Couplage entre un apprentissage par renforcement profond et une machine à états : approche théorique

Idriss Abdallah*†, Laurent Ciarletta†, Patrick Hénaff†, Jonathan Champagne* and Matthieu Bonavent*
*Naval Group
Gassin, France
†LORIA, CNRS, Université de Lorraine
Nancy, France
Mail : idriss.abdallah@loria.fr

***Abstract*—Dans le cadre de la mise au point de torpilles chez Naval group, les algorithmes de contrôles embarqués sont développés en simulation numérique. Le simulateur intègre plusieurs modèles pour évaluer les performances dans une mise en situation la plus proche possible de la réalité. Parmi tous ces modèles, on distingue le modèle opérateur dont le rôle est de simuler la communication entre le lanceur et la torpille. Cependant, les modèles opérateurs mis en place jusque là grâce à des méthodes symboliques, dont notamment des machines à états, n'ont pas un niveau de représentativité satisfaisants.**

D'un autre côté, l'apprentissage par renforcement profond a récemment montré de très bonnes capacités à résoudre des problèmes de décision séquentielle complexes. Il semble donc, a priori, capable de répondre à cette problématique industrielle. Cependant, cette approche souffre d'une inefficience de l'utilisation de ses données et d'un manque d'interprétabilité dû à l'utilisation des réseaux de neurones.

Nous proposons ici une approche théorique visant à étudier le couplage entre une machine à états et l'apprentissage par renforcement profond afin de tirer profit des connaissances métiers sur l'environnement opérationnel pour pallier certaines difficultés de l'apprentissage par renforcement profond et ainsi obtenir un modèle opérateur à la fois représentatif et explicable.

***Index Terms*—Apprentissage par renforcement, Apprentissage par renforcement profond, Machine à états, Connaissance extérieure, Interprétabilité**

I. Contexte industriel

Naval Group est un acteur international dans le domaine du naval de défense qui conçoit et produit une grande diversité de produits, dont les plus notables sont les bâtiments de surface et les sous-marins. C'est un systémier intégrateur présent sur l'ensemble du cycle de vie de ses produits. Pour l'armement de ces derniers, il s'occupe notamment du développement de torpilles.

Dans le cadre de la mise au point de torpilles, un effort important est porté sur le développement d'algorithmes embarqués de guidage et de prise de décision permettant d'atteindre une cible. Dans le cas étudié, une liaison bi-directionnelle entre le lanceur (i.e. bâtiment mettant en oeuvre l'arme) et la torpille permet une communication entre ces deux acteurs :

- Le logiciel embarqué de la torpille remonte différentes informations du contexte opérationnel à l'opérateur (cinématique, détections acoustiques, ...).
- L'opérateur a la possibilité d'influer sur les décisions de l'intelligence embarquée de la torpille.

En plus des informations remontées par la torpille, l'opérateur dispose également des informations données par les capteurs du lanceur. Le schéma 1 illustre les différents éléments et leurs interactions dans la situation tactique entretenue.

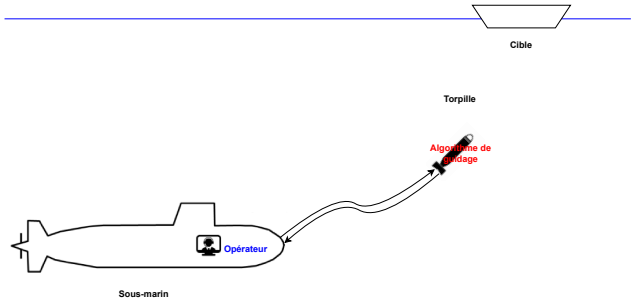


Fig. 1: Contexte industriel

Afin de développer et d'évaluer ces algorithmes embarqués, un simulateur numérique permet de simuler les différents éléments représentés dans le schéma 2. L'environnement doit être aussi représentatif que possible afin d'évaluer de façon fidèle les performances des algorithmes. Il est donc, entre autres, nécessaire de créer un modèle de l'opérateur permettant de recréer les interactions possibles avec l'algorithme au cours d'un tir.

Dans cette optique, plusieurs modèles opérateurs ont été mis au point à partir de méthodes symboliques, mais tous ces modèles offrent une représentativité assez faible tant au niveau du comportement que des performances. Cela est notamment induit par la complexité de la tâche à résoudre et le fait qu'il n'existe pas de modèle applicable simplement contrairement à des modèles physiques. Il est néanmoins intéressant de noter que, par cette voie de modélisation classique, le meilleur modèle opérateur a été créé à partir d'une machine à états.

En accord avec le contexte industriel et l'état de l'art dans le domaine de la décision séquentielle, une approche à base d'apprentissage par renforcement profond a été retenue pour créer un nouveau modèle opérateur. Bien que cette

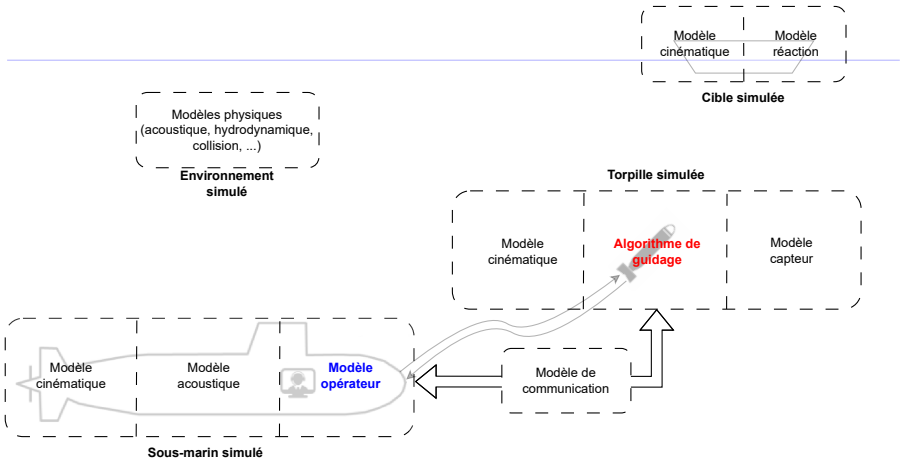


Fig. 2: Environnement de simulation du contexte industriel

méthode semble adéquate pour la mise en place d'un modèle opérateur performant, elle comporte des défauts dont les plus notables sont l'efficacité de l'utilisation des données, ainsi qu'un manque d'interprétabilité du modèle dû à l'utilisation des réseaux de neurones. Or, ces deux aspects sont cruciaux. L'augmentation de l'efficacité de l'utilisation de données permet de réduire les ressources calculatoires nécessaires à l'apprentissage. L'interprétabilité du modèle, quant à elle, est essentielle dans la validation de son utilisation dans un simulateur certifié. Pour pallier à ces difficultés, l'utilisation de la machine à états déjà existante est proposée dans une approche d'apprentissage couplée pour une accélération de la convergence de l'apprentissage vers un modèle performant et pour augmenter l'interprétabilité du modèle final.

Afin d'illustrer les approches proposées, l'environnement classique d'apprentissage par renforcement du Lunar Lander a été choisi afin d'avoir un substitut non sensible au simulateur industriel.

II. ÉTAT DE L'ART

L'apprentissage par renforcement (*Reinforcement Learning* ou RL) est une approche d'apprentissage automatique qui utilise les actions d'un agent dans un environnement pour apprendre un comportement maximisant une récompense obtenue au cours de ses interactions [1]. Cette approche couplée à la capacité d'approximation de fonctions des réseaux de neurones a permis l'apparition du *Deep Reinforcement Learning* (DRL) qui est à la base de plusieurs avancées récentes dans le domaine du contrôle séquentiel [2] [3].

Le succès vient notamment des approches dites "*Model-Free*" qui n'ont pas besoin d'avoir un modèle de transition de l'environnement, mais seulement d'une fonction de récompense. L'utilisation de fonction de récompense est une approche expressive pour définir un problème [4] bien qu'elle soit en pratique difficile à mettre en place, car son design influe fortement sur les capacités d'apprentissage des algorithmes [5]. Cette liberté vient également au prix de la nécessité d'un nombre important d'interactions pour obtenir un comportement optimal, notamment dû au dilemme exploration-utilisation [1]. De plus, ce couplage entre RL et *Deep Learning* entraîne plusieurs défauts inhérents aux réseaux de neurones : explicabilité [6], stabilité et reproductibilité de l'entraînement [7]. Ainsi, il reste encore plusieurs points cruciaux à améliorer afin que le DRL soit applicable à une plus grande variété de problèmes [8].

Une des principales préoccupations de la communauté scientifique du DRL est l'amélioration de la rapidité et de la capacité de convergence des algorithmes. Pour cela, il y a des approches agnostiques de l'environnement à la base de plusieurs concepts clés. Ces méthodes agissent sur des aspects différents des algorithmes [1]. Par exemple, les approches *On-Policy/Off-Policy*, les approches de types *Value-Actor*, *Policy Gradient*, *Actor-Critic*, le rejeu d'expérience [9], une optimisation de l'entropie [10] [11] ou l'ajout de motivation intrinsèque [12]. D'autres approches se concentrent sur l'ajout de connaissances extérieures sur le problème afin de faciliter la convergence vers une politique adéquate. Le format, la quantité et l'utilisation de ces connaissances sont très variables, on peut trouver parmi les approches les plus classiques l'Imitation Learning [13], le *Curriculum Learning* [14] ou bien le *Transfer Learning* [15].

Plusieurs approches fusionnent le DRL avec l'utilisation d'une machine à états finie (MAE) ou un automate déterministe fini. Une première approche est d'intégrer cette MAE dans la structure du problème de DRL en agissant sur l'espace d'observation ou/et d'action avec une approche hiérarchique, avec un macro et un micro contrôleur où l'un des deux est géré par une MAE et l'autre appris par DRL [16] [17]. D'autres approches couplent la MAE et le DRL au niveau algorithmique, c'est le cas des *Reward Machines* en ajoutant une structure de MAE sur la fonction de récompense, ce qui permet notamment d'augmenter l'espace d'observation et d'ajouter un mécanisme de rejeu d'expérience basé sur les états [18]. De plus, une MAE peut directement être apprise

afin d'aider la convergence comme pour les *Subgoal Automata* [19] ou [20]. Ainsi, le couplage d'une machine à états et d'un algorithme de DRL peut améliorer la capacité, la stabilité et l'interprétabilité [21] de l'apprentissage.

Dans cet article, nous nous intéressons à une approche de couplage algorithmique entre un algorithme de DRL et une MAE. Cette dernière représente des connaissances a priori sur l'environnement pour augmenter l'interprétabilité du processus d'apprentissage et potentiellement l'améliorer en permettant d'incorporer des heuristiques exploratoires externes.

Nous présenterons dans un premier temps les définitions nécessaires pour le DRL, les MAE ainsi que pour l'environnement (section III) et les outils utilisés pour les études menées (section IV). Dans un deuxième temps, nous définirons plusieurs approches théoriques pour le couplage d'une MAE avec un algorithme d'apprentissage par renforcement (section V). Puis, nous utiliserons ce formalisme afin de recontextualiser plusieurs méthodes de l'état de l'art (section VI). Par la suite, nous présenterons plusieurs possibilités de couplage identifiées (section VII). Finalement, nous conclurons sur les perspectives envisagées dans le contexte industriel qu'apporte le couplage proposé entre une MAE et un algorithme de DRL.

III. Définitions

A. Apprentissage par renforcement

L'apprentissage par renforcement est un domaine de l'apprentissage automatique où un agent interagit avec un environnement par une action, ce qui induit une transition de l'état de l'environnement. Il reçoit alors une récompense sous forme d'un scalaire qui évalue la transition effectuée. Le but de l'apprentissage par renforcement est d'apprendre grâce à des interactions avec l'environnement, le comportement qui maximise la récompense obtenue au cours de ses interactions.

Le formalisme mathématique utilisé classiquement pour représenter un tel problème est le processus de décision markovien, abrégé MDP pour *Markov Decision Process*. On se placera par la suite dans le cas d'un MDP déterministe. Il est défini par le tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{T}, \mathbf{R}, \rho_0 \rangle$ avec :

- $\mathcal{S}$, l'ensemble des états possibles.
- $\mathcal{A}$, l'ensemble des actions à la disposition de l'agent.
- $\mathbf{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$, une fonction de transition qui régit l'évolution de l'environnement.
- $\mathbf{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$, la fonction de récompense.
- ρ_0, la distribution de l'état initial sur l'ensemble des états

On définit également le gain cumulé réduit, $G_t = \sum_k \gamma^k r_{t+k+1}$, où r_t désigne la récompense reçue à l'instant t, qui permet d'estimer la récompense obtenue sur un certain horizon temporel en fonction d'un facteur de réduction γ. Une fonction de politique π représente alors une fonction de prise de décision qui permet à partir d'une observation, de choisir une action. La résolution du problème est la recherche de π^*, la politique optimale qui maximise le gain cumulé réduit.

B. Machine à états

Dans la suite, nous nous placerons dans le cadre d'une machine à états finie (MAE) définie par le tuple $\langle \mathcal{Q}, \mathcal{U}, \delta, q_0 \rangle$ un ensemble d'états possibles, $\mathcal{Q}$, un ensemble d'observable, $\mathcal{U}$, une fonction de transition, $\delta : \mathcal{Q} \times \mathcal{U} \rightarrow \mathcal{Q}$ ainsi qu'un état initial $q_o \in \mathcal{Q}$.

C. Environnement

Le type d'environnement étudié est une simulation où l'agent n'a accès qu'à une observation dégradée de l'environnement réel dans lequel il évolue. D'un côté, l'observation partielle à disposition du modèle opérateur est l'ensemble constitué des données fournies par ses propres capteurs ainsi que des informations remontées périodiquement par la liaison de communication. D'un autre côté, l'observation globale (i.e. réelle) est la position de la cible. Cette dichotomie est largement présente dans le cas de simulation de contrôle d'un véhicule ou bien de personnages dans un monde fictif où les données comme la position des obstacles ou de l'objectif ne sont observables qu'en fonction de la position de l'agent via des capteurs ou un point de vue [25] [26].

Ce type de problématique peut alors être posé sous la forme d'un processus de décision markovien partiellement observable (POMDP). Il se définit par un tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{T}, \mathbf{R}, \rho_0, \Omega, \mathrm{O} \rangle$ avec :

- $\langle \mathcal{S}, \mathcal{A}, \mathbf{T}, \mathbf{R}, \rho_0 \rangle$ est un MDP défini tel que précédemment.
- $\mathcal{S}$, l'espace d'observation globale.
- Ω, l'espace d'observation partielle.
- $\mathrm{O} : \mathcal{S} \rightarrow \Omega$ permet d'obtenir l'observation partielle à partir de l'observation globale.

Cependant, le formalisme du POMDP induit classiquement l'ajout d'une probabilité sur l'espace des états, communément appelé vecteur de croyance, car ce formalisme s'applique le plus souvent à des problèmes non-markoviens. Dans notre cas, le formalisme permet de traduire une plus grande complexité d'extraction de caractéristiques pertinentes pour la prise de décision à partir de l'observation partielle plutôt qu'à partir de l'observation globale.

Le problème à résoudre est donc de trouver la politique optimale choisissant une action à partir de Ω.

Dans le cas de la définition d'une MAE, l'espace $\mathcal{Q}$ peut être soit $\mathcal{S}$ ou Ω. Dans cette étude, les deux possibilités seront considérées. Les différences sur les possibilités d'utilisation et les intérêts de ces deux approches seront explicitées par la suite.

Pour la suite, nous utiliserons l'environnement Lunar Lander [23] afin d'illustrer simplement les approches présentées. L'espace $\mathcal{S}$ sera le vecteur d'observation physique et l'espace Ω sera construit à partir du rendu graphique de l'environnement. Afin de réduire l'aspect non-markovien, l'observation sera composée d'une concaténation comme proposée dans [2].

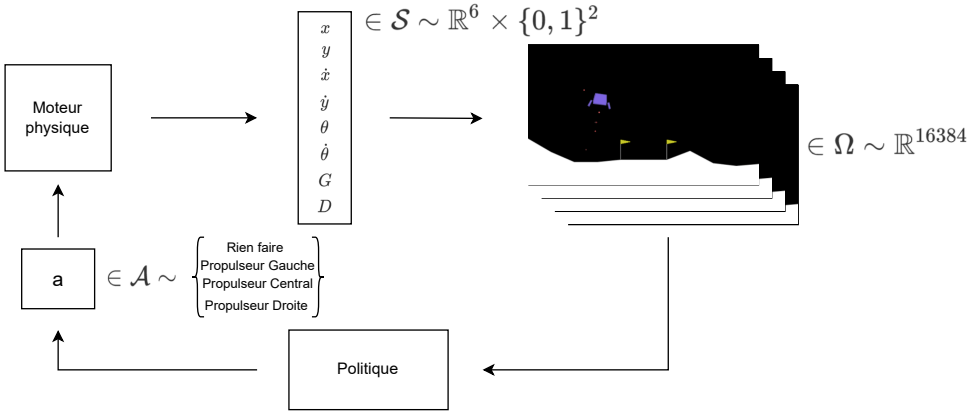


Fig. 3: Relation entre les différents espaces d'observation considérés pour l'environnement Lunar Lander

IV. Outils

A. Environnement

Le problème du Lunar Lander est issu de [23]. C'est un environnement de contrôle discret 2D dont le but est de faire atterrir un vaisseau spatial sur la Lune entre deux drapeaux. Le vecteur d'observation, appartenant donc à $\mathcal{S}$, est de dimension 8 et est constitué de :

- La position : x, y
- La vitesse : $\dot{x}$, $\dot{y}$
- L'angle de roulis : θ
- La vitesse de roulis : $\dot{\theta}$
- Le contact des pieds au sol (deux booléens) : G, D

Son espace d'action est un choix discret entre les 4 actions correspondantes à l'activation d'un de ses 3 réacteurs ou bien de ne rien faire.

Sa fonction de récompense l'encourage à se poser de manière la plus stable possible tout en minimisant l'utilisation de ses réacteurs. Après chaque pas de temps, une récompense est donnée. La récompense totale d'un épisode est la somme de toutes les récompenses obtenues au cours d'un épisode.

Pour chaque pas de temps, la récompense est :

- augmentée/réduite plus le vaisseau est proche/loin de la plate-forme d'atterrissage.
- augmentée/réduite plus le vaisseau est lent/rapide.
- résuite plus le vaisseau à un angle important par rapport à la verticale.
- augmentée de 10 points pour chaque pied qui est en contact avec le sol.
- réduite de 0.03 pour chaque pas de temps où un moteur latéral est utilisé.
- réduite de 0.3 pour chaque pas de temps où le moteur central est utilisé.

A la fin de l'épisode, un récompse finale de -100 ou +100 est donnée en fonction de la réussite ou non de l'atterrissage.

Un épisode est considérée comme étant réussie à partir d'un récompense totale d'au moins 200 points [24] . Les épisodes durant plus de 1000 pas de temps sont tronqués.

Une seconde version du problème est introduite où l'observation est constituée à partir du rendu graphique de l'environnement, en concaténant 4 images consécutives réduites en taille et mise en niveau de gris, ce qui permet de construire les observations issues d'Ω. Le schéma 3 explicite le placement des espaces d'observations $\mathcal{S}$ et Ω dans le cas du Lunar Lander.

B. Machine à états utilisée

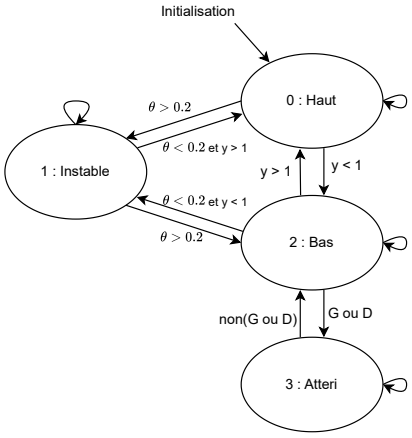


Fig. 4: États de la MAE avec leurs conditions de transition

La MAE utilisée pour la suite a pour but de découper l'espace d'état avec une heuristique simple. L'objectif de ce découpage est d'avoir des états où il est attendu que la politique soit différente afin de pouvoir déceler des difficultés d'apprentissage sur différents ensembles d'états. Les états sont :

- Etat 0 : Haut. Le vaisseau a un angle très faible et est dans la partie haute de l'environnement.
- Etat 1 : Instable. Le vaisseau a un angle non négligeable.

- Etat 2 : Bas. Le vaisseau a un angle très faible et est dans la partie basse de l'environnement.
- Etat 3 : Atterri. L'un des deux pieds du vaisseau touche le sol.

Le schéma 4 représente la MAE ainsi que les conditions de transitions entre ces différents états.

V. Mise en place de la MAE

Nous décomposerons simplement un algorithme de DRL comme étant la succession de deux phases. Une première phase exploratoire de récolte d'expérience et une seconde phase d'apprentissage utilisant ces expériences afin de mettre à jour l'état de la politique qui servira pour créer une nouvelle politique exploratoire pour une nouvelle phase exploratoire. Cette représentation est illustrée par la figure 5. Avec ce découpage simple, nous pourrons étudier plusieurs possibilités de placements de MAE.

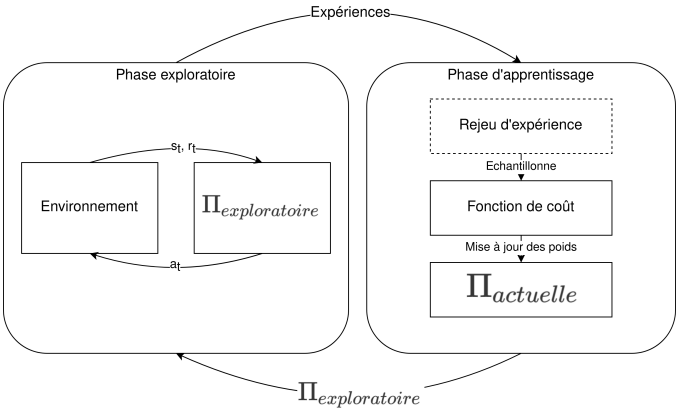


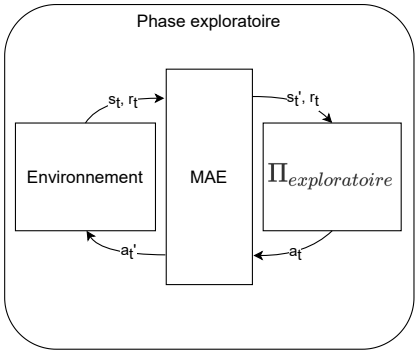
Fig. 5: Schéma générique d'apprentissage par renforcement

A. Définition du couplage de la MAE au sein de l'algorithme de DRL

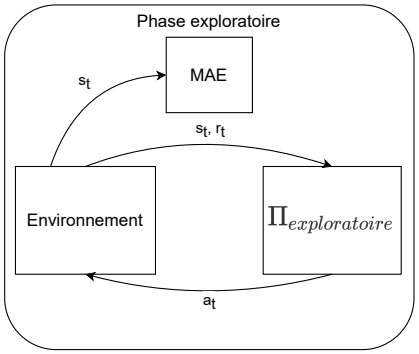
Une première question est si la fonction de transition de la MAE est utilisée lors de la phase d'exploration ou bien lors de la phase d'apprentissage. En effet, les deux placements n'ont pas accès aux mêmes informations.

Une première façon consiste à utiliser la MAE lors de phase d'exploration. Dans ce cas, on s'appuie sur la MAE à chaque interaction avec l'environnement. Elle peut alors être placée en "série", c'est-à-dire qu'elle permet de définir l'espace d'observation et/ou l'espace d'action de la politique apprise par DRL comme décrit par la figure 6a. Elle peut également être placée en parallèle de la politique, c'est à dire qu'elle utilise le même vecteur d'observation que la politique comme décrite par la figure 6b.

Lorsque la MAE est utilisée lors de la phase d'apprentissage, les informations à sa disposition sont l'ensemble des transitions en mémoire et des métriques d'apprentissage calculées.



(a) MAE placée en série



(b) MAE placée en parallèle

Fig. 6: Placement schématique d'une MAE en phase exploratoire

B. Modification algorithmiques permises par ces différents usages de la MAE

Les modifications algorithmiques induites par l'utilisation d'une MAE peuvent être soit en phase exploratoire, soit en phase d'apprentissage.

Pour les modifications en phase exploratoire, celles qui modifient l'espace d'entrée de la politique et/ou le choix des actions, doivent être gardées pour la solution finale. Dans le cas d'une MAE utilisée dans la phase exploratoire, le placement en série implique naturellement ce type de modifications. Pour une MAE placée en parallèle, cela dépend de si elle est utilisée pour augmenter l'espace d'observation et/ou pour changer activement l'action choisie. Ainsi, une MAE utilisant $\mathcal{S}$ et modifiant l'espace d'observation ou d'action ne peut être utilisée comme une solution finale à un problème de décision séquentielle, car cela revient à utiliser des informations non accessibles.

Pour les modifications en phase d'apprentissage, l'utilisation de la MAE a une influence sur la manière algorithmique de mettre à jour les poids du réseau de neurones et/ou sur la politique exploratoire utilisée pour la phase exploratoire. Une différence notable par rapport aux modifications en phase exploratoire est qu'elles n'induisent pas de modifications sur

la définition de la politique finale.

VI. Reconsidération de l'état de l'art au regard des définitions précédentes

A. Fonctionnement et modification en phase exploratoire

Les approches mêlant une MAE avec le DRL de cette façon modifient la problématique initiale de l'environnement en utilisant une MAE pour la définition de l'espace d'observation et/ou d'action de la partie apprise par DRL.

Ainsi, plusieurs approches placent la MAE en série avec une architecture utilisant un macro et un micro contrôleur afin de cibler l'apprentissage sur une partie considérée comme complexe tout en gardant un contrôle et une compréhension sur le modèle final. Par exemple, dans [17], la fonction de politique est apprise pour un seul état de la MAE et garde une définition symbolique pour les fonctions de transition et pour la politique appliquée dans les autres états. Dans [16], la politique apprend directement en utilisant l'état du système mais son espace d'action agit sur une MAE construite à partir d'un contrôleur Proportionnel Dérivé qui transmet des actions plus bas niveau aux actionneurs.

Une utilisation d'une MAE placée en parallèle est proposée par [18]. Tout d'abord, la fonction de récompense est une *Reward Machine* et l'observation est augmentée par l'état courant de la MAE qui est utilisée en parallèle des interactions de la politique afin d'obtenir la récompense. Un résultat intéressant prouvé par [18] est que cette utilisation peut rendre un problème, initialement non markovien, markovien grâce aux informations fournies par l'état courant de la MAE.

B. Fonctionnement en phase exploratoire et modification de la phase d'apprentissage

Ce type d'approche utilise une MAE lors de la phase d'exploration et tire profit des informations données par la MAE sur les transitions afin d'améliorer l'apprentissage.

Les algorithmes les plus notables sont le HRM et le QRM issus de [18] qui profitent tous les deux de la structure de MAE pour augmenter artificiellement le nombre de transitions rencontrées en rejouant chaque transition avec tous les états possibles de la MAE. De plus, le HRM maintient une politique différente pour chacun des états mais comme cela a été soulignée, cela implique que la MAE doit être gardée pour la politique finale afin de choisir quelle politique doit être utilisée.

Plusieurs travaux se sont penchés sur le fait d'apprendre automatiquement une MAE ou un automate dans ce contexte [19] mais ces approches perdent alors l'intérêt d'utiliser une MAE créée par un humain qui est plus facilement interprétable.

C. Fonctionnement en phase d'apprentissage et modification de la phase exploratoire

Les méthodes de l'état de l'art se rapprochant le plus de ce concept sont issues du *Curriculum Learning* mais n'utilisent pas la structure de MAE, mais plutôt celle de graphe [14]. En effet, plusieurs approches mettent en place un ensemble de tâches et font évoluer la tâche courante à partir de laquelle les interactions sont générées en phase exploratoire en fonction de l'état d'avancement de la politique dans son processus d'apprentissage.

Placement			Modification	Article
Exploratoire	Parallèle	Ω	Exploratoire	[18] [19]
			Apprentissage	[18] [19]
		$\mathcal{S}$	Exploratoire	
			Apprentissage	
	Série	Ω	Exploratoire	[16] [17]
			Apprentissage	
Apprentissage			Exploratoire	[14]
			Apprentissage	

TABLE I: Tableau récapitulatif des méthodes présentées avec l'approche proposée

VII. Perspectives

Cette section propose des premières pistes de couplage entre une MAE et un algorithme de DRL appliquées à l'environnement *Lunar Lander*. Ces études ont comme but d'évaluer des gains en performances et en explicabilité afin d'avoir une approche permettant une application finale exploitable dans notre contexte industriel.

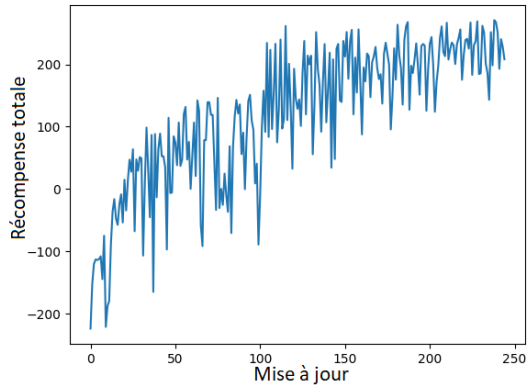
A. Nouvelles métriques d'évaluation de l'état courant de la politique

Plusieurs approches se basent sur des métriques de l'état courant de la politique afin de pouvoir mettre en place des rétroactions dans le processus d'apprentissage, par exemple, l'utilisation de l'entropie pour [10]. Ainsi, ces métriques permettent d'ajuster le mécanisme d'apprentissage en prenant en compte l'état courant de la politique afin d'ajuster la rétroaction.

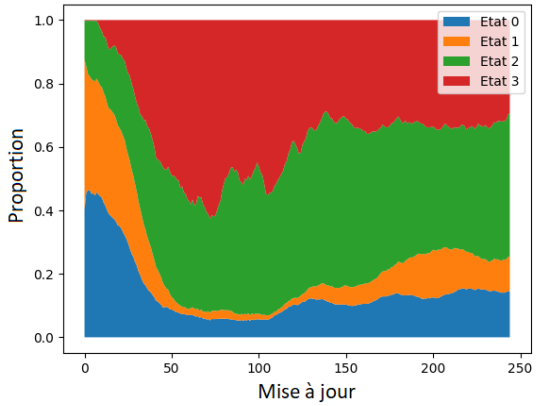
L'utilisation d'une MAE en phase exploratoire de façon parallèle peut permettre de classifier chacune des observations à un état issu de la fonction de transition. Cet ajout d'information peut permettre de créer de nouvelles métriques permettant d'évaluer l'état courant de la politique. Afin d'illustrer l'approche proposée, un entraînement utilisant PPO [10] sur l'environnement Lunar Lander a été fait, en utilisant la MAE de la section IV-B en parallèle lors de la phase exploratoire. Cela permet d'avoir, pour chacune des transitions de l'environnement, la valeur de l'état dans lequel la MAE est au cours de l'épisode sans modification algorithmique de l'apprentissage.

Un premier type de métrique est d'utiliser la proportion de chacun des états comme une métrique. La figure 7 montre la proportion moyenne de chacun des états rencontrés par la politique par épisode au cours d'un apprentissage. Ainsi, ce type de métrique peut permettre d'avoir des éléments de compréhension sur l'état courant de la politique au cours de l'apprentissage. En effet, on voit ici que le début de l'apprentissage se concentre sur la phase aérienne car la proportion de l'état 3 est très faible, puis le concept d'atterrissage

devient beaucoup plus présent avec une forte proportion d'état 3. Finalement, on observe une phase où les proportions des états convergent vers des valeurs stables avec notamment une récompense totale moyenne qui évolue beaucoup moins.



(a) Récompense moyenne par épisode



(b) Proportion moyenne par épisode de chaque état

Fig. 7: Apprentissage sur l'environnement Lunar Lander avec PPO en utilisant une MAE en parallèle à partir de $\mathcal{S}$ pour classifier les différents états

Un second type de métriques est d'utiliser des métriques déjà existantes, mais de les évaluer par état afin d'avoir une évaluation de l'état de la politique pour chacun des états. La figure 8 montre l'écart à la valeur moyenne de l'entropie de l'acteur par état cours de l'entraînement. On peut voir ici une différence notable entre les différents états et plus particulièrement de l'état 3 par rapport aux autres états donc l'évaluation de métriques par état semble intéressante pour déceler des différences au cours de l'apprentissage face à des transitions venant d'états différents.

De façon similaire à l'algorithme HRM issu de [18], il peut être intéressant de différencier le traitement algorithmique en fonction des états grâce à ces différentes métriques afin de profiter de la structure de la MAE pour diviser la complexité du problème global en sous-problèmes plus simple.

B. Utilisation d'une MAE Politique

Dans un cas où une MAE existe déjà pour répondre au problème de décision séquentiel visé, comme dans notre cas

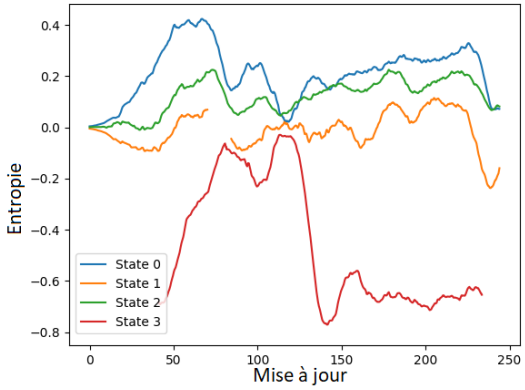


Fig. 8: Écart à la valeur moyenne de l'entropie de l'acteur par état

industriel, il peut être intéressant d'étudier la capacité de tirer profit de la structure de MAE tout en ayant accès à une politique pour chacun de ces états. Cependant, il est bon de noter, qu'on se place dans un contexte où la politique fournie par la MAE est sous-optimale car sinon, l'intérêt d'appliquer une méthode d'apprentissage par renforcement est inexistant.

On définit ici une machine à états politique (MAE-π) par une machine de Mealy définie par un tuple $\langle \mathcal{Q}, \mathcal{U}, \delta, q_0, \mathcal{A}, \pi \rangle$ avec $\langle \mathcal{Q}, \mathcal{U}, \delta, q_0 \rangle$ une MAE, $\mathcal{A}$ l'ensemble des actions possibles et $\pi : \mathcal{Q} \times \mathcal{U} \rightarrow \mathcal{A}$, une fonction de sortie, qui permet donc d'avoir une politique, dépendant de l'état courant de la MAE et de l'observable.

Dans la littérature, on trouve différentes méthodes pouvant utiliser des MAE politiques. En effet, il est possible d'utiliser les approches de *Transfer Learning* basées sur l'utilisation d'une politique extérieure (notamment le *Policy Transfer* ou le *Learning from demonstration*), bien qu'elles ne tirent pas profit de la structure de la MAE et qu'elles doivent être capables de tirer profit d'une politique imparfaite. Il peut donc être intéressant d'explorer l'utilisation de ces approches, mais en tirant profit de la structure de MAE, ce qui permettrait d'avoir un traitement différent et potentiellement plus adéquat pour chacun des états.

Les MAE politiques semblent intéressantes pour une utilisation en parallèle avec des modifications dans la phase exploratoire, car cela permet d'introduire une action alternative à celle fournie par la politique optimisée, ce qui peut permettre de guider l'exploration lors de l'apprentissage grâce à une heuristique métier si certains états sont complexes. De plus, il est également possible de restreindre l'utilisation de la politique issue du DRL à seulement certains états de façon similaire à [17].

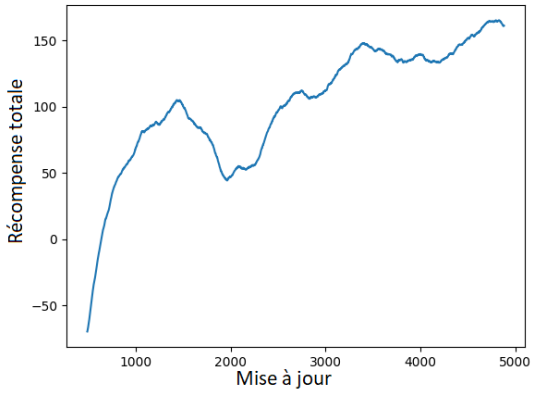
C. Utilisation de $\mathcal{S}$ comme espace d'entrée de la MAE

Une MAE utilisant directement Ω peut être complexe à mettre en place lorsque l'observation est difficile à traiter, notamment avec des données déstructurées (images, nuage de points lidars, sonars). Cependant, lorsque l'espace $\mathcal{S}$ existe et qu'il est possible grâce au simulateur d'y avoir accès, une

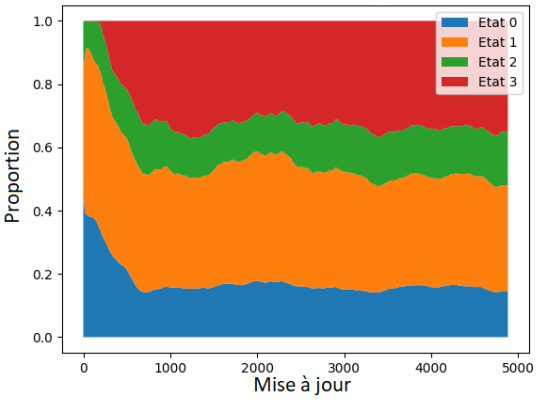
approche pourrait être d'utiliser directement $\mathcal{S}$ comme espace d'entrée de la MAE. En reprenant l'approche et les termes de [26] qui décompose le problème de la décision séquentielle en un problème de compréhension de l'observation et un problème de choix de l'action, dans notre cas, cette approche permet de nous affranchir du problème de compréhension de l'observation pour la création de la MAE. Ainsi, l'utilisation d'Ω semble intéressante dans les environnements où le problème de compréhension de l'observation est complexe.

Cependant, cela implique une utilisation de la MAE sans nécessité de l'utiliser pour l'inférence de la politique finale. En effet, les informations issues de $\mathcal{S}$ ne sont pas accessibles dans le cas d'une résolution réelle. Toutes les approches modifiant la phase d'apprentissage ou bien agissant seulement sur la politique exploratoire peuvent être envisagées, comme celles présentées dans les sections VII-A et VII-B. Cela permet de simplifier la création de la MAE et donc potentiellement d'augmenter son efficacité.

Afin d'illustrer ce concept, un apprentissage à partir de Ω tout en utilisant la MAE de la section IV-B en parallèle à partir de l'espace $\mathcal{S}$ a été fait à l'aide de PPO, ce qui implique aucun changement algorithmique dans le processus d'apprentissage. La figure 9 montre l'évolution de la récompense moyenne ainsi que de la proportion d'état par épisode. Tout d'abord, on note que la modification de l'observation complexifie l'apprentissage avec un nombre d'interactions nécessaires plus grand et une performance plus faible. L'ajout de la MAE permet notamment de voir que la proportion de l'état 1 (instable) est beaucoup plus grande que celle observée pour l'apprentissage à partir de $\mathcal{S}$ (figure 7) ce qui donne des informations sur les difficultés rencontrées au cours de l'apprentissage pour obtenir une politique capable de stabiliser l'angle de roulis du vaisseau à partir de Ω.



(a) Récompense totale moyenne par épisode



(b) Proportion moyenne par épisode de chaque état

Fig. 9: Apprentissage sur l'environnement Lunar Lander utilisant PPO à partir d'Ω en utilisant une MAE en parallèle à partir de $\mathcal{S}$ pour classifier les différents états

Conclusion

L'apprentissage par renforcement profond est une approche très intéressante pour la résolution de problème de décision séquentielle comme celui de la mise en place d'un modèle opérateur en simulation numérique. Cependant, cette approche présente des lacunes sur l'efficience de l'utilisation des données et l'explicabilité du modèle final. Dans un contexte industriel, et en particulier celui de la défense nationale, la nécessité d'obtenir un résultat explicable avec un coût calculatoire raisonnable ce qui est très critique dans la stratégie d'investissement matériel et logiciel.

Les MAE sont déjà fortement présentes pour résoudre ce type de problème, car elles montrent de bonnes performances tout en restant explicables. En outre, l'utilisation couplée d'une MAE avec le DRL est une approche efficace et déjà largement utilisée dans la littérature pour ajouter des connaissances extérieures afin d'aider le processus d'apprentissage. En effet, cela permet d'accélérer la convergence et/ou d'augmenter l'interprétabilité du modèle final soit par des restrictions de l'espace d'action ou bien par des processus d'apprentissage plus compréhensibles.

Nous avons donc défini un cadre théorique pour l'incorporation d'une MAE au sein d'un processus d'apprentissage par renforcement. Ainsi, ces définitions permettent d'identifier les contraintes et les gains potentiels en fonction du placement du couplage et nous les avons utilisées pour replacer plusieurs approches de l'état de l'art. En outre, cela a permis d'identifier des couplages algorithmiques intéressants, et non explorés jusqu'à présent. Tout d'abord, la mise en place de nouvelles métriques utilisant la structure d'une MAE afin d'avoir plus d'informations sur l'état courant de la politique au cours de l'apprentissage. En outre, l'utilisation d'une MAE politique semble également être une approche intéressante pour tirer profit de connaissances a priori sur le comportement souhaité. Finalement, l'utilisation d'information supplémentaire non accessible pour le modèle final peut permettre de simplifier et d'améliorer la mise en place de MAE et donc d'améliorer les gains d'explicabilité et de performance d'un couplage. Ces différentes approches proposées sont en cours de développement sur l'environnement Lunar Lander, dans un but de résolution de la problématique industrielle exposée initialement.

REFERENCES

[1] R. S. Sutton and A. G. Barto, “Reinforcement Learning : An Introduction, ”, Second Edition. Cambridge, Massachussetts : The MIT Press, 2018, isbn : 978-0-262-03924-6.

[2] V. Mnih, K. Kavukcuoglu, D. Silver et al. “Human-level control through deep reinforcement learning, ” Nature, 518 (7540):529–533, 2015

[3] D. Silver, J. Schrittwieser, K. Simonyan et al., “Mastering the game of Go without human knowledge, ” Nature, t. 550, no 7676, p. 354-359, 2017.

[4] D. Silver, S. Singh, D. Precup R. S. Sutton, ”Reward is enough, “ Artificial Intelligence, t. 299, p. 103 535, 2021, issn : 00043702.

[5] A. Gupta, A. Pacchiano, Y. Zhai, S. Kakade and S. Levine, “Unpacking Reward Shaping : Understanding the Benefits of Reward Engineering on Sample Complexity, ” in Advances in Neural Information Processing Systems, 2022, p. 15 281-15 295.

[6] T. Zahavy, N. Ben-Zrihem and S. Mannor, “ Graying the black box : Understanding DQNs, ” in Proceedings of The 33rd International Conference on Machine Learning, New York, New York, USA : PMLR, 2016, p. 1899-1908.

[7] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup and D. Meger, “ Deep Reinforcement Learning That Matters, ” Proceedings of the AAAI Conference on Artificial Intelligence, t. 32, no 1, 29 avr. 2018, issn : 2374-3468, 2159-5399.

[8] G. Dulac-Arnold, N. Levine, D. J. Mankowitz et al. “ An empirical investigation of the challenges of real-world reinforcement learning, ” arXiv : 2003.11881, preprint 2020.

[9] W. Fedus, P. Ramachandran, R. Agarwal et al., “ Revisiting Fundamentals of Experience Replay, ” arXiv : 2007.06700, preprint 2020.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov. « Proximal Policy Optimization Algorithms. » arXiv : 1707.06347.

[11] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine. “ Soft Actor-Critic : Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, ” arXiv : 1801.01290.

[12] A. Aubret, L. Matignon and S. Hassas. “ A survey on intrinsic motivation in reinforcement learning, ” arXiv : 1908.06976.

[13] . Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel and J. Peters, “ An Algorithmic Perspective on Imitation Learning, , ” Foundations and Trends in Robotics, t. 7,no 1-2, p. 1-179, 2018, issn : 1935-8253, 1935-8261.

[14] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. Taylor and P. Stone, “ Curriculum Learning for Reinforcement Learning Domains : A Framework and Survey, ” Journal of Machine Learning Research, 2021.

[15] Z. Zhu, K. Lin, A. K. Jain and J. Zhou. “ Transfer Learning in Deep Reinforcement Learning :A Survey, ” arXiv : 2009.07888.

[16] G.-C. Kang and Y. Lee, « Finite State Machine-Based Motion-Free Learning of Biped Walking, » IEEE Access, t. 9, p. 20 662-20 672, 2021, issn : 2169-3536.

[17] S. Hwang, K. Lee, H. Jeon and D. Kum, “ Autonomous Vehicle Cut-In Algorithm for Lane-Merging Scenarios via Policy-Based Reinforcement Learning Nested Within Finite-State Machine, ” IEEE Transactions on Intelligent Transportation Systems, t. 23, no 10, p. 17 594-17 606, oct. 2022, issn : 1524-9050, 1558-0016.

[18] R. T. Icarte, T. Q. Klassen, R. Valenzano and S. A. McIlraith, “ Using Reward Machines for High-Level Task Specificationand Decomposition in Reinforcement Learning, ” Proceedings of the 35 th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[19] D. Furelos-Blanco, M. Law, A. Russo, K. Broda and A. Jonsson, “ Induction of Subgoal Automata for Reinforcement Learning, ” Proceedings of the AAAI Conference on Artificial Intelligence, t. 34, no 04, p. 3890-3897, 2020, issn : 2374-3468, 2159-5399

[20] R. T. Icarte, R. Valenzano, E. Waldie, M. P. Castro, T. Q. Klassen and S. A. McIlraith, “ Learning Reward Machines for Partially Observable Reinforcement Learning, ” in Advances in Neural Information Processing Systems, 2019.

[21] C. Glanois, P. Weng, M. Zimmer et al. “ A Survey on Interpretable Reinforcement Learning, ” arXiv : 2112.13112.

[22] J. E. Hopcroft, R. Motwani and J. D. Ullman, “ Introduction to Automata Theory, Languages, and Computation, ” 3rd ed. Boston : Pearson/Addison Wesley, 2007, 535 p., isbn : 978-0-321-45536-9 978-0-321-46225-1 978-0-321-45537-6.

[23] G. Brockman, V. Cheung, L. Pettersson et al. « OpenAI Gym. » arXiv : 1606.01540.

[24] https://gymnasium.farama.org/environments/box2d/lunar_lander/#lunar-lander

[25] B. Baker, I. Kanitscheider, T. Markov et al. “ Emergent Tool Use From Multi-Agent Autocurricula, ” arXiv : 1909.07528.

[26] D. Chen, B. Zhou, V. Koltun and P. Krähenbühl, “ Learning by Cheating, ” Proceedings of Machine Learning Research, t. 100, p. 66-75, 2020.