sous-systèmes : modèles, politique, valeurs.

Alternance entre interaction active et introspection passive.

Détails du projet Dream. Détails des différents objectifs. Description des différentes waves.

Au terme de ce projet, les différents travaux de recherche des différents membres ont vocation à s'imbriquer les uns avec les autres dans le but de fournir un système global, allant de la détection et le suivi d'objet, à l'apprentissage d'affordances en passant par l'apprentissage de lancers de balle.

La Coruna : travaux sur les motivations extrinsèques.

1.3 État de l'art

Medias are nowadays talking about smart robots for tomorrow or the day after tomorrow. Such mass medias are explaining that people would find in their house, in a very near future, a robot able to cook for you when you are short in time, mow the lawn and take care of elder people. However the path which conducts to these intelligent machines is still long and strewn with pitfalls. Several challenges have to be addressed before people interact with those robots as they do with other human people. Among these challenges, it is possible to cite movements learning.

La robotique développementale constitue un domaine de recherche relativement jeune. Il dérive principalement de la robotique mais tire également ses racines des neurosciences cognitives. Apprentissage autonome, généralisation. Plus efficace de créer un robot qui apprend qu'un robot auquel on apprend beaucoup de choses.

Affordances L'apprentissage des affordances est une étape cruciale dans l'approche de la robotique développementale. Le terme a été défini dans [1] comme *it offers the animal, what it provides or furnishes*. L'idée principale derrière ce terme est d'être capable de découvrir les différentes actions qui peuvent être réalisées sur un objet, en considérant les capacités disponibles. Cette définition établit une relation forte entre les objets, les actions et les effets (see Figure 1.3). Les affordances permettent de découvrir les différentes actions qui peuvent être réalisées sur un objet

² <http://www.robotsthatdream.eu>

en fonction des capacités de l'acteur. Par exemple, pour un robot avec des pinces, une boîte offre les affordances *movable* et *levable*, parmi d'autres. Les affordances sont capitables pour réaliser des actions, c'est-à-dire que le robot peut utiliser sa connaissance des affordances pour choisir la prochaine action à effectuer sur un objet pour accomplir une tâche. Par ailleurs, les affordances peuvent également être utilisées pour identifier ou reproduire des actions, telles qu'expliqué dans [2].

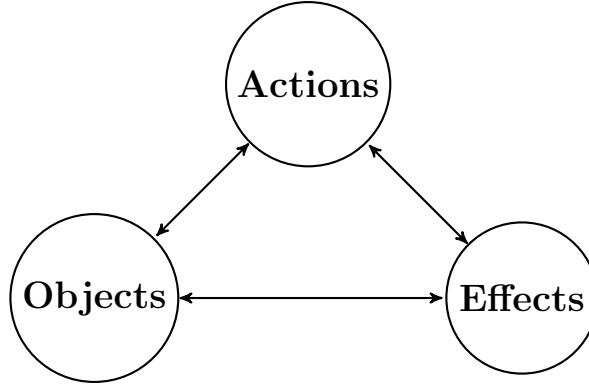


Fig. 1. Relation entre Objets, Actions et Effets.

Bostrapping Meltzoff et al. explained in [3] that infants do not know a priori what muscle movements achieve a particular state of organ relations. The authors call that process body babbling, directly inspired from babbling when infants experiment vocal sounds. In short, they explain that infants do repeated moves as a game. Eventually, after a certain amount of time and repetitions, infants would learn the relation between the moves they perform and theirs effects. That concept was reused in developmental robotics. Indeed, for a robot, learning affordances requires to interact with the environment through an exploration stage, which corresponds to babbling. Several heuristics exist to perform that exploration.

Motor babbling Une première approche naïve consiste à utiliser des mouvements aléatoires. Cette approche est la plus simple à implémenter mais elle peut générer un nombre important de mouvements qui ne produisent pas de contact avec les objets.

Another approach is presented in [4] where Maestre et al. described the use of Novelty Search, an evolutionary algorithm, in order to generate trajectories aimed at maximizing the environment's exploration. Results show that trajectories allowing to touch objects are more and more privileged over the different generations.

Intrinsic motivations Une troisième approche réside dans les motivations intrinsèques. Ce concept a été décrit par Ryan et Deci, puis par Hull et plus tard formalisé en modèles computationnels par Oudeyer et Kaplan dans [5]. Les motivations intrinsèques permettent d'explorer progressivement l'environnement en favorisant une spécialisation progressive des modules. Ce concept est utilisé en robotique et spécifiquement en robotique développementale.

TODO : Définition de motivations intrinsèques, extrinsèques, internes et externes. Équivalent à la Curiosité.

Context: The ability to perform in an open-ended environment and to build on previously acquired knowledge to quickly adapt to changes and unknown situations is key to building multi-purpose assistive robots which can be helpful in a wide range of realistic situations.

Cognitive Developmental robotics is a promising approach to create robots with higher cognitive functions as shown in [6] by Asada et al.

Utilisation d'outils Au delà des affordances, des recherches prometteuses se concentrent sur la manipulation d'outils. Dans [7], Gonçalves et al. proposent une approche intéressante pour apprendre les affordances visuelles d'objets ou d'outils. Dans cet article, un robot apprend les caractéristiques visuelles d'objets et d'outils à l'aide

de descripteurs visuels. De cette façon, ils montrent que le robot est capable de réaliser une action spécifique ou d'obtenir un résultat désiré. L'utilisation des descripteurs visuels permet au robot de généraliser avec des objets inconnus.

Réseaux bayésiens Une fois que le babillage sensorimoteur a été réalisé, les relations entre les objets, actions et effets nécessitent d'être modélisées. Les réseaux bayésiens constituent une méthode pour réaliser cela. Ce sont des modèles probabilistes qui représentent les relations entre leurs variables. Dans [8], Montesano et al. présentent un modèle de réseau bayésien qui permet d'apprendre des affordances concernant des objets.

Our work focused on improving the paper of [9]. In that paper, Ugur *et al.* described a system that allows a robot to learn goal emulation by using learnt affordances. In order to that, the robot follows an observation stage then an imitation stage.

- bootstrap object-oriented behavior - discover object in an unknown environment

1.4 Related work

Ce stage de fin d'études l'a été dans le cadre du projet DREAM, débuté en 2015. Ce projet implique 5 partenaires académiques: UMPC/CNRS (coordinateur), ENSTA ParisTech, Universidade da Coruña, Vrije Universiteit Amsterdam et Queen Mary University of London. Ces différents acteurs ont différents rôles au sein du projet DREAM. See waves.

At the end of the DREAM project, the different modules are aimed at being linked together.

Our work is directly based on actual PhD student's work on learning affordances inspired from motor babbling.

2 Notre travail

2.1 Environnement

L'environnement de travail durant notre stage a consisté en différents modules reliés entre eux : ROS, le simulateur Gazebo, le framework Moveit et le robot Baxter. ROS is a *flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms*³.

Gazebo est un simulateur⁴. La figure 7 donne un aperçu d'une simulation lancée avec Gazebo.

Moveit

Quant au robot Baxter⁵ (voir figure 2.1), c'est un robot développé par Rethink Robotics et utilisé à la fois à des fins industrielles et de recherche. Ce robot anthropomorphique à 2 bras dispose de 2 fois 7 degrés de liberté et de multiples senseurs⁶.

2.2 Adaptative discretization

Previous works Afin de réaliser cela, le nombre des actions prédéfinies disponibles était fixé à 4 (**Gauche, Droite, Haut, Bas**). Voir figure 3.

En raison de la nécessité de généraliser, For purpose of generalization, it would not be possible to let such a value. The first step of the internship focused on the goal of achieving environment's **adaptative discretization**. The main idea behind is to let the agent itself discover actions without specifying an arbitrary number of actions to discover. In order to do so, clusterization is used to distinguish those different actions. X-means algorithm[10], based on k-means, was used at the beginning. Basically, X-means is a k-means algorithm's generalization, which finds itself the best value for k .

³ <http://www.ros.org/about-ros/>

⁴ <http://gazebosim.org>

⁵ <http://www.rethinkrobotics.com/baxter/tech-specs/>

⁶ http://sdk.rethinkrobotics.com/wiki/Hardware_Specifications



Fig. 2. Baxter robot

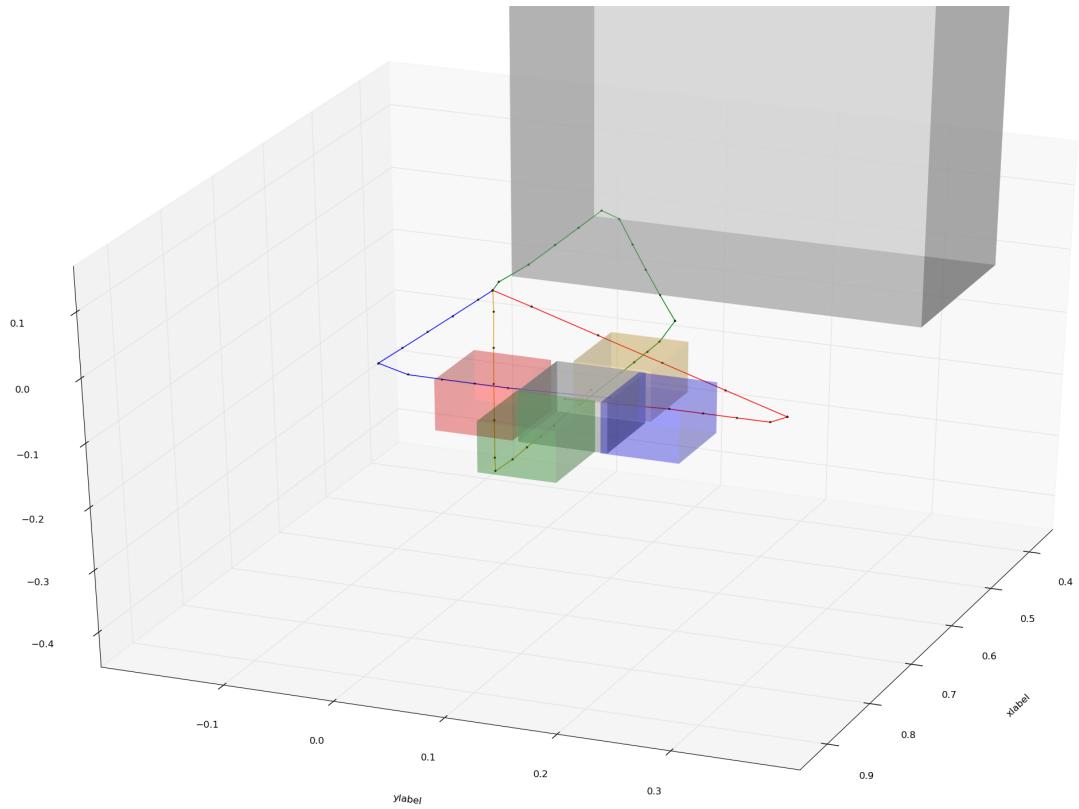


Fig. 3. Movements trajectories performed by the robot (gray shape in background). Initial position of the object is represented by the gray cube in the center. Each colored cube corresponds to a movement of the same color (e.g. the red cube is the final position of the gray cube after being pushed by the robot's end effector along the red trajectory).

Algorithmes de clustering En nous basant sur [11], [12], [13] et [14], nous avons réalisé une synthèse des algorithmes de clustering les plus communément utilisés, groupés en catégories (c.f. Annexe pour plus de détails). Afin de filter et sélectionner les algorithmes, nous avons choisi des critères non arbitraires. En se référant à la généralisation mentionnée ci-dessus, un premier critère concerne la nécessité de ne pas fournir le nombre attendu de clusters en paramètre. En effet, puisque le système doit être capable de trouver lui-même le nombre correct d'actions. Un second critère concerne la possibilité de travailler avec de nombreuses dimensions et un troisième concerne la disponibilité d'une implémentation en Python ou C++. Les algorithmes qui ne satisfaisaient pas à ces critères ont été rejettés.

7 algorithmes de clustering ont été comparés lors de cette première phase : X-means¹, Affinity Propagation¹, DBSCAN², HDBSCAN², OPTICS², Mean-Shift² et Level Set Tree. (1) correspondent à des algorithmes de partitionnement et (2) correspondent à des algorithmes basés sur la densité.

The intuition behind that choice is based on the fact that the different categories of actions have roughly the same end position. In other terms, a **Left** action will end in the specific area for left actions. Hence, a large amount of space will be empty and will not contain any end positions. The points distribution would lead to focus on finding areas of high density. For that purpose, density-based algorithms exist.

Afin de comparer les différents algorithmes, une série de tests a été réalisée avec 2 jeux de données différents : un dataset avec une distribution uniforme de points et un jeu de données généré en simulation.

Implémentation Nous avons utilisé le framework C++ Sferes [15] développé au sein de l'équipe AMAC et dédié aux calculs évolutifs. Ce framework a pour avantages d'être léger, multi-coeur et facilement extensible. Un très grand nombre

A large amount of clustering algorithms exist and were found in a previous stage (see in Annex). Seven of them were selected (mainly partition and density-based algorithms). A problem was to find the best algorithm and parameters for a given dataset. We needed also a way to validate the parameters found. That algorithm is used by the fitness function within the evolutionary algorithm. The proposed method consists in an iterative algorithm in which a clustering algorithm is tuned in order to satisfy three different objectives: ① the number of moves to reach the goal, to be minimized to make the robot as fast as possible, ② the number of clusters, to be minimized in order to simplify computations and ③ the final distance to the goal, to be minimized for the accuracy of the control. Dans la mesure où plusieurs objectifs sont à minimiser, nous avons sélectionné l'algorithme évolutif NSGA-II[16] pour remplir cette tâche.

Algorithm 1 Evaluation algorithm for fitness function

```

1:  $E = \{C_{e1}, \dots, C_{en}\}$                                      ▷ Set of clusters containing effects
2:  $F = \{f_1, \dots, f_n\}$                                        ▷ Set of final points
3:  $I = (0;0)$                                                  ▷ Initial point
4:  $t = 0$                                                        ▷ Number of tries
5: Compute the set of effects  $E$  from genotype
6: for all final point  $f_i$  in  $F$  do
7:   for  $j = 1$  to  $nb\_repeat$  do
8:     while  $dist(I, f_i) > \varepsilon$  and  $t < t_{max}$  do
9:        $M = \{m_i \mid m_i = rand(C_{ei}) \forall C_{ei} \in E\}$ 
10:       $m = \underset{m_i \in M}{\operatorname{argmin}}(dist(I + m_i, f_i))$ 
11:      Apply movement  $m$  to  $I$ 
12:       $t = t + 1$ 
13:    end while
14:  end for
15: end for
16: return ( $\overline{\text{nb movements}}$ , nb clusters,  $\overline{\text{final distance}}$ )

```

NSGA-II étant un algorithme multi-objectif, plusieurs meilleures solutions peuvent être trouvées, correspondants aux individus dominants le front de Pareto. La figure 4 montre la valeur du paramètre (de l'algorithme de clustering) correspondant aux individus dominants. Différentes valeurs peuvent être relevées, par exemple 0.05, 0.12 ou 0.31.

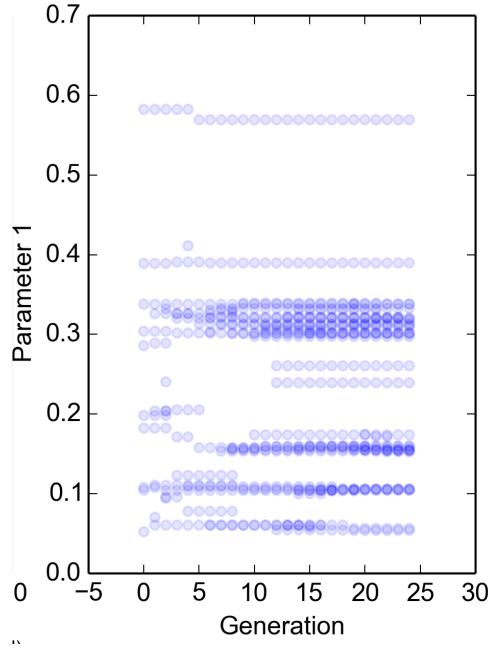


Fig. 4. Valeur du paramètre de l'algorithme de clustering Mean Shift correspondant aux individus dominants le front de Pareto.

4

À partir de ces meilleures valeurs de paramètres, nous avons généré (Figure 5) deux séries de graphiques. Les deux ensembles de clusters optimisent différents objectifs : vitesse et précision contre coût computationnel. La colonne de gauche contient davantage de clusters et permet un contrôle rapide et précis. À l'inverse, l'autre solution repose sur un nombre moins important de clusters.

Clusterisation based on Ugur's paper: 40 features. Compare Ugur's work with us.

Cette première partie de notre stage a fait l'objet d'une soumission d'un abstract long au workshop *Developmental learning and representation building in human-robots and ambient intelligence systems* de la conférence ECAL2017⁷ qui se tient à Lyon en septembre 2017.

Pushing object experiment La seconde partie de notre stage a consisté à mener une expérience en se servant de l'algorithme de clusterisation d'effets décrit ci-dessus. Une première expérimentation a eu lieu en simulation puis à l'aide du robot réel. Cette expérience peut être divisée en 3 étapes : ① génération du babbling sensorimoteur et enregistrement des effets pour les différents objets, ② réalisation de la clusterization des effets et ③ test de l'apprentissage. Voir figure 7 pour la mise en place initiale.

Afin d'obtenir des jeux de données contenant les différents effets à apprendre, le babillage sensorimoteur a été effectué à l'aide de l'algorithme génétique Novelty Search[17]. Cette approche est utilisée par Maestre et al. dans [4]. Pour cela, le calcul des trajectoires est effectué dans une première étape. Pour chaque trajectoire, on regarde si elle intersecte avec le cube. Si c'est le cas, cette trajectoire est gardée et jouée par le robot en simulation. La figure 6 montre un exemple de trajectoire touchant l'objet considéré.

Durant la première étape, différents objets sont placés successivement sur la table : une boîte de forme cubique, un cylindre et un cône. Ces objets ont différentes propriétés : certains peuvent être poussés à partir de n'importe quelle face et générer un comportement identique (boîte), tandis que d'autres nécessiteront d'être poussés depuis

⁷ <https://project.inria.fr/ecal2017/>

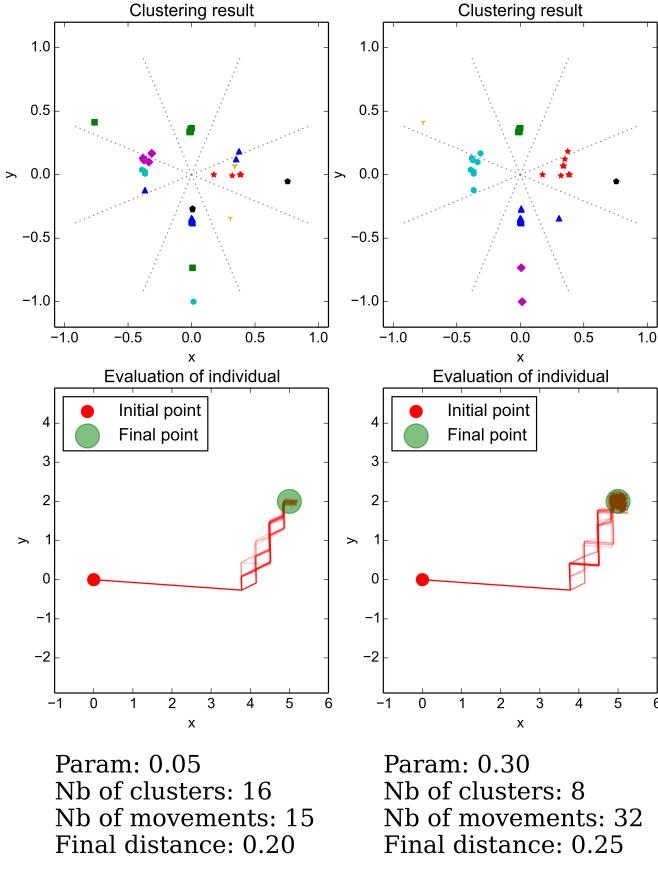


Fig. 5. Les deux colonnes montrent différents résultats de clustering et de trajectoires d'évaluations basés sur les valeurs de paramètres appartenant aux individus dominants le front de Pareto.

une face particulière (cylindre et cône) afin de garder une trajectoire cohérente. Le babillage réalisé par le robot sur ces différents objets permet de produire des jeux de données d'effets. La seconde étape implique l'algorithme que nous avons présenté ci-dessus et tente de clusteriser les effets des jeux de données. Il en résulte une variété de paramètres qui peuvent être exploités durant la troisième étape. Les meilleurs paramètres trouvés dans l'étape précédente sont utilisés pour pousser les différents objets vers la zone cible.

La première étape consiste donc en un babillage. Concrètement, la position de l'objet sur l'espace de travail est récupérée ainsi que la position de l'effecteur. Une trajectoire est ensuite générée à l'aide de la cinématique inverse et Novelty Search afin d'atteindre l'objet. Cet objet est positionné de façon aléatoire dans les intervalles $[x;x]$ et $[y;y]$.

3 Discussion

4 Future works & Perspectives

Plusieurs axes d'améliorations peuvent être proposés. Durant ce stage, nous avons testé l'algorithme présenté ci-dessus avec un seul algorithme de clustering, *Mean Shift*. Le premier axe d'amélioration consisterait à tester davantage d'algorithmes de clustering, tels qu'*Affinity Propagation*, *Birch* ou *DBSCAN*. Un second axe d'amélioration pourrait être de tester l'algorithme avec des tâches plus complexes.

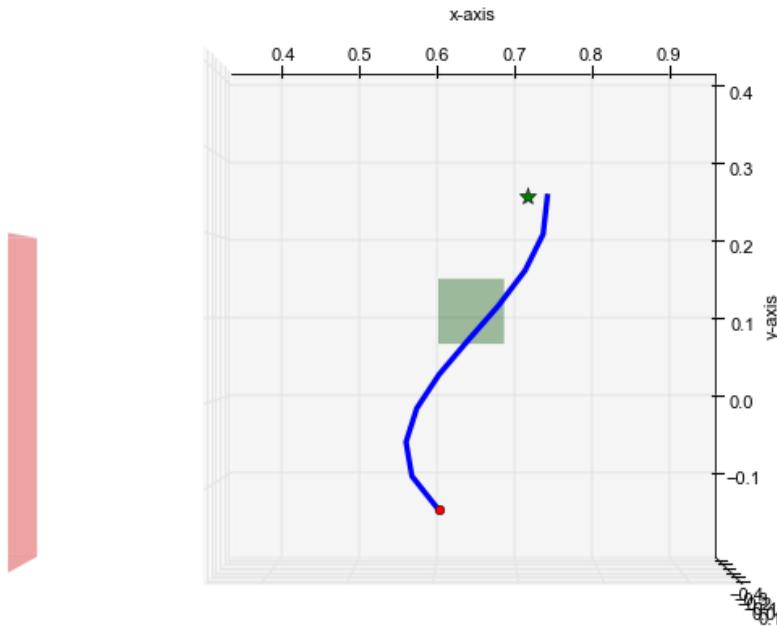


Fig. 6. Le rectangle rouge indique la position du robot. Le point rouge correspond au point de départ de l'effecteur du robot. Le tracé en bleu correspond à la trajectoire de cet effecteur vers l'objet. L'étoile verte indique la position de l'objet à la fin de l'exécution de la trajectoire de l'effecteur.

5 Conclusion

Ce stage de fin d'études de Master 2 m'a permis de davantage découvrir le domaine de la robotique développementale et ce qu'il implique, après la brève introduction que nous avons eu durant le semestre de M2 à Lyon. Ce domaine de recherche est fascinant et très intéressant dans la mesure où les chercheurs tendent à créer des robots qui pourraient bootstrap from almost no knowledge. Le lien étroit entre neurosciences et spécialement neurosciences cognitives permet d'implémenter des modèles présent chez l'être humain.

References

1. James Jerome Gibson. *The ecological approach to visual perception*. Houghton Mifflin, Boston, 1979. Includes indexes.
2. M. Lopes, F. S. Melo, and L. Montesano. Affordance-based imitation learning in robots. In *2007 IEEE/RSJ ICIRS*, pages 1015–1021, Oct 2007.
3. Andrew N. Meltzoff and M. Keith Moore. Explaining facial imitation: a theoretical model. *Early Development and Parenting*, 6(3-4):179–192, 1997.
4. Carlos Maestre, Antoine Cully, Christophe Gonzales, and Stephane Doncieux. Bootstrapping interactions with objects from raw sensorimotor data: A novelty search based approach. *5th Joint International Conference on Development and Learning and Epigenetic Robotics, ICDL-EpiRob 2015*, pages 7–12, 2015.

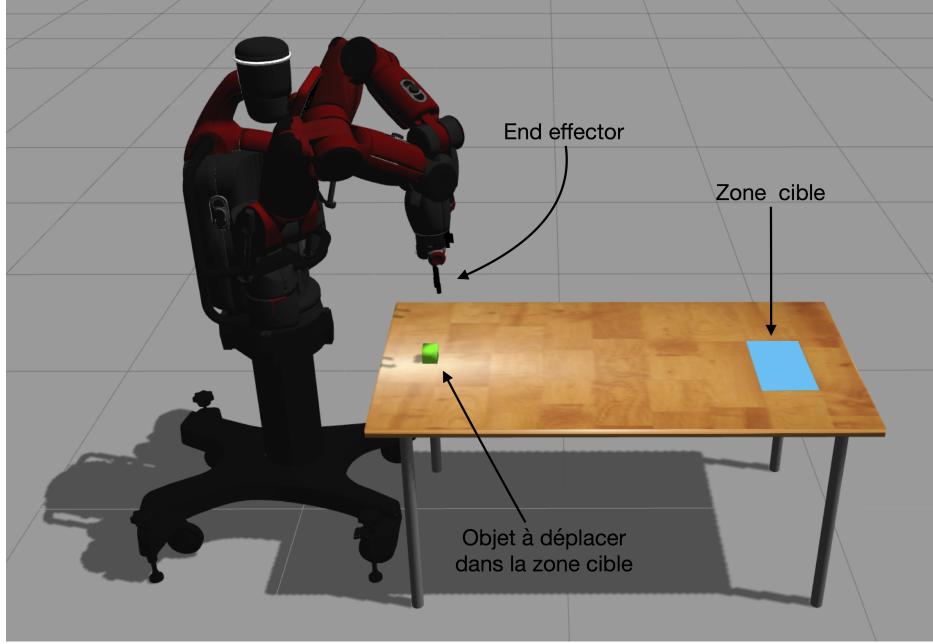


Fig. 7. L'objectif du robot est d'apprendre à pousser différents objets (par exemple, un cube) depuis sa position initiale jusqu'à la zone cible.

5. Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in Neurorobotics*, 1:6, 2009.
6. Minoru Asada, Koh Hosoda, Yasuo Kuniyoshi, Hiroshi Ishiguro, Toshio Inui, Yuichiro Yoshikawa, Masaki Ogino, and Chisato Yoshida. Cognitive Developmental Robotics: A Survey. *IEEE Transactions on Autonomous Mental Development*, 1(1):12–34, 2009.
7. A. Gonçalves, G. Saponaro, L. Jamone, and A. Bernardino. Learning visual affordances of objects and tools through autonomous robot exploration. In *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 128–133, May 2014.
8. L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor. Learning object affordances: From sensory–motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15–26, Feb 2008.
9. Emre Ugur, Erhan Oztop, and Erol Sahin. Goal emulation and planning in perceptual space using learned affordances. *Robotics and Autonomous Systems*, 59(7-8):580–595, 2011.
10. Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 727–734, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
11. Dongkuan Xu and Yingjie Tian. A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2):165–193, 2015.
12. Bill Andreopoulos, Aijun An, Xiaogang Wang, and Michael Schroeder. A roadmap of clustering algorithms: Finding a match for a biomedical application. *Briefings in Bioinformatics*, 10(3):297–314, 2009.
13. Adil Fahad, Najlaa Alshatri, Zahir Tari, Abdullah Alamri, Ibrahim Khalil, Albert Y. Zomaya, Sebti Foufou, and Abdelaziz Bouras. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Transactions on Emerging Topics in Computing*, 2(3):267–279, 2014.
14. T. Sajana, C. M. Sheela Rani, and K. V. Narayana. A survey on clustering techniques for big data mining. *Indian Journal of Science and Technology*, 9(3):1–12, 2016.
15. J.-B. Mouret and S. Doncieux. SFERESv2: Evolvin' in the multi-core world. In *Proc. of Congress on Evolutionary Computation (CEC)*, pages 4079–4086, 2010.
16. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, 6(2):182–197, April 2002.
17. J. Lehman and K. O. Stanley. Improving evolvability through novelty search and self-adaptation. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2693–2700, June 2011.