

Rapport de stage
Master 2 d'informatique fondamentale
Analyse d'un algorithme de curiosité artificielle

Auteur :
Tom DUSSÉAUX

Encadrants :
Frédéric ARMETTA
Mathieu LEFORT

Laboratoire :
Laboratoire d'Informatique en Image
et Systèmes d'Information

Equipe :
Systèmes Multi-Agents

9 juin 2017

Remerciements

Je tiens en premier lieu à remercier mes deux encadrants Frédéric Armetta et Mathieu Lefort. Ils ont été d'une grande aide grâce à leurs connaissances et au temps qu'ils m'ont accordé.

Je remercie l'ensemble des membres du laboratoire pour leur accueil.

Mes compagnons de box également stagiaires pour l'entraide et les bons moments passés.

Je remercie l'ensemble des professeurs de l'ENS pour les connaissances scientifiques et le challenge que représenta cette année.

Je remercie spécialement mon tuteur, Philippe Audebaud pour avoir cru en mes capacités et m'avoir soutenu.

Je remercie également l'ensemble des enseignants m'ayant amené à poursuivre dans de longues études.

Je remercie plus généralement tous les amoureux de la science qui m'ont partagé leur passion.

Je remercie tous mes proches qui se reconnaîtront.

Je remercie Sophie Billon pour la relecture de mon rapport (entre autres).

Je remercie Marc Heinrich pour l'aide mathématique indispensable qu'il m'a apportée pour mes rattrapages.

Table des matières

1	Introduction	0
1.1	Robotique développementale et agent autonome	0
1.1.1	Motivations	0
1.1.1.1	Motivations générales	0
1.1.1.2	Motivations de la création d'agents autonomes	1
1.1.2	Inspiration des sciences cognitives	1
1.1.2.1	Lois du développement psychomoteur	1
1.1.2.2	Le développement incarné	1
1.1.3	Apprentissage sensorimoteur	2
1.1.3.1	Modèle direct	2
1.1.3.2	Modèle inverse	2
1.1.4	Algorithme d'apprentissage	2
1.2	Défis et stratégies de l'exploration	2
1.2.1	Les différents défis de l'exploration	2
1.2.1.1	L'inapprenabilité	2
1.2.1.2	Les espaces hétérogènes	3
1.2.1.3	Les espaces non-bornés ou très grands	3
1.2.1.4	Les espaces de grande dimensionnalité	3
1.2.2	Contraintes développementales	3
1.2.2.1	Méthodes supervisées	3
1.2.2.2	Contraintes biologiques et maturationnelles	3
1.3	Motivations de la curiosité	4
1.3.1	Hédonisme et homéostasie	4
1.3.2	Motivations extrinsèques et intrinsèques	4
2	RIAC	5
2.1	Cadre d'application de RIAC	5
2.1.1	Machine de prédiction	5
2.1.2	Spécificité de la machine de prédiction	5
2.2	Fonctionnement détaillé de RIAC	5
2.2.1	Calcul du progrès en apprentissage	6
2.2.2	Représentation des régions	6
2.2.3	Mécanisme de choix d'action	7
2.2.3.1	Premier mode	7
2.2.3.2	Deuxième mode	7
2.2.3.3	Troisième mode	8
2.2.4	Mécanisme de découpe	8
3	Expérimentation de RIAC et analyses	9
3.1	Description de l'environnement	9
3.2	Expérimentations de RIAC	10
3.2.1	Note sur l'expérience	10
3.2.2	résultats	10
3.3	Analyse du fonctionnement de RIAC	10
3.3.1	Erreur moyenne par régions	10
3.3.2	Progrès en apprentissage par régions	11
3.3.3	Efficacité du mécanisme de choix d'action	11
3.3.4	Analyse du mécanisme de découpe	11
3.3.4.1	Analyse des découpes effectuées dans un espace à deux dimensions	11

3.3.4.2	Différence du progrès en apprentissage	12
3.3.4.2.1	Hypothèse de conservation de la représentativité temporelle	13
3.3.4.2.2	Comparaison entre la pratique et la théorie	13
3.3.4.2.3	Conclusion	14
4	Proposition de nouvelle mesure pour la découpe	15
4.1	Découpe en utilisant le progrès pondéré	15
4.1.1	Principe	15
4.1.2	Résultats théoriques et pratiques	15
4.1.3	Découpe en utilisant l'erreur moyenne	16
4.1.4	Principe	16
4.1.5	Résultats théoriques et pratiques	16
4.1.6	Découpes obtenues	16
4.1.7	Tests qualitatifs	17
4.1.8	Conclusion	17
5	Conclusion et perspectives	18
6	Annexe	19
6.1	Erreur moyenne et progrès en apprentissage par régions	19

Chapitre 1

Introduction

Ce stage se situe dans l'intersection de trois domaines de recherche différents. L'intelligence artificielle, les sciences cognitives et la robotique. L'intersection de la robotique et de l'IA traduit la volonté de créer des robots capables de réaliser des tâches plus complexes qu'une suite d'instructions simples. L'intersection de l'IA et des sciences cognitives correspond à la création d'intelligences artificielles inspirées par les phénomènes observés dans les sciences cognitives. L'intersection des trois domaines correspond ainsi naturellement à des robots intelligemment inspirés par les sciences cognitives. Les différents domaines sont illustrés dans la figure 1.1.

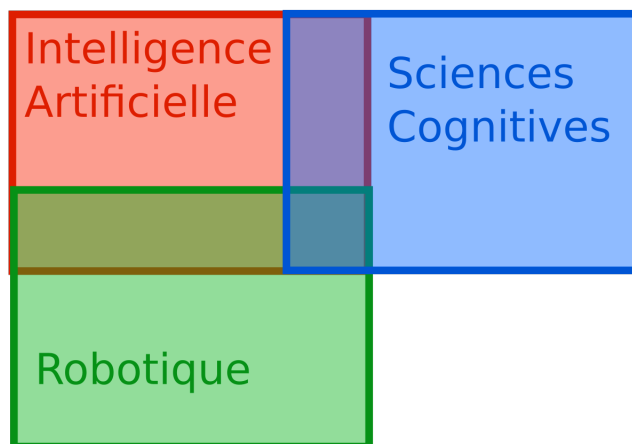


FIGURE 1.1 – Schéma représentant l'interaction entre les différents domaines

1.1 Robotique développementale et agent autonome

1.1.1 Motivations

1.1.1.1 Motivations générales

La fin du 19^{ème} siècle et le début du 20^{ème} furent bouleversés par les avancées technologiques de la révolution industrielle. Elles renversèrent le rapport au travail et à la production. La continuité de ces avancées fut promise par la robotique, et notamment la perspective de robot intelligent. L'apport de robot intelligent aurait des applications actuellement difficilement prédictibles tant elles bouleverseraient notre manière de penser. L'avènement de l'informatique permit de faire naître le domaine de l'intelligence artificielle et le rêve de la création d'une intelligence comparable à celle de l'homme. Au milieu du 20^{ème} siècle Alan Turing propose un test pour évaluer l'intelligence des IA [15], en 1997 une IA Deep Blue bat le champion d'échec Garry Kasparov [6]. L'on prédit alors une victoire imminente des IA sur différents jeux comme le jeu de go. Cette victoire viendra près de vingt ans plus tard, grâce à l'algorithme AlphaGo, qui surpassera enfin l'homme. Mais ces victoires ne constituent que des avancées mineures dans la conception d'une intelligence artificielle réelle. Les difficultés énoncées que l'on pensait rapidement surmontables se trouvent demeurer bloquantes. Durant ce rapport nous énoncerons différents points de blocages de la création d'une intelligence artificielle.

1.1.1.2 Motivations de la création d'agents autonomes

Actuellement on sait réaliser des robots et intelligences artificielles bien plus performants que l'Homme. On remarque par contre que les robots sont souvent incapables de réaliser un grand nombre de tâches. La plupart des intelligences artificielle on été conçues afin de répondre à un nombre restreint de problèmes spécifiques. Pour parler de l'instance d'un algorithme d'intelligence artificielle ou d'un robot, nous utiliserons le terme d'agent. Il serait ainsi souhaitable qu'un agent soit capable d'apprendre de nouvelles capacités et connaissances au cours de sa vie. Il serait même idéal que cet agent soit capable d'apprendre de manière autonome [16]. Le nombre de situations dans lequel pourra se trouver le robot est en effet trop important pour pouvoir anticiper tout ce que le robot devrait connaître ou savoir faire. Nous reviendrons dans la partie 1.2.1.4 sur le problème du nombre de connaissances et comportements apprenables. La création d'agents autonomes est donc indispensable à la volonté de robots polyvalents.

1.1.2 Inspiration des sciences cognitives

La création d'un agent autonome intelligent constitue un défi conséquent. Les recherches ne permettent actuellement pas de connaître les meilleures stratégies pour le réussir. Il se peut également que certaines techniques soient indispensables à l'émergence de l'intelligence. Afin d'orienter les pistes de recherche, on peut se tourner vers les mécanismes étudiés dans le domaine des sciences cognitives. Ce domaine s'intéresse à l'étude des seules preuves de l'existence de l'intelligence. Si certains mécanismes sont démontrés primordiaux dans l'apprentissage d'un être vivant, on peut supposer qu'ils favorisent grandement l'apprentissage en général. Ils constituent alors une bonne piste de recherche dans la création d'un agent autonome intelligent.

1.1.2.1 Lois du développement psychomoteur

A travers l'étude du développement de l'enfant, les études ont montré l'existence de différents processus et lois influant sur l'apprentissage. Ces processus sont notamment liés à l'ordre dans lequel l'enfant apprendra différents comportements. Avant d'apprendre des comportements complexes comme la marche, un enfant apprendra à s'asseoir, à se déplacer à quatre pattes et à se tenir debout. Bien que l'ordre d'apprentissage puisse différer d'un enfant à l'autre, on observe un apprentissage incrémental des compétences.

Plusieurs lois on été étudiées afin de caractériser le développement de l'enfant. La loi de différenciation [10] spécifie que l'enfant commence par effectuer des mouvements sans but précis. Il réalise des mouvements correspondant à un babillage de ses muscles. Au fur et à mesure, l'enfant différencie le monde extérieur de son corps et peut ainsi commencer à agir consciemment.

La loi de variabilité spécifie que la vitesse d'apprentissage des enfants est fortement irrégulière. Les enfants n'apprennent pas à la même vitesse les uns des autres. Et la vitesse d'apprentissage peut subir des phases de pauses, des évolutions soudaines et même des régressions. Ces variations peuvent s'expliquer par de nombreuses hypothèses décrites en partie par la suite. Elles peuvent être la conséquence d'un développement physique différent ou d'une orientation de l'apprentissage différent.

En observant le niveau de contrôle des différents muscles de l'enfant, on a observé que les muscles les plus éloignés du cerveau. On appelle ce phénomène loi de succession, il est composé de deux phénomènes plus précis, les lois céphalo-caudale et proximo-distale[3]. La première précise que l'on apprend en premier vers le haut du corps. La deuxième précise que les extrémités du corps seront contrôlables plus tardivement. Ainsi on contrôle efficacement le cou avant le bassin et les bras avant les mains.

On constate également un phénomène étonnant, certaines compétences sont plus facilement apprenables à un moment du développement. Il est ainsi possible que si une compétence n'a pas été apprise à un moment, elle devienne beaucoup plus complexe à rattraper voire impossible à apprendre[12].

1.1.2.2 Le développement incarné

Le développement incarné [14] est un courant de pensée considérant que l'intelligence est inhérente au corps. Le problème de la création d'une représentation du monde découlerait des actions que l'on peut effectuer dedans. A haut niveau d'abstraction, les objets ne sont plus définis par leurs caractéristiques mais par l'usage que l'on peut en avoir.

Une expérience permet d'illustrer ce phénomène [8]. On place deux chatons dans une chambre circulaire. L'un des chatons peut se déplacer tandis que l'autre sera contraint mécaniquement d'effectuer des mouvements similaires à ceux du premier chaton. Les conditions de l'expérience impliquent que les deux chatons auront la même vision de l'environnement. On observe après huit semaines que le premier chaton est capable de se déplacer sans problèmes dans l'environnement. Le deuxième chaton lui éprouve des difficultés, notamment pour appréhender les distances. Cette expérience démontre que l'impact de son corps dans l'environnement influe sur l'apprentissage.

1.1.3 Apprentissage sensorimoteur

Le chapitre précédent nous indique que l'apprentissage de l'impact des actions sur l'environnement peut aider à l'apprentissage d'un robot. Il est donc important de différencier les informations concernant l'environnement récupéré par les différents capteurs. Il s'agit de la dimension sensitive de l'environnement S . Il peut s'agir de données de bas niveau comme les entrées de ses capteurs sensoriels. Ou des données traitées à plus haut niveau comme la position d'un objet dans le monde. La dimension motrice ou d'action M correspond quant à elle aux actions effectuables par le robot. Là encore, il peut s'agir d'une donnée de bas niveau comme l'ensemble des commandes données à chacun des moteurs. Mais également à une commande plus avancée correspondant par exemple à un ordre de plus haut niveau. Par exemple "avance d'un mètre" ou "va dans la cuisine". Une donnée dans la dimension motrice ou sensitive est caractérisée par rapport à un temps noté t

Pour simplifier le vecteur $M(t) \times S(t)$ sera noté $SM(t)$, cet espace est communément appelé espace sensorimoteur.

1.1.3.1 Modèle direct

Il existe deux modèles que l'agent peut apprendre afin de comprendre son corps. La première est la plus simple des méthodes et consiste à apprendre l'impact des actions sur l'environnement et ainsi à pouvoir réaliser une tâche de prédiction. C'est à dire l'apprentissage de la relation $SM(t) \rightarrow S(t+1)$ soit chercher à reproduire la fonction f définie par $S(t+1) = f(SM(t))$. Cette fonction est la conséquence sensorielle induite par une action réalisée dans un certain contexte sensorimoteur.

1.1.3.2 Modèle inverse

L'apprentissage d'un modèle inverse consiste à apprendre quelle action permet de passer d'un état donné à un autre. Cela se traduit par l'apprentissage de la relation $S(t) \times S(t+1) \rightarrow M(t)$ et ainsi approximer la fonction g définie par $M(t) = f(S(t) \times S(t+1))$

1.1.4 Algorithme d'apprentissage

Afin d'apprendre les différents types de modèle, on utilise des algorithmes d'apprentissage permettant d'approximer une fonction. On peut séparer ces algorithmes en deux catégories :

- Les modèles paramétriques sont définis par un nombre fini de paramètres. L'apprentissage consiste à trouver la valeur de ces paramètres approxinant le mieux la fonction. La modélisation de la fonction se fait de manière globale, la modélisation se fait simultanément pour l'ensemble de l'espace. Parmi les différents modèles paramétriques, on trouve les perceptrons multi-couches (aussi appelés réseaux neuronaux) [?], les modèles de mélange de gaussiennes (GMM)[5].
- Les modèles non-paramétriques ne possèdent pas un nombre de paramètre fixe et par conséquent une structure fixe. Ils évoluent typiquement proportionnellement au nombre d'exemples. Parmi ces modèles, le plus trivial consiste à stocker les exemples, puis pour prédire la fonction de trouver l'exemple le plus proche de l'entrée donnée. Ce type d'algorithme correspond à un algorithme de plus proche voisin. On peut également considérer un plus grand nombre de voisins. Il existe bien évidemment des modèles non-paramétriques beaucoup plus complexes. Nous nous attarderons dans la partie 2.1.2 à détailler un mécanisme permettant de rendre un modèle paramétrique non-paramétrique.

1.2 Défis et stratégies de l'exploration

1.2.1 Les différents défis de l'exploration

1.2.1.1 L'inapprenabilité

Un problème que l'on rencontre quand on tente d'apprendre dans un espace quelconque, est la présence de régions inapprenables. Dans ces régions, il n'y a pas de relation à apprendre. L'exemple typique est de chercher une relation entre le déplacement d'une main et la position d'un nuage dans le ciel. Quels que soient les mouvements effectués par la main, le nuage se déplacera indépendamment. Dans le cadre de l'usage de modèle paramétrique la présence de ce type de zone peut rendre l'apprentissage impossible en raison de la nature globale du modèle 1.1.4. Dans le cadre de l'usage d'un modèle paramétrique la présence de ce type de zone peut faire perdre du temps à l'algorithme qui tentera d'apprendre une relation inexistante.

1.2.1.2 Les espaces hétérogènes

Certains espaces ne sont pas homogènes en terme d'apprentissage. Ils peuvent par exemple comprendre des régions où l'apprentissage est quasiment immédiat dans des régions dites triviales. Par exemple la position d'un objet statique. Certaines régions peuvent être plus longues à apprendre que d'autres. Par exemple apprendre la position de l'extrémité du bras d'un robot sera sûrement plus difficile que la position de son coude. Pour des modèles paramétriques ces contraintes peuvent rendre l'apprentissage impossible. Et dans le cadre d'un modèle quelconque cela pose problème si l'on dispose d'un temps limité pour apprendre. Si l'on souhaite apprendre convenablement un maximum de comportements, il ne faut pas s'attarder sur les régions triviales et ne pas se concentrer sur celles trop longues à apprendre.

1.2.1.3 Les espaces non-bornés ou très grands

Une difficulté récurrente concernant des espaces à apprendre concerne leur taille. Ils peuvent être de taille très grande voire même de taille non-bornée. Cela implique qu'un nombre très important voir infini d'exemples peut être nécessaire pour comprendre l'ensemble de l'espace. Si l'on ne se concentre pas spécifiquement dans des régions, on risque d'être très mauvais dans l'ensemble de l'espace.

1.2.1.4 Les espaces de grande dimensionnalité

Le nombre de dimension de l'espace sensitif et de l'espace moteur peut être également source de problèmes. Dans le cas d'un être humain le nombre de primitives motrices et sensorielles est par exemple très élevé. Si l'on souhaite échantillonner un espace à une dimension $[0 : 1]$ avec des points espacés de 0.01, 100 points suffisent. Si l'on passe à un hypercube à n dimensions, il faut 10^{2*n} points pour obtenir la même précision.

1.2.2 Contraintes développementales

Afin de ne pas se perdre dans des régions inapprenables ou sans intérêt, on aimerait contraindre l'apprentissage dans certaines régions afin de palier aux problèmes cités dans la section 1.2.1. Les contraintes doivent également permettre de suivre un certain ordre d'apprentissage pour correspondre à la stratégie développementale décrite dans la section 1.2.1. Afin d'orienter l'apprentissage en suivant le développement de l'agent, plusieurs stratégies peuvent être envisagées.

1.2.2.1 Méthodes supervisées

La méthode la plus simple dans le principe mais complexe dans la réalisation consiste à donner une liste d'exemples à l'agent. On peut ainsi choisir des exemples dans les régions d'intérêt et les ordonner pour correspondre à l'ordre d'apprentissage. La liste des exemples peut être très compliquée et fastidieuse à réaliser. On peut même faire remarquer qu'une partie de l'intelligence se cache dans la liste des exemples.

Cette solution entre en contradiction avec les motivations de la section 1.1.1.2 justifiant l'utilisation d'un agent autonome. Une autre proposition non-autonome proposée dans la littérature consiste à guider le robot vers les régions apprenables. Cette méthode se base sur le guidage social [1] On donne un exemple dans la région apprenable puis le robot explore les alentours de ce point pour apprendre. Cette méthode est par exemple utilisée en manipulant manuellement le bras du robot dans une première phase pour lui faire effectuer une tâche. Rendre un robot manipulable demande d'adapter les moteurs pour qu'il soit libre et qu'il imprime les mouvements réalisés [4]. L'impression des mouvements n'étant pas précise, si le robot effectue la commande qu'il a utilisée, il ne pourra pas forcément reproduire la tâche. Mais il pourra apprendre dans la région de cette commande afin de trouver la réelle commande à réaliser.

1.2.2.2 Contraintes biologiques et maturationnelles

Une autre solution se base sur des contraintes physiologiques et psychiques observées chez les êtres vivants. On observe que les êtres vivants naissent avec des capacités motrices et psychiques plus faibles qu'à la maturité. Les capacités se dévoilent au fur à mesure de la maturation permettant l'apprentissage de comportements plus complexes. Des études ont montré que ces capacités permettent de réduire la taille de l'espace à apprendre. Ces contraintes ne répondent pas à nos motivations de la section 1.1.1.2 car elles impliquent que l'on connaît suffisamment l'environnement pour pouvoir concevoir un robot évoluant.

La dernière solution consiste à créer un robot s'orientant naturellement vers les régions comportant un intérêt. Cette technique se base sur le postulat que les différentes régions sont clairement identifiables et continues. Les régions où l'agent apprend le plus sont alors appelées dans la littérature zones proximales de développement. Ce mécanisme correspond dans le langage courant à la curiosité.

1.3 Motivations de la curiosité

Concernant les agents autonomes se pose la question des motivations à l'apprentissage. On rajoute ici la contrainte d'un apprentissage curieux qui contraint les propositions de motivations à explorer.

1.3.1 Hédonisme et homéostasie

L'hédonisme consiste à considérer que les êtres vivants sont uniquement motivés par le plaisir et la fuite du manque [7]. L'homéostasie est que l'on cherche à maintenir son corps dans un état d'équilibre satisfaisant ses besoins physiologiques. Si ces notions fonctionnent apparemment pour décrire le comportement de certains animaux, elle est montrée insuffisante pour certaines espèces. On remarque par exemple que des animaux peuvent manger alors qu'ils n'en ont pas besoin ou que l'animal choisit une nourriture moins nutritive mais plus à son goût, ces deux comportements ne sont pas expliqués par la théorie hédoniste. Concernant l'homéostasie des études sur des rats démontrent qu'ils peuvent supporter des souffrances afin d'explorer leur environnement.

1.3.2 Motivations extrinsèques et intrinsèques

Il s'agit de motivations fournies par l'extérieur à l'agent. Par exemple chez l'Homme on peut citer le désir de reconnaissance, la soif de pouvoir ou l'appât du gain. Ce types de motivations ne semblent là encore pas en mesure d'expliquer certains comportements observés chez l'Homme. Il semble que le gain de connaissance soit une motivation en soi, une motivation venant de l'individu directement. Ce types de motivations en opposition aux motivations extrinsèques s'appellent motivations intrinsèques.

Afin de reproduire dans un agent le phénomène de curiosité intrinsèque, il faut identifier pourquoi une connaissance serait plus ou moins intéressante. Une première théorie consiste à proposer que l'Homme cherche à réduire au maximum les dissonances cognitives. C'est à dire à créer une construction du monde la plus cohérente possible. Cette théorie permet d'apporter des pistes de réponses mais n'explique pas pourquoi l'homme n'explore pas continuellement le même espace connu. Afin de s'orienter une théorie issue des sciences cognitives, la théorie du Flow propose de considérer que l'on cherche à maximiser l'apprentissage. Les régions que l'on ne connaît pas sont considérées comme comportant du challenge. S'évaluer dans des régions qui ne comportent pas de challenge permet un bon taux de réussite mais cela ne correspondra pas à une région d'intérêt. S'évaluer dans des régions avec du challenge en échouant est frustrant et ces régions sont donc pour le moment à éviter. S'évaluer dans une région comportant du challenge mais dans laquelle on réalise un résultat encore jamais obtenu est très satisfaisant. Cette région correspond à la zone proximale de développement

Chapitre 2

RIAC

RIAC (Robust-Intelligent Adaptive Curiosity) est un algorithme de meta-apprentissage intrinsèquement motivé. Il se base dans la continuité de l'algorithme IAC (Intelligent Adaptive Curiosity) en apportant des améliorations dans le but de le rendre plus efficace. L'objectif de RIAC est d'orienter les choix des actions à effectuer afin d'accélérer l'apprentissage. Il s'agit d'un algorithme de meta-apprentissage car il se place au-dessus du modèle d'apprentissage. Pour cela l'algorithme va identifier différentes régions et apprendre dans les régions où l'apprentissage est le plus effectif.

2.1 Cadre d'application de RIAC

2.1.1 Machine de prédiction

La machine de prédiction comporte un modèle d'apprentissage direct décrit dans la section 1.1.3.1.

La fonction f' fournit une approximation de l'état réel nous noterons ainsi cette approximation $\tilde{S}(t+1) = f'(SM(t))$

Afin de pouvoir orienter l'exploration RIAC a également besoin de connaître l'erreur $e(t+1)$ de prédiction de la machine de prédiction pour le contexte sensorimoteur $SM(t)$. Le calcul de l'erreur pouvant de manière basique correspondre à $e(t+1) = |\tilde{S}(t+1) - S(t+1)|$

2.1.2 Spécificité de la machine de prédiction

Bien que RIAC ne spécifie pas les moyens utilisés dans la machine de prédiction, le cadre d'utilisation de RIAC lui restreint le choix de la machine de prédiction ou demande des ajustements. La machine de prédiction doit être capable de fonctionner dans un environnement hétérogène. En effet l'objectif de RIAC est d'orienter l'apprentissage dans un environnement possédant des régions s , la machine doit alors pouvoir apprendre dans des environnements possédant de telles régions. La machine doit également être incrémentale et en ligne. C'est à dire que la machine doit effectuer des prédictions alors même qu'elle ne possède pas de données. Et que la machine doit pouvoir apprendre de données arrivant une par une.

Afin de pouvoir utiliser des outils de régression ne répondant pas à ces pré-requis, un modèle dit ILO est proposé dans la thèse [2]. Il consiste à effectuer un k-voisins afin d'entraîner le modèle sur les voisins de la donnée à prédire. Le modèle réapprend alors à chaque prédiction. Cela permet de transformer un modèle dit paramétrique, qui consiste à être un modèle apprenant la valeur d'un nombre fini de paramètres, en un modèle non-paramétrique qui retient les exemples précédemment effectués.

2.2 Fonctionnement détaillé de RIAC

L'algorithme de RIAC possède plusieurs mécanismes, un mécanisme de choix d'action et un mécanisme de découpe de région. Le schéma 2.1 présente le fonctionnement global de RIAC. Le mécanisme de choix d'action (1) choisira de préférence des actions dans les régions où l'on apprend de la manière la plus efficace. La machine de prédiction (2) réalisera une prédiction de l'effet de l'action. L'environnement (3) calculera le véritable effet de l'action. L'erreur d'approximation sera calculé et stocké. Si il y a plus de T_{split} exemples dans une région feuille en sera découpé par le module de découpe (4).

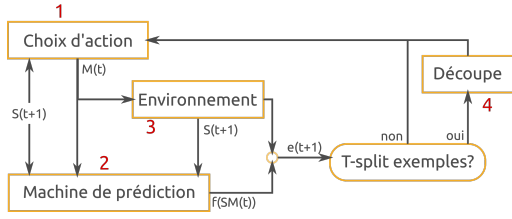


FIGURE 2.1 – Schéma représentant les différentes étapes de l'algorithme RIAC. Les différents modules de RIAC sont numéroté par leur ordre

2.2.1 Calcul du progrès en apprentissage

Afin de pouvoir réaliser effectuer le choix et la découpe l'algorithme se base sur une valeur appelée progrès en apprentissage. Elle correspond au gradient de l'erreur dans le temps. Le progrès en apprentissage est calculé à partir de l'ensemble des dernières erreurs effectuées dans la région notée r par la machine de prédiction. Les régions correspondent à des restrictions d'un espace, ici l'espace sensorimoteur. Nous revenons plus en détail sur la notion de région dans la partie 2.2.2.

Cet ensemble sera noté $E^r = e(1), e(2), \dots$ et sera de taille $|E^r|$. La valeur E_i^r correspondant à la valeur $e(i)$ contenue dans E^r . Afin de réaliser les calculs nous utiliserons des fenêtres permettant d'obtenir des sous-ensembles d'erreurs consécutives. Nous noterons k_{max} la taille maximale de la fenêtre utilisée pour le calcul du progrès en apprentissage. Nous noterons k^r la taille réelle de la fenêtre utilisée pour le calcul, $k^r = \min(k_{max}, \frac{|E^r|}{2})$.

Le calcul du progrès en apprentissage dans une zone r noté LP^r correspond alors à la différence de la somme des erreurs les moins récentes avec les erreurs les plus récentes.

$$LP^r = \frac{\sum_{i=|E^r|-2*k^r}^{|E^r|-k^r} E_i^r - \sum_{i=|E^r|-k^r}^{|E^r|} E_i^r}{2}$$

La division par 2 permet de correspondre à la définition théorique du calcul du gradient. Mais l'utilisation de cette donnée dans RIAC rend cette division usuelle car le progrès en apprentissage d'une région ne sera utilisé que pour le comparer au progrès en apprentissage d'autres régions.

Le progrès en apprentissage permet théoriquement de différencier trois types de régions :

- Les régions inapprenables où l'erreur moyenne ne diminue pas et où le progrès en apprentissage devrait ainsi être nul.
- Les régions d'apprentissage triviales où l'erreur moyenne chute si rapidement que le progrès sera également quasi-nul.
- Les régions où l'on est actuellement en train d'apprendre et où ainsi l'erreur moyenne décroît dans le temps. Ces régions ont ainsi théoriquement un progrès en apprentissage plus élevé que les autres. Ces régions correspondent à la notion de Flow défini dans la section 1.3.2 et aux zones proximales de développement de la section 1.2.2.2.

Le schéma 2.2 illustre les caractéristiques des différentes régions.

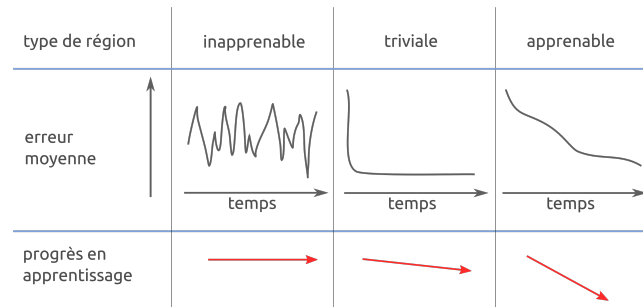


FIGURE 2.2 – Illustrations des différents types de régions en terme d'apprenabilité.

2.2.2 Représentation des régions

Le mécanisme de découpe de RIAC vise à séparer l'espace sensorimoteur en régions. Les différentes régions sont représentées par une structure en arbre appelé Arbre-kd². Il s'agit d'un arbre binaire où chaque noeud représente une zone de l'espace. La racine de l'arbre représente l'ensemble de l'espace. Les deux fils d'un noeud correspondent aux deux sous-régions présentes de part et d'autre d'un hyper-plan intersectant la région mère. Dans le cadre de RIAC on restreint le plan à être parallèle à l'une des dimensions. Chaque noeud est ainsi le

produit cartésien d'intervalles ou en d'autres mots correspond à un hyper-rectangle. Un exemple de découpe dans une zone à une dimension sensorielle et une dimension moteur est montré dans la figure 2.3. Chaque feuille

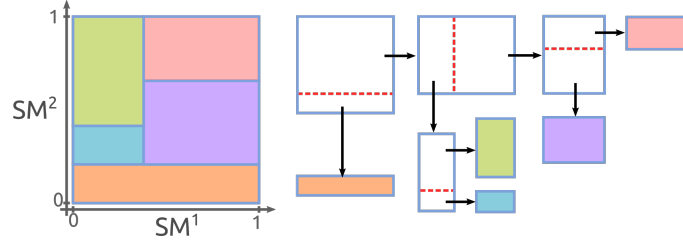


FIGURE 2.3 – Illustrations des régions de l'espace sensorimoteur et de la structure en arbre.

correspond à une région n'ayant pas actuellement de sous-région.

Cette structure en arbre permet de séparer en deux l'espace dans une dimension avec un seul paramètre. Elle facilite également les calculs notamment pour identifier à quelle région appartient un point.

2.2.3 Mécanisme de choix d'action

L'objectif de cette partie de l'algorithme est étant donné un état sensitif $S(t)$ d'associer une commande $M(t)$ à réaliser afin d'entraîner et d'évaluer le module d'apprentissage. L'objectif étant de favoriser les actions résultant à un vecteur sensorimoteur $SM(t)$ optimisant l'apprentissage. Le mécanisme de choix d'action comporte trois modes avec une probabilité paramétrable d'être effectués. Des paramètres conseillés pour les pourcentages sont donnés dans la définition de chaque mode. Les probabilités sont données dans la définition de RIAC mais l'on pourrait les considérer comme des paramètres de RIAC.

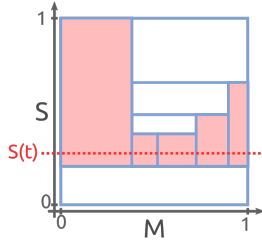


FIGURE 2.4 – Illustration d'un découpage dans le cadre d'un espace sensorimoteur à deux dimensions et des régions (en rose) contenant une valeur dans l'espace sensitif à un instant t .

2.2.3.1 Premier mode

Ce mode s'effectuera avec une probabilité de $p_1 \approx 60\%$. Dans un premier temps, une région est choisie par un tirage aléatoirement pondéré par son progrès. L'ensemble des régions à considérer R est limité, on ne considérera que les régions coupées par l'hyper-plan donné par $S(t)$ dans $SM(t)$ (hyper-plan de même dimension que M). Le schéma 2.4 illustre l'ensemble des régions (uniquement feuilles) à considérer. À noter que cet ensemble comporte également des noeuds, notamment tous les parents d'un fils coupé par l'hyper-plan. Ce qui a pour objectif de réduire l'impact d'une mauvaise découpe. La probabilité p^r du choix d'une zone r est :

$$p^r = \frac{LP^r - LP_{min}}{\sum_{i=1}^{|R|} (LP^i - LP_{min})}$$

avec r le nombre de régions considérées et LP_{min} le progrès en apprentissage le plus bas de toutes les régions, soit $LP_{min} = \min(\{LP^i | i \in [1, |R|]\})$. Une fois la région sélectionnée, on tire uniformément dans les dimensions motrices de la région sélectionnée.

2.2.3.2 Deuxième mode

Le deuxième mode a lieu avec une probabilité de $p_2 \approx 10\%$, on effectue comme dans le premier mode un choix d'une zone dans laquelle on choisira une action. L'action ne sera ici pas tirée au hasard mais l'on sélectionnera l'action résultant à un vecteur sensorimoteur maximisant l'erreur en prédiction. Pour cela, on utilise une méthode de régression dont le choix est laissé libre, approximant la fonction $SM(t) \rightarrow e(t+1)$ à partir des données de la région. Ce mode permet d'évaluer le progrès en apprentissage aux endroits où la marge de progrès est la plus forte. Si la région est prédictible, on peut supposer que l'on aura appris et que l'erreur sera désormais faible. Dans le cas d'une région inapprenable, l'erreur sera toujours élevée.

2.2.3.3 Troisième mode

Le dernier mode consiste à tirer uniformément dans l'ensemble l'espace moteur. Ce mode est utilisé avec une probabilité de $p_3 = 1 - p_1 - p_2 \approx 30\%$. Il permet de palier à d'éventuelles mauvaises découpes et continuer à mettre à jour le progrès dans chacune des régions même celles avec un progrès en apprentissage actuellement bas qui ne serait pas considéré par l'algorithme.

2.2.4 Mécanisme de découpe

Lorsque T_{split} exemples sont tirés dans une feuille, la zone en question est alors découpée en deux régions filles. Afin de déterminer quelle découpe sera effectuée, on réalise n découpes virtuelles dans chaque dimension. Dans une dimension d_i , les découpes s'effectuent selon n hyper-plan parallèle aux dimensions autres que d_i dans SM et régulièrement répartie la dimension d_i . Afin d'évaluer la qualité d'une découpe on mesure le progrès en apprentissage des deux sous-régions d'une zone r , respectivement LP_1^r et LP_2^r . On calcule alors la qualité d'une découpe $D^r = |LP_1^r - LP_2^r|$. L'objectif étant de couper entre deux régions qui possèdent la plus grande dissimilarité de progrès en apprentissage ; et ainsi pouvoir se concentrer sur la zone possédant le meilleur progrès en apprentissage.

Chapitre 3

Expérimentation de RIAC et analyses

Afin de pouvoir analyser et mesurer l'efficacité de l'algorithme, j'ai réalisé une implémentation de RIAC en python. Cette implémentation reprend une partie du code d'explauto, librairie reprenant les principes de RIAC afin de pouvoir réaliser des expériences simulées ou sur des robots réels. Le code a subi de nombreuses modifications afin de permettre l'acquisition de données et correspondre parfaitement à l'algorithme RIAC.

3.1 Description de l'environnement

Afin de pouvoir réaliser suffisamment d'expériences et ne pas perdre de temps dans des problèmes techniques, j'ai choisis de ne travailler que sur des simulations d'expériences. L'essentiel de mes recherches s'est appuyé sur l'expérience simulée suivante inspirée d'une expérience décrite dans la thèse d'Adrien Baranes [2]. Le robot est constitué d'un bras à m moteurs évoluant dans un espace à deux dimensions. Chaque partie du bras est de même longueur. La donnée sensitive acquise par le robot correspond à la position de l'extrémité du bras dans l'espace à deux dimensions. La figure 3.1 représente un bras à quatre dimensions. La spécificité du robot réside dans l'efficacité des moteurs, en effet ils ne réalisent pas forcément la commande en radian demandée. Si l'angle dépasse un seuil θ , le moteur tourne d'un angle aléatoire dans $[-\pi, \pi]$. Pour toutes les expériences effectuées par la suite, nous prendrons $\theta = 1.5$ (en radian). La figure 3.2 montre l'angle réellement effectué par un moteur en fonction de l'angle commandé.

Il s'agit d'une simplification de l'expérience de base où le caractère aléatoire était proportionnel à l'angle demandé. L'objectif étant de simuler le fait que l'imprécision d'une commande augmente pour de grandes valeurs. Cette expérience rend néanmoins l'analyse et la prédiction des résultats attendus plus complexes. L'utilisation d'un seuil soudain permet de supposer que les coupes les plus judicieuses se situent au niveau de ce seuil.

Afin de simplifier l'expérience, l'on considère que le bras retourne automatiquement à son état initial après avoir été déplacé. Cela revient à ne pas avoir d'espace sensitive $S(t)$ et ainsi $SM(t) = M(t)$. Dans cette expérience on apprend ainsi la relation entre un espace sensorimoteur à m dimensions et un espace sensitive à deux dimensions.

Afin d'étudier en détail RIAC on s'intéressera dans la suite ce rapport à différentes régions de l'espace moteur d'environnement avec un bras à deux moteurs. Afin de faciliter la compréhension, le schéma 3.3 attribut un numéro à chacune des régions étudiées.

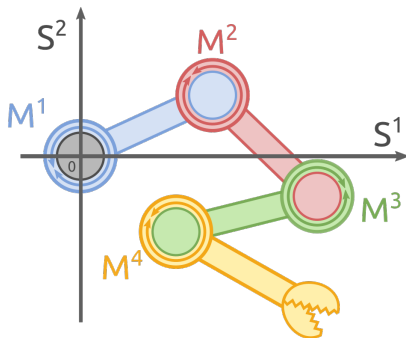


FIGURE 3.1 – Schéma d'un bras à quatre moteurs

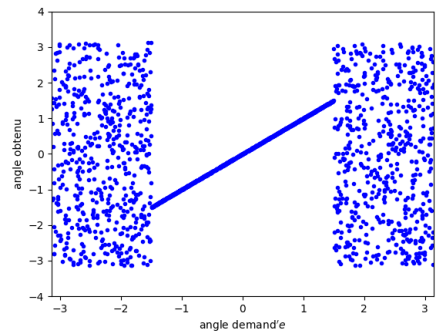


FIGURE 3.2 – Angle réalisé par moteur en fonction de l'angle demandé

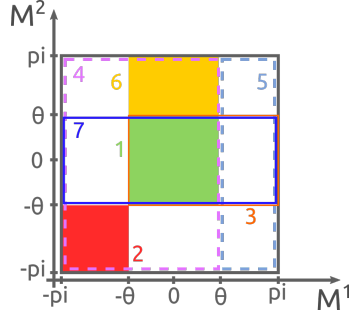


FIGURE 3.3 – Représentation de différentes régions dans l'espace moteur d'un bras à deux moteurs. La première régions correspond à la zone proximale de développement 1.2.2.2. Dans la deuxième région, tout les moteurs sont totalement bruité. La troisième région ne nécessite qu'une découpe afin d'isoler la région 1. Les régions 4 et 5 correspondent à deux régions que RIAC pourrait séparer en premier. La région 6 possède un moteur bruité et un moteur stable. La région 7 présente une région où seule une des dimensions reste à découper afin d'isoler la zone 1

3.2 Expérimentations de RIAC

3.2.1 Note sur l'expérience

La plus value de RIAC étant d'orienter l'apprentissage vers les régions où l'on apprend le plus, utiliser un tirage uniforme comme expérience témoin permet d'évaluer la plus value de RIAC. Les résultats de la thèse [2] présente des résultats où l'algorithme de prédiction du plus proche voisin (1.1.4) permet de constater le bénéfice de RIAC. Il s'agit de l'algorithme le plus naïf pour prédire des résultats et il est rapide à implémenter. L'étude ne portant pas dans un premier temps sur l'algorithme de prédiction utilisé, j'ai donc opté pour le plus simple.

Afin d'évaluer les deux algorithmes, on sélectionne un nombre statistiquement significatif de points qui seront utilisés pour chaque expérience afin de pouvoir être comparés. Ces points sont tirés dans la région centrale de l'espace sensorimoteur, qui correspond à la zone où tout les moteurs se comportent de manière prédictible. Afin de comparer les deux algorithmes, il faut comparer la dynamique d'apprentissage et ainsi réaliser plusieurs mesures dans le temps. Comparer la dynamique permet d'identifier si l'un des algorithmes mène vers une meilleure erreur moyenne où s'il apprend plus rapidement. Plusieurs mesures sont également nécessaires par algorithme afin d'être statistiquement significatif (environ 20).

Les tests de qualité de RIAC tentent de reproduire ceux obtenus dans la thèse [2]. Les paramètres choisis pour chaque expériences correspondent aux valeurs permettant d'obtenir les meilleurs résultats. Si les probabilités de choix des modes de RIAC ne sont pas spécifiés, il on la valeur par défaut. L'implémentation que j'ai fourni n'a pas permis de reproduire les résultats de la thèse. Les raisons de ces différences n'ont pas pu être identifiées en raison du manque de détails fournis pour ces expériences. Le nombre d'expériences effectuées pour les résultats de la thèse ne permet notamment pas dans mon implémentation d'obtenir des résultats statistiquement significatifs.

3.2.2 résultats

Différents paramètres entrent en jeu dans le paramétrage de RIAC, le nombre d'exemples dans une région avant une découpe T_{split} , la taille fenêtre de calcul du progrès en apprentissage k_{max} . Et également dans ceux de l'expérience, le seuil de bruit θ et le nombre de moteur m . Afin de s'assurer de la conclusion suivante, il est nécessaire de faire varier ces différents paramètres afin de montrer le résultat qui contredit le plus la conclusion. Ainsi, le paramétrage de RIAC décrit dans la figure 3.4 correspond aux meilleurs résultats obtenus par RIAC en comparaison avec un tirage uniforme. On constate que la marge statistique correspondant à la déviation standard des deux courbes ne permet pas de confirmer que RIAC est plus efficace que le tirage uniforme. On constate néanmoins que RIAC se trouve en moyenne en dessous du tirage uniforme.

Afin de comprendre pourquoi RIAC ne manifeste pas une amélioration significative par rapport à une méthode naïve. Il est ainsi nécessaire d'analyser en détail les données issues des expériences.

3.3 Analyse du fonctionnement de RIAC

L'algorithme RIAC ne remplit pas actuellement son objectif, une des raisons serait que l'un des mécanisme ne répond pas aux attentes. Il se peut également que ce soit le cumul des mécanisme qui soit la cause de ces résultats.

3.3.1 Erreur moyenne par régions

On peut dans un premier temps observer l'évolution de l'erreur moyenne afin de vérifier que l'algorithme du plus proche voisin apprend convenablement pour cette expérience. La théorie veut que dans les régions centrales (région 1 dans le schéma 3.3), l'erreur moyenne décroît de manière significative. Que dans les régions bruitées

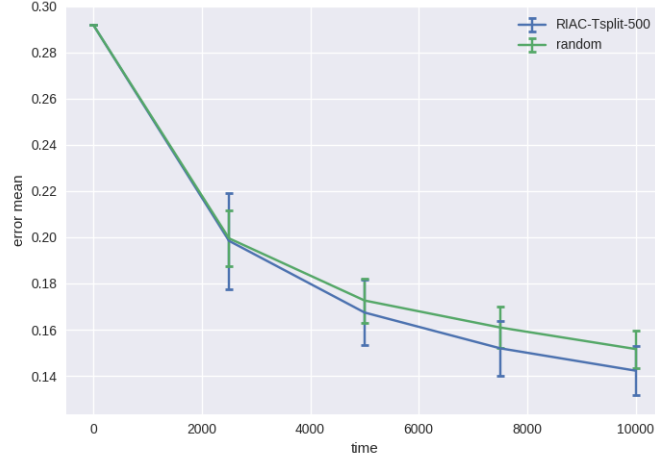


FIGURE 3.4 – Comparaison de RIAC avec un tirage uniforme au cours du temps

aucune tendance ne soit détectable (2 et 6). Et que dans les régions mixtes (3, 4 et 5), correspondant aux régions en attente d’être découpées, on constate une diminution légère de l’erreur mais présente perceptiblement.

Les résultats sont concluants concernant les régions homogènes. L’on observe une nette décroissance de l’erreur dans la région centrale et du bruit dans les régions inapprenables. Concernant les régions mixtes que RIAC est censé différencier afin de réaliser les premières coupes, la différence n’est pas perceptible grâce aux courbes.

3.3.2 Progrès en apprentissage par régions

L’erreur moyenne n’étant pas directement utilisée on peut se poser plus directement les questionnements de la partie précédente par rapport au progrès en apprentissage. On s’attend notamment à observer un meilleur progrès dans la région centrale.

De manière cohérente avec les résultats précédents le progrès en apprentissage de la région centrale est meilleur que celui des autres régions. Mais cette différence est moins notable que la comparaison visuelle des courbes d’évolution de l’erreur effectuées dans la section 3.3.1. Les régions correspondant à celles obtenues par une première découpe idéale (régions 4 et 5 dans le schéma 3.3 sont par exemple indifférenciables en observant le progrès en apprentissage. Le calcul du gradient est en effet très sensible au bruit constituant la courbe d’erreur moyenne. Ces analyses nous permettent également d’identifier la taille de la fenêtre de calcul du progrès en apprentissage la plus à même de différencier les zones. Ils permettent également d’identifier le nombre d’exemples nécessaires à la découpe et ainsi la valeur de T_{split} . Ces valeurs correspondent aux valeurs donnant les meilleurs résultats de la section 3.2.2 où l’on compare RIAC à un tirage uniforme (T_{split} : 500, k_{max} : 1000).

3.3.3 Efficacité du mécanisme de choix d’action

Les résultats donnés par RIAC n’étant pas satisfaisants, j’ai souhaité identifier quelle partie du mécanisme ne donnait pas les résultats attendus. Il est assez facile d’identifier si le mécanisme de choix d’action seul est efficace. Il suffit de réaliser manuellement les coupes que RIAC devrait effectuer, de désactiver le mécanisme de découpe afin que de nouvelles coupes ne soient pas effectuées. En prenant le pari que les coupes effectuées correspondent bien aux coupes les plus judicieuses, ce qui devrait, par construction de l’expérience être le cas.

Puis d’évaluer grâce au protocole décrit dans la section 3.2.1 cette version frauduleuse de RIAC. On obtient de cette manière de très bons résultats, comme le témoigne la figure 3.5. On peut ainsi en conclure qu’avec une découpe immédiate et parfaite, RIAC peut-être performant. En réalisant à la main les coupes successivement dans le temps on obtient également ces résultats écartant l’hypothèse que les résultats sont concluants grâce à l’aspect immédiat des coupes.

3.3.4 Analyse du mécanisme de découpe

3.3.4.1 Analyse des coupes effectuées dans un espace à deux dimensions

Afin d’évaluer la fiabilité du mécanisme de découpe, il est judicieux dans un premier temps d’observer les régions créés par RIAC. Afin de pouvoir visualiser les coupes, j’ai réalisé les expériences avec un robot à deux

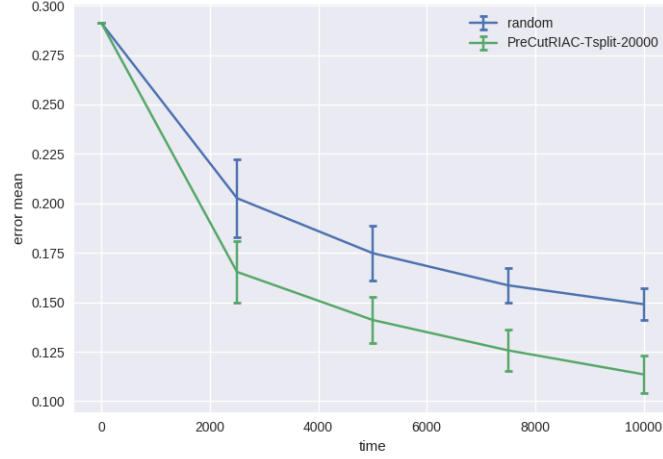


FIGURE 3.5 – Comparaison de RIAC avec une découpe pré-fixée avec un tirage uniforme au cours du temps

moteurs. On observe grâce au schéma 3.6 que les découpes ne correspondent que très peu à celles attendues. Certaines découpes correspondent bien à la valeur attendue, mais il serait statistiquement improbable que ce ne soit jamais le cas.

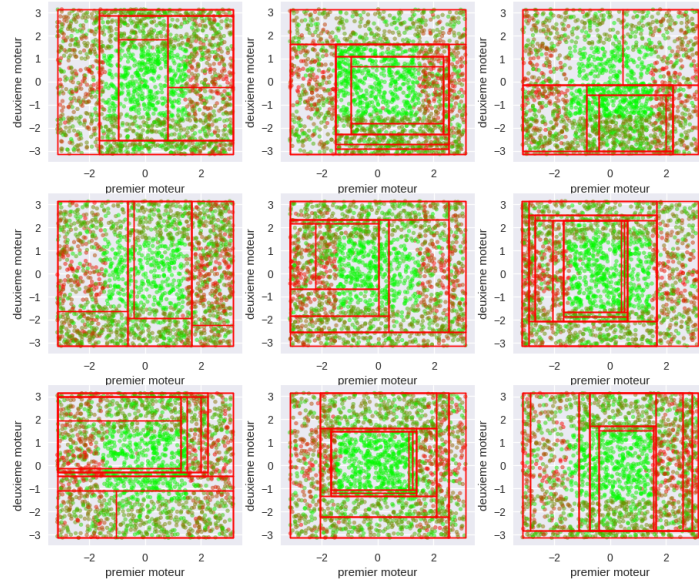


FIGURE 3.6 – Affichage des découpes effectués par RIAC et de l'erreur de prédiction de chaque commande testée. $T_{split} : 500$, $k_{max} : 500$, $m : 2$, $\theta : 1.5$

3.3.4.2 Différence du progrès en apprentissage

Afin d'identifier au mieux pourquoi les découpes ne sont pas assez pertinentes, il faut visualiser la qualité des différentes découpes en concurrence. Qui correspond à la valeur de la différence des progrès en apprentissage des deux régions produites par la coupe, selon le calcul décrit dans la partie 2.2.4. Dans le but de pouvoir évaluer si les résultats correspondent aux résultats théoriques, il convient d'évaluer dans un premier temps quels sont les résultats théoriques attendus. Les résultats théoriques correspondent à la région décrite dans la figure 3.7. L'objectif étant de visualiser les coupes dans une dimension où des découpes devraient être effectuées M^1 et une dimension où les découpes ne sont pas utiles M^2 .

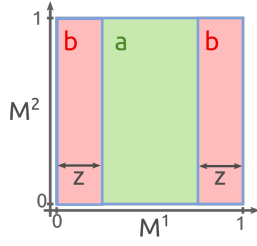


FIGURE 3.7 – Illustration d'un espace moteur à deux dimensions composé de deux régions périphériques en rose avec un mauvais progrès en apprentissage b et d'une zone centrale en vert avec un bon progrès en apprentissage a . Cette région correspond à la région 7 du schéma 3.3. Les tailles des régions dans la première dimension motrice M^1 sont déterminées par une variable $z = \pi - \theta$

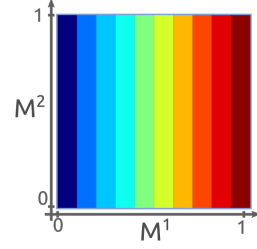


FIGURE 3.8 – Illustration de la notion de tranche dans une dimension dans un espace moteur à deux dimensions. Chaque tranche correspond à une bande de couleur

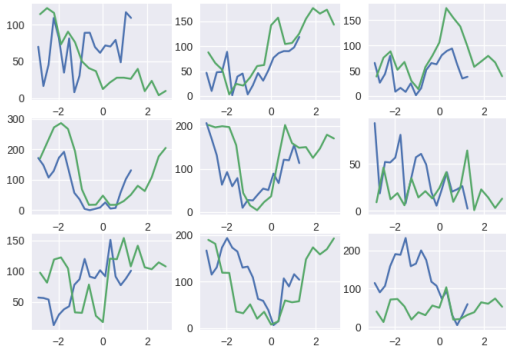


FIGURE 3.9 – Progrès en apprentissage par coupe dans un environnement bruité dans la première dimension à partir de 1.5. La courbe bleue et verte représentent respectivement le premier et le deuxième moteur.



FIGURE 3.10 – Progrès en apprentissage par coupe approximé dans un environnement bruité dans la première dimension à partir de 1.5. La courbe bleue et verte représentent respectivement le premier et le deuxième moteur.

3.3.4.2.1 Hypothèse de conservation de la représentativité temporelle

Afin de calculer les résultats théoriques par coupe, il faut valider l'hypothèse de conservation de la représentativité dans le temps. Il s'agit de valider que dans une sous-zone d'un espace donné l'ensemble des exemples de la sous zone sont répartis régulièrement dans le temps. Si c'est le cas on peut, dans une région R possédant un progrès en apprentissage homogène. Considérer que le progrès en apprentissage d'une sous-région R_1 sera linéairement lié au progrès de la région R par sa taille. La taille de la région influant sur le nombre d'exemples que le tirage uniforme tirera dans la région. Nombre d'exemples qui influera sur la somme présente dans le calcul du progrès en apprentissage. Afin de valider cette hypothèse en pratique, on peut calculer la dis similarité de progrès en apprentissage de chaque coupe de la manière décrite dans RIAC. Et réaliser le calcul en calculant le progrès en apprentissage pour chaque tranche de dimension. Tranche correspondant à chaque région différenciant deux coupes successives comme l'illustre le schéma 3.8. On peut alors calculer le progrès en apprentissage approximé d'une région comme la somme de tout ses progrès si la théorie s'avère correcte. En réalisant la différence de ces progrès en apprentissage approximés, on obtient une approximation de la qualité d'une découpe. En comparant les valeurs approximatives 3.9 aux valeurs réelles 3.10, on obtient dans les simulations des courbes comparables. Ce résultat conforte la validité de l'hypothèse et permet de réaliser des courbes théoriques de la qualité des découpes.

3.3.4.2.2 Comparaison entre la pratique et la théorie

Les résultats théoriques de la qualité des découpes dans l'espace décrit précédemment produisent les courbes de la figure 3.11. Ce lien permet de visualiser les courbes en faisant varier les paramètres. On constate que si le progrès dans la zone où le progrès bas est négatif, la qualité de la découpe maximale devrait correspondre à l'une des découpes souhaitées. Si le progrès de cette région est positif, le mécanisme de découpe risque de réaliser une découpe à l'intérieur de cette zone, ce qui aura pour effet de réduire le bruit ce qui est un résultat acceptable.

Pour les résultats pratiques j'ai choisis de travailler dans une zone où seul un côté de la région est encore bruité

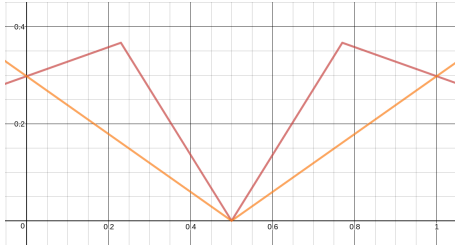


FIGURE 3.11 – Courbes théoriques de la qualité par coupe pour chaque moteur en utilisant le progrès en apprentissage.

(région 3 du schéma 3.3). Les calculs sont réalisés sur un grand nombre d'exemples afin de pallier à des problèmes de nombre d'exemples non-statistiquement significatif. Problème pouvant particulièrement se poser dans l'utilisation réelle de RIAC, notamment dans les coupes proche du bord de la région à découper. La région bruitée commence dans la dimension du premier moteur pour une valeur supérieure à 1.5. Cela simule le cas où seule une dernière découpe dans la dimension du premier moteur reste à réaliser. Les résultats pratiques de la figure 3.12 ne correspondent pas du tout aux résultats théoriques où un pic devrait apparaître dans les environs de 1.5. Ces résultats confortent l'idée que le progrès en apprentissage est une donnée trop sensible au bruit pour permettre une découpe efficace.

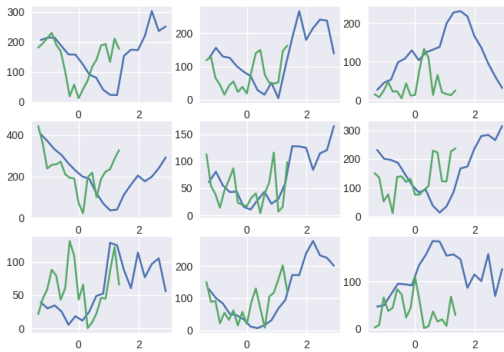


FIGURE 3.12 – Courbes expérimentales de la qualité par coupe pour chaque moteur en utilisant le progrès en apprentissage. La courbe bleu correspond au premier moteur. La courbe verte correspond au deuxième moteur

3.3.4.2.3 Conclusion

On peut conclure des différentes analyses que le progrès en apprentissage est une bonne donnée pour évaluer l'intérêt des différentes régions comme le confirme la partie 3.3.3. L'utilisation du progrès en apprentissage est d'ailleurs nécessaire afin d'éviter à la fois les régions inapprenables et les régions triviales. Le progrès en apprentissage ne permet néanmoins pas de réaliser des coupes pertinentes et nous proposons dans le chapitre suivant différentes propositions pour évaluer les différentes coupes.

Chapitre 4

Proposition de nouvelle mesure pour la découpe

Les résultats obtenus dans la section précédente témoignent d'une faiblesse de RIAC au niveau de la mesure utilisée pour la découpe. Au cours de mon stage j'ai émis plusieurs propositions de mesure dans l'espoir d'obtenir des découpes plus satisfaisantes. Nous décrirons dans ce chapitre, deux propositions de mesure pour la découpe et nous donnerons les résultats applicatifs.

4.1 Découpe en utilisant le progrès pondéré

4.1.1 Principe

La première proposition part d'une hypothèse concernant la pertinence de comparer directement le progrès en apprentissage de deux régions des coupes potentielles. Le calcul du progrès en apprentissage fait intervenir une somme sur le nombre d'exemples. Si la taille de t_{max} n'est pas suffisamment petite pour borner le nombre d'exemples, le nombre d'exemples dans chaque région sera différent. Ce phénomène s'amplifie pour les coupes proches du bord de la région à découper. Si la théorie 3.11 laisse penser que cela n'aura pas d'impact, les résultats pratiques 3.12 rendent caduques les attentes théoriques.

J'ai donc proposé de pondérer le calcul du progrès par le nombre d'exemples de la région. J'ai appelé cette nouvelle mesure progrès en apprentissage pondéré (LPD).

$$LPD^r = \frac{\sum_{i=|E^r|-2*k^r}^{|E^r|-k^r} E_i^r - \sum_{i=|E^r|-k^r}^{|E^r|} E_i^r}{k^r}$$

L'espérance de ce calcul est que la pondération compense les effets du bruit et permette de converger vers les résultats théoriques.

4.1.2 Résultats théoriques et pratiques

Comme dans la section 3.3.4.2.2 on peut estimer la forme théorique de la différence absolue du progrès en apprentissage pondéré par coupe. L'espace utilisé pour ces résultats est le même que celui décrit dans le schéma 3.7. Les variables a et b sont également conservées, car le progrès en apprentissage pondéré peut se calculer à partir de celui non pondéré.

Les courbes de la figure 4.1 correspondent aux valeurs de la différence absolue du progrès en apprentissage pondéré pour chaque coupe. Ce lien permet de visualiser les courbes en faisant varier les paramètres.

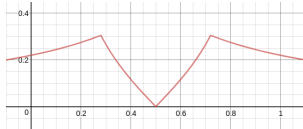


FIGURE 4.1 – Courbes théoriques de la qualité par coupe pour chaque moteur en utilisant le progrès en apprentissage pondéré. La courbe rouge correspond au premier moteur. La courbe bleue est confondue avec zéro et correspond au deuxième moteur

Les résultats théoriques sont plus satisfaisants que ceux obtenus pour le progrès en apprentissage de base. Dans la dimension où les coupes sont inutiles la courbe est à zéro. Il s'agit ainsi du meilleur résultat possible. Dans la première dimension les résultats sont également plus satisfaisants dans le cas où le progrès de la zone la

moins apprenable est tout de même positif. Grâce à ce calcul les meilleurs endroits de découpe correspondent bien aux plus grandes valeurs de qualité de découpe.

La figure 4.2 contient les valeurs en pratique dans la zone 3 du schéma 3.3 dans différentes expériences indépendantes. Le grand nombre d'exemples tirés (10 000) fait tendre la qualité des découpes vers 0. Mais l'on ne retrouve pas les résultats théoriques et les découpes seront également proches de l'aléatoire.

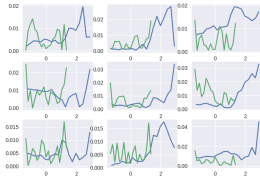


FIGURE 4.2 – Courbes expérimentales de la qualité par coupe pour chaque moteur en utilisant le progrès en apprentissage pondéré. La courbe bleu correspond au premier moteur. La courbe verte correspond au deuxième moteur

4.1.3 Découpe en utilisant l'erreur moyenne

4.1.4 Principe

Le principal argument à l'utilisation du progrès en apprentissage est celui illustré par le schéma 2.2. Cet argument ne justifie que l'utilisation du progrès en apprentissage, on peut donc se permettre d'utiliser d'autres mesures pour la découpe. On peut notamment s'autoriser à tenter la pertinence de l'erreur moyenne. Il semble cohérent de supposer que si deux zones ont un progrès en apprentissage différent, elles possèdent également une erreur moyenne différente. Il est évidemment possible que ce ne soit pas le cas à un instant t , mais les différences de progrès en apprentissage impliquent que ce ne sera plus le cas à un instant $t + 1$.

4.1.5 Résultats théoriques et pratiques

Comme dans la section précédente, on peut approximer la qualité des découpes par coupes. L'espace utilisé pour ces résultats est le même que celui décrit dans le schéma 3.7. Les variables a et b quant à elles ne représentent plus le progrès en apprentissage mais l'erreur moyenne. Cela implique notamment que a et b ne peuvent plus être négatifs. Les courbes de la figure 4.3 correspondent aux valeurs de la différence absolue du progrès en apprentissage pondérée pour chaque coupe. Ce lien permet de visualiser les courbes en faisant varier les paramètres.

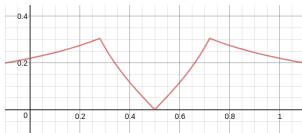


FIGURE 4.3 – Courbes théoriques de la qualité par coupe pour chaque moteur en utilisant l'erreur moyenne. La courbe rouge correspond au premier moteur. La courbe bleue est confondue avec zéro et correspond au deuxième moteur

Les résultats théoriques sont équivalents à ceux obtenus pour le progrès en apprentissage pondéré. Pour les mêmes raisons ils sont ainsi très satisfaisants et devraient permettre si les résultats pratiques concordent de réaliser de bonnes découpes. La figure 4.4 témoigne de la consistance de l'erreur moyennes. La découpe s'effectue toujours à l'endroit voulu. Ces résultats sont obtenus à partir d'un très grand nombre de données. On se place dans un environnement facile à découper car une découpe suffit à séparer la région totalement bruitée de la région apprenable.

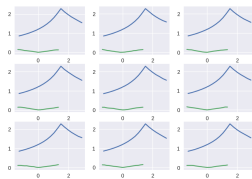


FIGURE 4.4 – Courbes expérimentales de la qualité par coupe pour chaque moteur en utilisant le progrès en apprentissage. La courbe bleu correspond au premier moteur. La courbe verte correspond au deuxième moteur

4.1.6 Découpes obtenues

La figure 4.5 représente les coupes effectuées par différentes expériences indépendantes réalisant les coupes en fonction de l'erreur moyenne. On observe que dans toutes les expériences, les coupes se situent à l'endroit

espéré. La variable n est dans ces expériences égale à 20, cela implique qu'il faut plusieurs découpes dans la même dimension avant d'approximer correctement la valeur θ . La valeur de la fenêtre nécessaire à obtenir ces découpes 50 est très faible ce qui permet de très rapidement obtenir les différentes régions. On constate également que le mécanisme de choix d'action tire bien ces exemples dans la zone centrale.

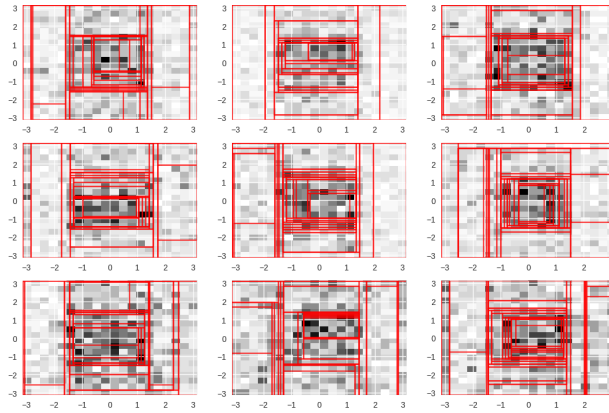


FIGURE 4.5 – Affichage des découpes effectuées par RIAC et de la densité des exemples tirés. T_{split} : 200, k_{max} : 1000, m : 5, θ : 1.5

4.1.7 Tests qualitatifs

Les tests qualitatifs de cette version modifiée de RIAC pour une valeur T_{split} de 50 sont donnés par la figure 4.6. Les résultats témoignent d'un apport significatif de cette version de RIAC utilisant l'erreur moyenne pour découper les régions. On remarque que dans cet version de RIAC désactiver le deuxième mode de choix d'action permet d'obtenir de meilleurs résultats

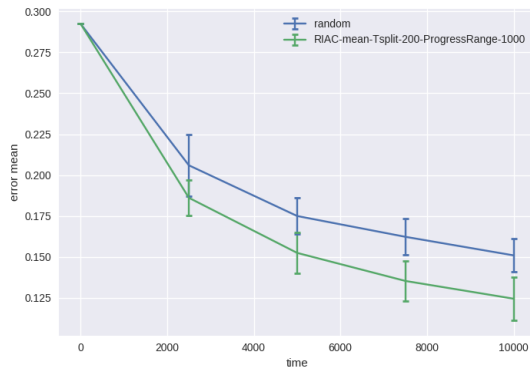


FIGURE 4.6 – Comparaison de RIAC utilisant l'erreur moyenne pour la découpe avec un tirage uniforme au cours du temps. T_{split} : 200, k_{max} : 1000, m : 5, θ : 1.5, $p_1 = 0.7, p_2 = 0, p_3 = 0.3$

4.1.8 Conclusion

Chapitre 5

Conclusion et perspectives

Afin d'obtenir des robots capables d'effectuer un grand nombre de tâches, il est nécessaire qu'ils soient capables d'apprendre de manière autonome. En s'inspirant des sciences cognitives on peut observer plusieurs mécanismes indispensables à l'émergence de l'intelligence. Un robot développant ses capacités de manière incrémentale à partir de l'impact de ses actions simule certains de ces mécanismes. Le robot doit apprendre la relation liant ses actions à l'environnement. Un modèle d'apprentissage est utilisé afin d'approximer la relation. On remarque tout de même que des problèmes liés à l'exploration de l'environnement surviennent indépendamment de la qualité du modèle d'apprentissage. L'environnement dans lequel apprend le robot peut être trop grand et complexe à apprendre sans concentrer son intérêt dans des régions spécifiques. Afin de se restreindre à apprendre au sein de ces régions et en s'inspirant des résultats des sciences cognitives, l'introduction d'un processus de curiosité est pertinente.

RIAC, un méta-algorithme d'apprentissage permet d'orienter l'apprentissage grâce à une formalisation de la notion de progrès en apprentissage. RIAC fonctionne en découpant l'espace en région en se basant sur le progrès en apprentissage. Et choisit d'apprendre dans les régions où l'apprentissage est le meilleur.

Afin d'étudier le fonctionnement de RIAC, j'ai choisi de me concentrer sur une expérience comportant un bras articulé à plusieurs moteurs partiellement incontrôlable. J'ai utilisé un algorithme de plus proche voisin comme machine de prédiction. Cette expérience n'a pas permis de démontrer statistiquement le bénéfice qu'apporte RIAC à l'apprentissage. Après une étude poussée des différents mécanismes en jeu, j'ai identifié que l'utilisation du progrès en apprentissage pour la découpe ne permet pas de trouver les régions pertinentes.

J'ai ensuite émis plusieurs propositions afin d'obtenir de meilleures découpes. La première proposait l'utilisation d'un progrès en apprentissage pondéré mais celui-ci ne fut pas plus performant. Je me suis ensuite éloigné du progrès en apprentissage en proposant l'utilisation de l'erreur moyenne pour effectuer la découpe. Cette mesure permet d'obtenir des découpes très satisfaisantes. Les test comparatifs permettant d'évaluer l'apport par rapport à l'apprentissage non-orienté sont également concluants.

Afin de confirmer l'utilisation de cette nouvelle découpe, plus d'expériences sont nécessaires. Il serait judicieux de tester ce nouvel algorithme sur de nouveaux environnements ainsi qu'avec d'autres algorithmes de prédiction. L'étape suivante consiste à réaliser des expériences dans des environnements réels.

Certains algorithmes de régression ne permettent pas d'apprendre dans des espaces partiellement inapprenables. Dans la thèse [2], il est proposé une méthode pour adapter tous les algorithmes à de tels espaces. Pour cela, on utilise une structure en arbre parallèle et indépendante à celle de RIAC 2.1.2. Utiliser directement la structure d'arbre de RIAC devrait permettre d'obtenir de meilleurs résultats que la structure en arbre de ILO. On pourrait simplement associer à chaque région de RIAC son modèle d'apprentissage. Des solutions plus complexes pour modéliser les régions, je pense notamment à l'utilisation de réseaux neuronaux.

L'utilisation d'un modèle d'apprentissage capable d'apprendre en se reposant sur les précédentes connaissances/compétences acquises permettrait d'expérimenter les mécanismes de RIAC de manière plus fine. En effet dans les modèles actuellement utilisés les connaissances à apprendre ne deviennent pas plus faciles à apprendre. Si tel était le cas on observerait des hausses de progrès en apprentissage dans les zones désormais apprenables.

Il serait de manière plus globale pertinent de coupler RIAC avec des techniques d'intelligences artificielles plus poussées. On peut en effet supposer que l'intelligence des êtres vivants est la conséquence de différents phénomènes intervenant simultanément.

Chapitre 6

Annexe

6.1 Erreur moyenne et progrès en apprentissage par régions

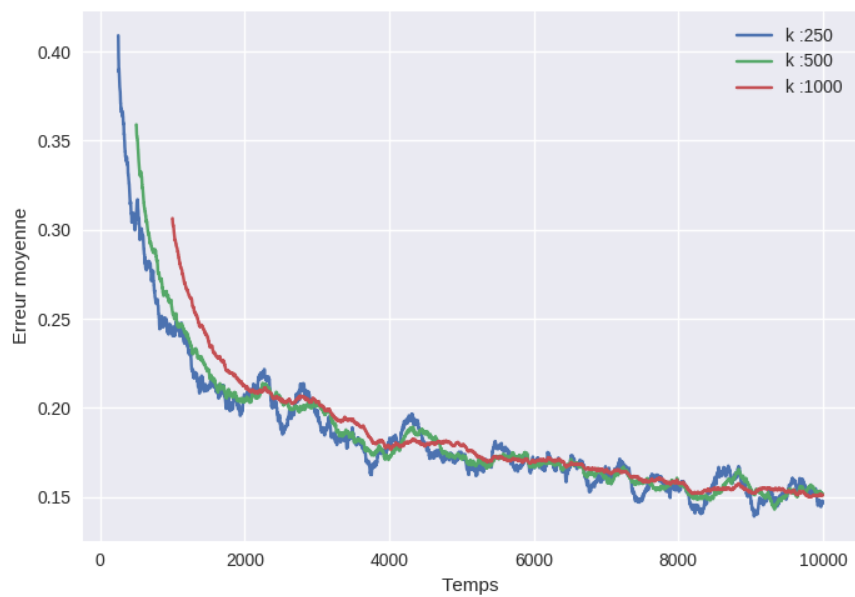


FIGURE 6.1 – Erreur moyenne dans la zone 1 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .



FIGURE 6.2 – Progrès en apprentissage dans la zone 1 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .

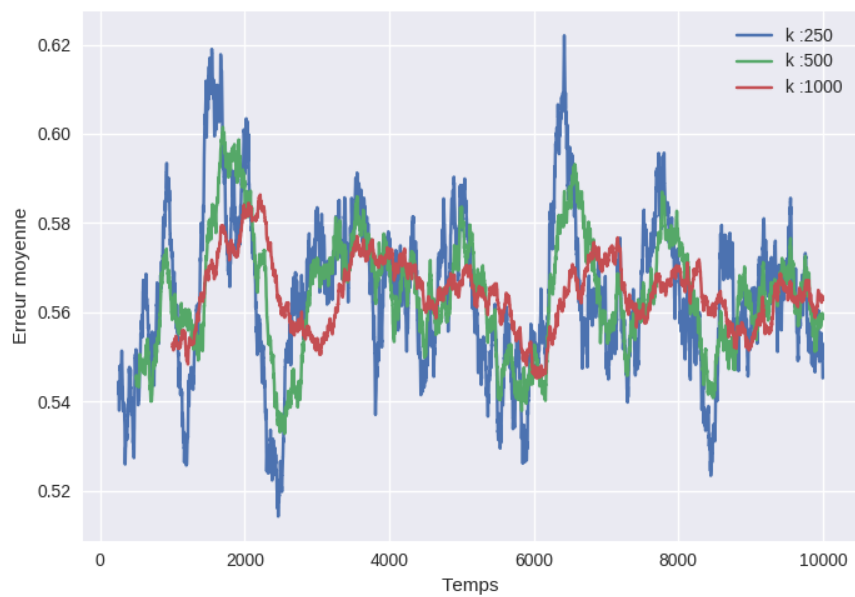


FIGURE 6.3 – Erreur moyenne dans la zone 2 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .



FIGURE 6.4 – Progrès en apprentissage dans la zone 2 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .

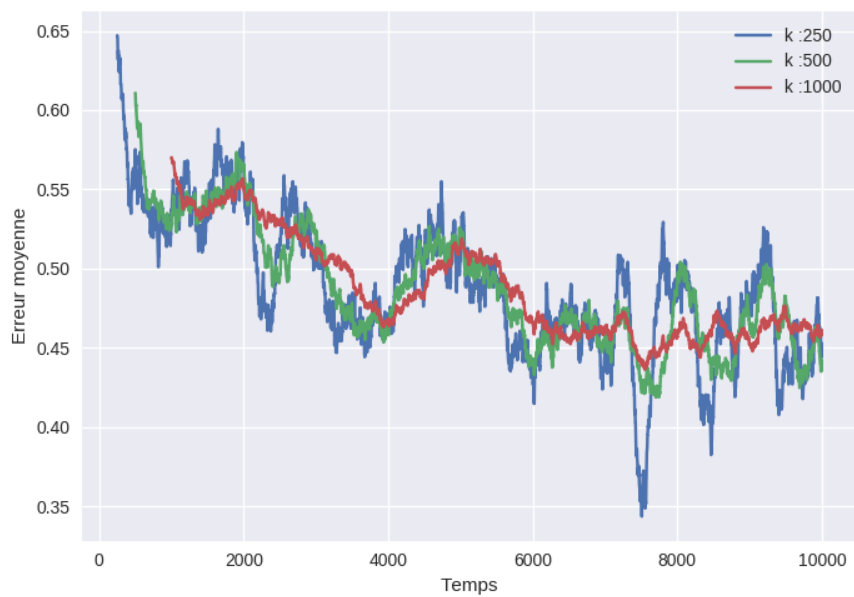


FIGURE 6.5 – Erreur moyenne dans la zone 3 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .



FIGURE 6.6 – Progrès en apprentissage dans la zone 3 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .

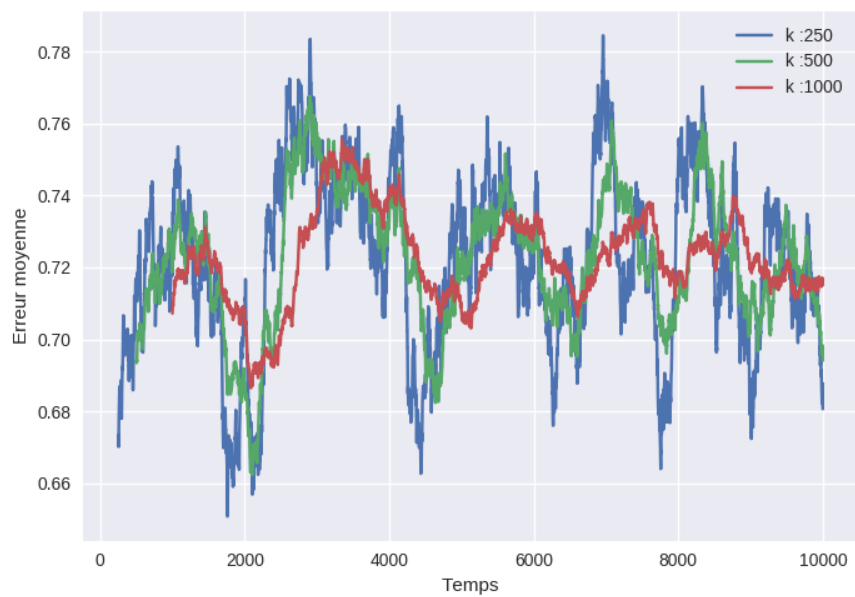


FIGURE 6.7 – Erreur moyenne dans la zone 4 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .



FIGURE 6.8 – Progrès en apprentissage dans la zone 4 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .



FIGURE 6.9 – Erreur moyenne dans la zone 5 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .

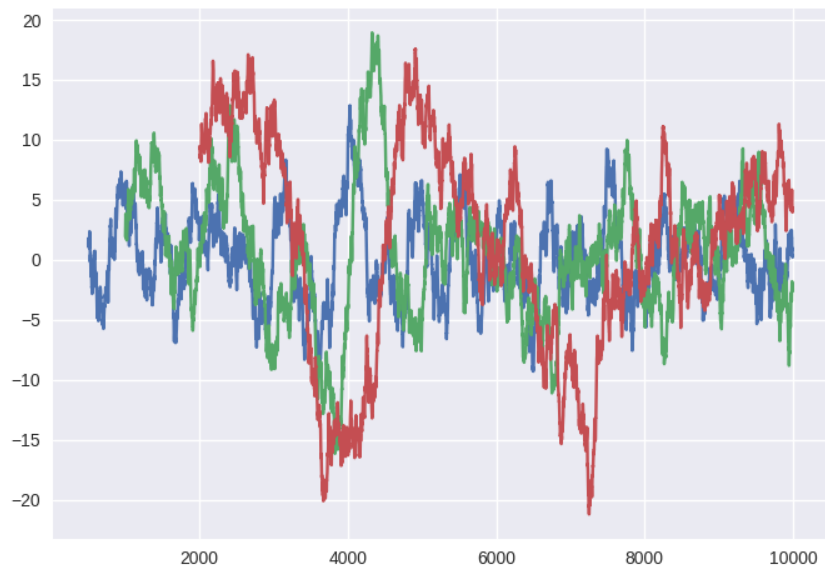


FIGURE 6.10 – Progrès en apprentissage dans la zone 5 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .

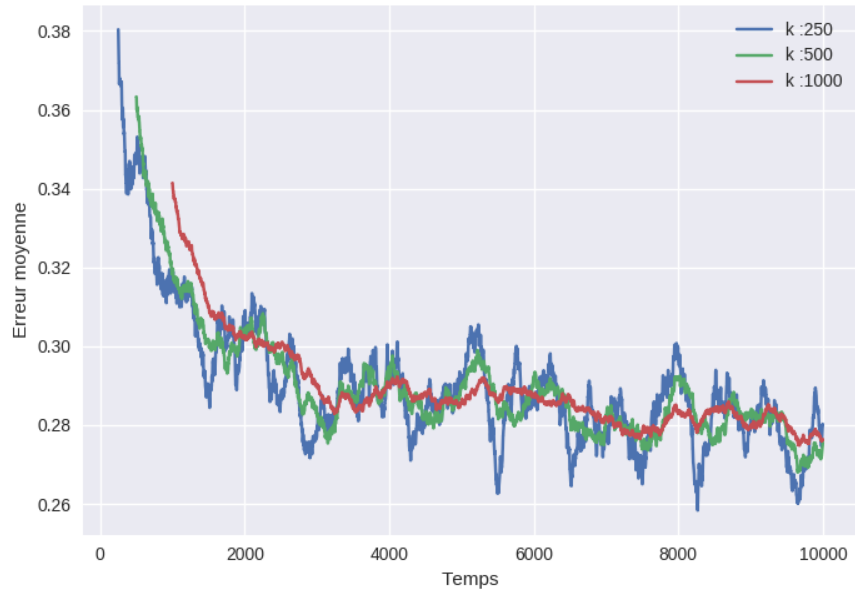


FIGURE 6.11 – Erreur moyenne dans la zone 6 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .



FIGURE 6.12 – Progrès en apprentissage dans la zone 6 correspondant au schéma 3.3 pour différentes valeurs de fenêtre k .

Bibliographie

- [1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 1–, New York, NY, USA, 2004. ACM.
- [2] Adrien Baranes. Motivations Intrinsèques et Contraintes Maturationnelles pour l'Apprentissage Sensorimoteur, December 2011. Thèse de Doctorat de Adrien Baranès.
- [3] N E Berthier, R K Clifton, D D McCall, and D J Robin. Proximodistal structure of early reaching in human infants. *Experimental brain research*, 127 3 :259–69, 1999.
- [4] S. Calinon and A. Billard. A probabilistic programming by demonstration framework handling constraints in joint space and task space. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 367–372, Sept 2008.
- [5] S. Calinon, F. Guenter, and A. Billard. On learning the statistical representation of a task and generalizing it to various contexts. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2978–2983, May 2006.
- [6] Murray Campbell, A. Joseph Hoane, and Feng hsiung Hsu. Deep blue. *Artificial Intelligence*, 134(1) :57 – 83, 2002.
- [7] Justin C. B. Gosling. *Pleasure And Desire : The Case For Hedonism Reviewed*. Oxford, : Clarendon Press, 1969.
- [8] Richard Held and Alan Hein. Movement-produced stimulation in the development of visually guided behavior. *Journal of comparative and physiological psychology*, 56(5) :872, 1963.
- [9] Clément Moulin-Frier, Pierre Rouanet, and Pierre-Yves Oudeyer. Explauto : an open-source Python library to study autonomous exploration in developmental robotics. In *ICDL-Epirob - International Conference on Development and Learning, Epirob*, Genoa, Italy, October 2014.
- [10] J. Piaget. *La naissance de l'intelligence chez l'enfant*. Actualités pédagogiques et psychologiques. Delachaux et Niestlé, 1977.
- [11] F. Rosenblatt. The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.
- [12] Matthew Schlesinger. Heterochrony : It's (all) about time! 2008.
- [13] David J. Slate and Lawrence R. Atkin. *CHESS 4.5—The Northwestern University chess program*, pages 82–118. Springer New York, New York, NY, 1983.
- [14] C. Taylor. *Philosophical Arguments*. Barnard New Women Poets. Harvard University Press, 1995.
- [15] Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236) :433–460, 1950.
- [16] Juyang Weng, James McClelland, Alex Pentland, Olaf Sporns, and et al. Autonomous mental development by robots and animals. *SCIENCE*, 26 :599–600, 2001.