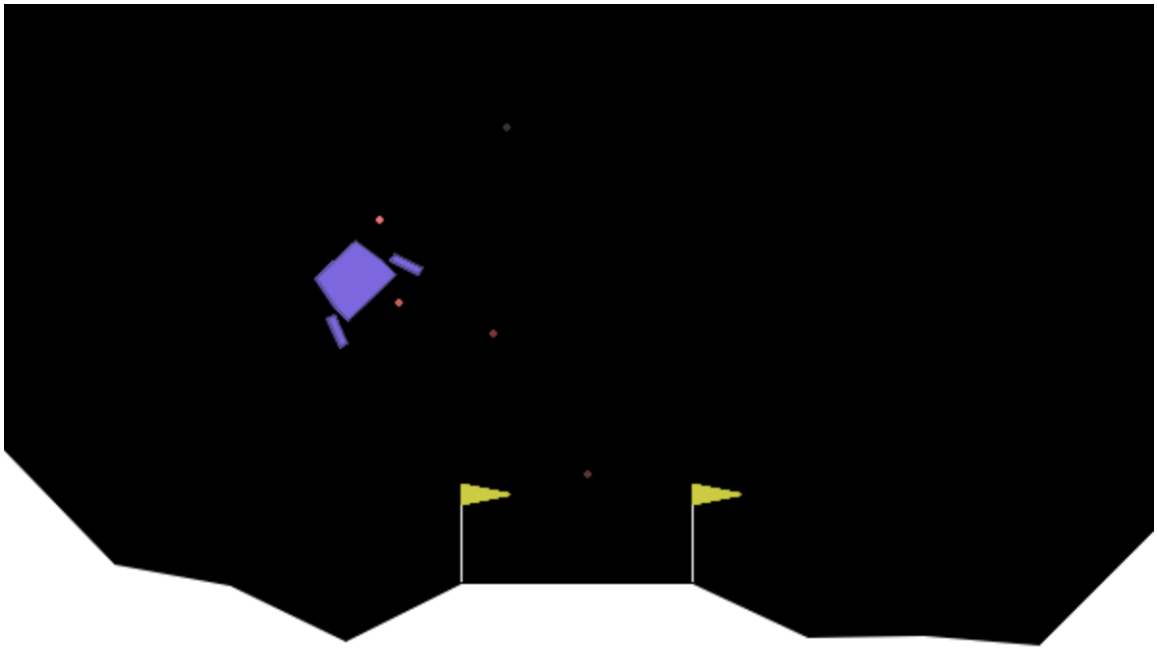
Projet d'Apprentissage par Renforcement

Analyse des algorithmes de classiques de DeepRL - DQN, PPO et A2C - sur la tâche d'alunissage de Gymnasium.



Oscar PERIANAYAGASSAMY

Enseignant : Stéphane AIRIAU

Dauphine | PSL
UNIVERSITÉ PARIS

Formation : Master 2 - Intelligence Artificielle, Systèmes, Données - Parcours apprentissage
Établissement : Université Paris-Dauphine PSL
Année universitaire : 2025–2026

Table des matières

1 Introduction

1.1 Présentation de la tâche

Le problème qui a été choisi est la tâche d'alunissage d'un aéronef issu de la bibliothèque Python `Gymnasium`, plus communément appelée Lunar Lander. Elle consiste à faire alunir un module entre deux drapeaux sans accident lors de la descente vers la surface. Dans sa version classique, il s'agit d'un problème épisodique et stationnaire.

L'espace des actions a une taille finie de 4 possibilités qui correspondent à actionner soit le propulseur central, soit l'un des propulseurs latéraux ou ne rien faire du tout.

Concernant l'espace des états, nous avons affaire à un espace continu dont les représentants sont des tenseurs à 8 dimensions. Parmi leurs composantes, nous retrouvons les coordonnées du module, ses vélocités linéaires dans les deux directions du plan, son angle d'inclinaison, sa vélocité angulaire et deux variables indicatrices du contact au sol des pieds du module.

Enfin, le modèle de récompenses n'est pas entièrement connu. Il nous est simplement indiqué que la récompense immédiate varie selon la distance à la zone d'alunissage et la vitesse du module. Cette récompense est réduite si le module a un angle trop élevé. Néanmoins, nous apprenons sur la documentation `Gymnasium` qu'avoir l'un des pieds au sol rapporte $+10$, l'actionnement d'un moteur latéral coûte -0.03 là où celui du moteur central est à -0.3, l'alunissage correct ajoute $+100$ et un crash coûte -100.

Un épisode de longueur T est considéré comme réussi lorsque la récompense totale sur l'épisode est supérieure à 200. Autrement dit, il est validé si l'inégalité suivante est vérifiée :

$$R := \sum_{k=0}^{T} r_k \geq 200$$

pour r_k la récompense immédiate au pas de temps k.

Cette présentation nous montre d'emblée que les modèles tabulaires seront inefficaces pour la résolution de ce problème. En effet, l'espace des états est un espace continu. Sa discrétisation est envisable mais ne permettra pas d'atteindre des résultats satisfaisants en un nombre d'épisodes raisonnable. Par exemple, si nous décidions d'utiliser un pas de discrétisation de 0.01, nous aurions à considérer au moins : 2 500 000 000 000 000 états. La table $\mathcal{Q}$ à charger en mémoire RAM aurait donc une taille de 10 000 000 000 000 000, ce qui n'est évidemment pas quelque chose que nous souhaitons faire dans un contexte local pour le bien de nos machines.

Il convient donc d'utiliser des méthodes de deep reinforcement learning telles que Deep Q-Learning (`DQN`), Proximal Policy Optimization (`PPO`) et Advantage Actor-Critic (`A2C`) qui abordent chacune le problème d'une manière différente.

En plus de la configuration stationnaire avec environnement d'actions discret, `Gymnasium` nous permet également de rendre le problème continu en gérant l'actionnement d'un levier de poussée pour chacun des propulseurs. Dans ce contexte, `DQN` ne sera plus apte pour la résolution de par le calcul d'un maximum selon les actions. De plus, il est aussi possible d'ajouter des turbulences qui viennent induire de la stochasticité dans l'apprentissage.

1.2 Problématique et protocole d'expérimentation

Le but est d'étudier, pour la tâche précise de l'alunissage, l'influence du taux d'apprentissage pour `DQN`, `PPO` et `A2C`. Nous limitons cette étude aux résultats d'entrainement uniquement, et plus précisément à la rémunération moyenne obtenue comme définie par R plus haut.

Comprenons dans un premier temps ce que la notion de "step" représente pour la librairie `Stable-Baselines3`. D'après la documentation, il s'agit du nombre total d'échantillons utilisés pendant l'entrainement. Il est important de noter que chaque algorithme a une manière différente de mettre à jour ses poids. Par exemple, `DQN`, qui est un algorithme off-policy, la réalise à chaque pas de temps. Au contraire, `PPO` et `A2C` l'effectuent après avoir collecté un certain nombre d'échantillons.

Le choix s'est porté sur le test de 5 taux d'apprentissage différents distribués de manière logarithmique (base 10) entre 10^{-7} et 10^{-2}. Cela nous permet notamment d'étudier à quel point l'"agressivité" du taux d'apprentissage joue un rôle majeur dans la construction de la fonction de valeur ou de la politique optimale. Ensuite, nous avons choisi 200 000 pas d'apprentissage. C'est une valeur qui nous permet d'avoir un temps d'apprentissage inférieur à une heure pour chaque modèle tout en ayant simulé assez d'époques pour avoir des résultats interprétables. Chacun des modèles est entrainé sur la même graine aléatoire et l'est une seule fois par taux d'apprentissage. Ce n'est pas optimal comme méthode d'évaluation car il ne nous permet pas de voir le comportement en moyenne des algorithmes entrainés. Néanmoins, nous les évaluons chacun sur un petit nombre d'époques pour analyser leurs résultats après entrainement et générons des vidéos pour `PPO` par exemple dans le dossier du

projet. Enfin, en plus du lunar lander classique, nous entraînons, quand c'est possible, chacun des modèles sur les versions suivantes du problème :

— Version classique à actions discrètes *[CLASSIC VANILLA]*
— Version classique à actions continues *[CLASSIC CONTINUOUS]*
— Version venteuse à actions discrètes *[WINDY VAILLA]*
— Version venteuse à actions continues *[WINDY CONTINUOUS]*
— Version zero-gravité venteuse à actions discrètes *[ZERO-GRAVITY-WINDY VANILLA]*
— Version zero-gravité venteuse à actions continues *[ZERO-GRAVITY-WINDY CONTINUOUS]*

D'un point de vue matériel, l'ensemble des entraînements a été réalisé en local sur un ordinateur portable doté d'une puce Apple M4 avec un moteur neuronal 16 coeurs et 32 Go de RAM. Au total, plus de 20 heures d'entrainement ont été réalisées.

Au niveau de la programmation, les trois algorithmes n'ont pas été réimplémentés pour l'occasion. À la place, il a été décidé d'utiliser la librairie `Stable-Baselines3` qui est très bien raccordée aux environnements `Gymnasium` et qui a un grand pouvoir d'abstraction de par sa philosophie "haut-niveau". De plus, ce module est pensé pour fonctionner avec `TensorBoard` qui est un outil de visualisation interactif associé à `Tensorflow`. Nous tirons donc parti de ces fonctionnalités pour nous concentrer sur les aspects algorithmiques relatifs à l'apprentissage.

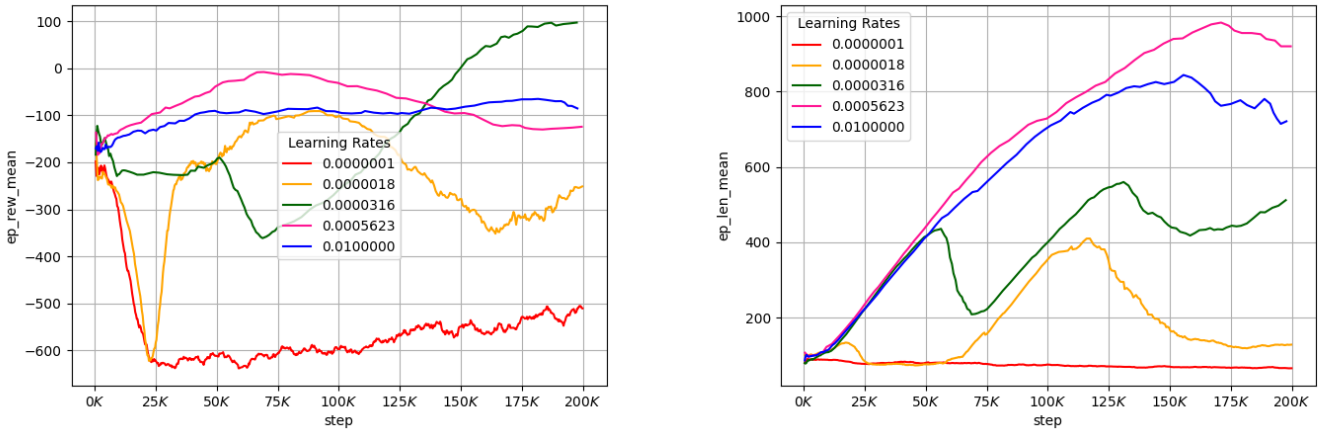
2 Analyse des résultats expérimentaux

Dans cette section, nous allons évaluer chacun des algorithmes sur la tâche d'alunissage avec les différentes variantes présentées en introduction. Le but est de confirmer ou d'infirmer des intuitions que nous pourrions avoir sur ces méthodes. Ainsi, nous présentons les différentes variantes avec une analyse de l'évolution de différentes grandeurs relatives à l'apprentissage selon le taux d'apprentissage choisi. Pour le problème `CLASSIC`, nous essayons d'être aussi exhaustif que possible. Pour les deux suivants, `WINDY` et `ZERO-GRAVITY-WINDY`, nous nous concentrons sur certains résultats intéressants sans entrer dans un descriptif complet. Tous les résultats non-décrits sont disponibles en annexe de ce document.

2.1 Problème CLASSIC

Cette variante est la variante classique où le module lunaire subit la gravité mais ne rencontre aucune turbulence.

2.1.1 Variante `VANILLA`



(a) DQN - Récompense moyenne

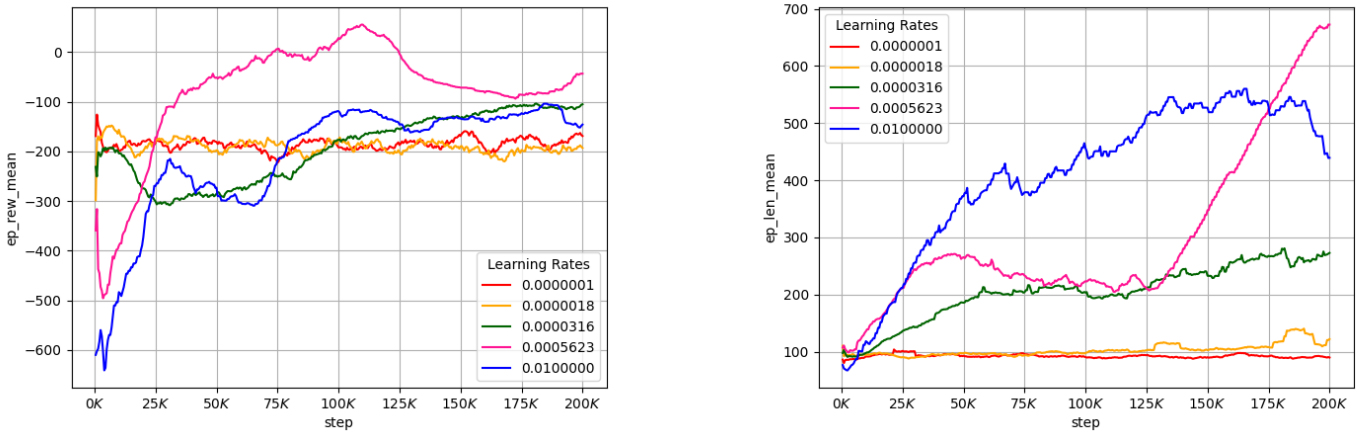
(b) DQN - Longueur moyenne d'épisode

FIGURE 1 – Résultats comparatifs sur la variante `CLASSIC VANILLA` du Lunar Lander pour `DQN`

Tout d'abord, `DQN` présente des résultats différenciés selon le learning rate. Comme présenté sur la FIGURE 1A, le plus petit taux d'apprentissage ($1.0 \cdot 10^{-7}$) conduit à des résultats catastrophiques avec un effondrement de la récompense moyenne en début d'entrainement, là où le second ($1.8 \cdot 10^{-6}$) connaît le même effondrement suivi d'un retour à la valeur initiale puis une oscillation entre -350 et -100. Les deux plus grands taux d'apprentissage ($5.623 \cdot 10^{-4}$ et $1.0 \cdot 10^{-2}$) ont des apprentissages très similaires avec des performances assez moyennes mais bien meilleures que les deux plus petits. La meilleure performance est détenue par $3.16 \cdot 10^{5}$ avec une valeur de récompense moyenne en fin d'apprentissage autour de 100, qui est un résultat plus que correct. En parallèle, le graphique FIGURE 6B semble nous indiquer qu'à long terme, la durée moyenne de l'épisode

nous donne certaines garanties quant à la récompense. Les épisodes longs semblent correspondre à des moments où l'apprenti prend son temps pour l'alunissage et donne donc une certaine sécurité sur la valeur de la récompense. A contrario, les épisodes courts ont des comportements avec une moins bonne rémunération. Enfin, la courbe verte, associée au taux d'apprentissage $3.16 \cdot 10^5$, propose un bon compromis : une durée médiane et une excellente rémunération en moyenne.

Nous voyons donc que `DQN` a des résultats disparates et il est donc assez délicat de donner un avis définitif sur la capacité de l'algorithme à apprendre cette tâche efficacement. Les résultats associés au taux d'apprentissage $3.16 \cdot 10^5$ pourraient seulement être la conséquence d'une trajectoire probable mais rare.



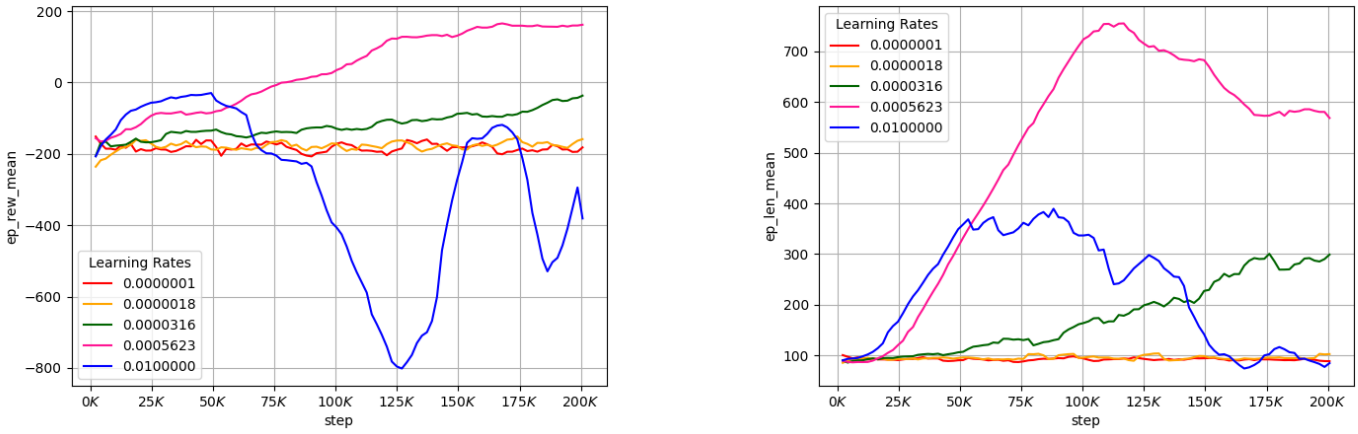
(a) A2C - Récompense moyenne

(b) A2C - Longueur moyenne d'épisode

FIGURE 2 – Résultats comparatifs sur la variante `CLASSIC VANILLA` du Lunar Lander pour `A2C`

Concernant `A2C`, la FIGURE 2A affiche des apprentissages très bruités. Cela s'explique par la méthode de mis-à-jour de l'algorithme : il récolte un certain nombre d'observations puis s'entraîne dessus avant de les oublier. `PPO` utilise la même technique mais avec une nombre de pas bien plus petit. Au contraire, `DQN` s'entraîne à chaque nouveau pas. Ce que nous montre également le graphique est que, pour chaque taux d'apprentissage, la méthode semble converger autour d'une récompense moyenne entre -200 et -50 environ. Ces résultats ne sont pas excellents mais néanmoins cela nous donne certaines garanties en matière de vitesse de convergence. En parallèle, la FIGURE 5B montre une décorrélation entre la récompense et la longueur d'un épisode en moyenne.

`A2C` se démarque par une certaine stabilité du point de vue de la convergence. Néanmoins, l'algorithme subit malgré tout d'oscillations assez légères.



(a) PPO - Récompense moyenne

(b) PPO - Longueur moyenne d'épisode

FIGURE 3 – Résultats comparatifs sur la variante `CLASSIC VANILLA` du Lunar Lander pour `PPO`

Enfin, pour `PPO`, les récompenses moyennes affichées sur la FIGURE 3A sont similaires à `A2C` pour les learning rates $1.0 \cdot 10^{-7}$ et $1.8 \cdot 10^{-6}$, de même que la longueur moyenne des épisodes. Il est fort probable que ces traces témoignent du fait que ces taux sont faibles et ne permettent pas à l'algorithme des pas de gradient assez élevés pour sortir d'un plateau sous-optimal.

Nous voyons également que l'augmentation trop forte de ce taux d'apprentissage, ce qui nous conduit à la courbe bleue, implique une récompense suivant une trajectoire très instable et propice à l'effondrement.

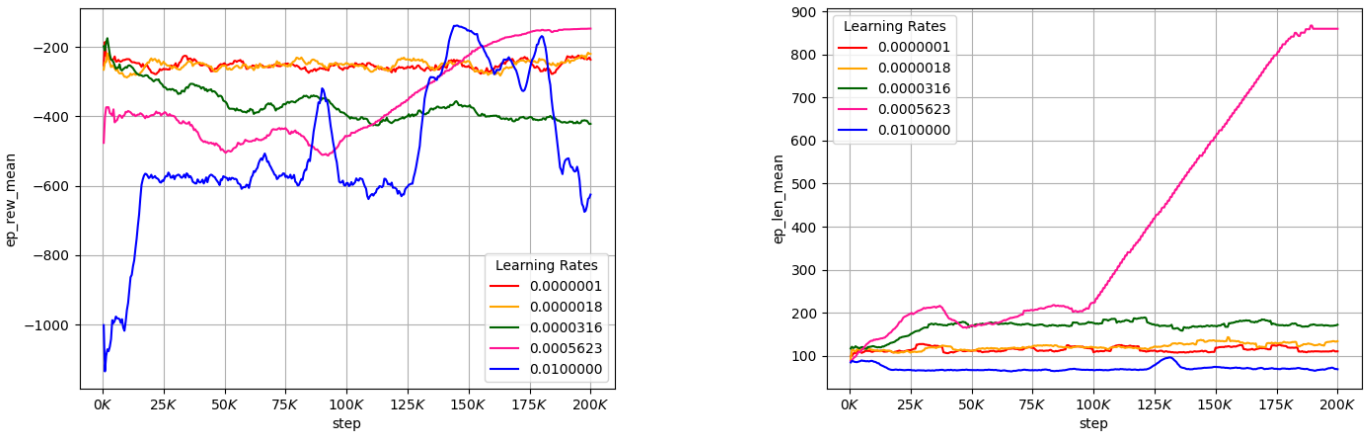
Néanmoins, une trace d'apprentissage se démarque de par sa convergence autour de 170 de récompense moyenne. Elle est atteinte grâce à un taux d'apprentissage modérément élevé ($5.623 \cdot 10^{-4}$). Malgré cela, la durée moyenne des épisodes associés est assez élevé comme affichée sur la FIGURE 5B mais a connu une belle décroissance à partir du 100 000ème pas de temps. C'est un résultat remarquable car avoir des bonnes récompenses est une excellente chose. Néanmoins, si la longueur des épisodes est très longue, ce n'est pas forcément une solution pertinente. Par exemple, nous pourrions considérer une situation où l'objectif de la mission est quelque peu différent : le but n'est plus seulement d'alunir mais de réaliser un alunissage d'urgence pour la survie de l'équipage. Cette situation requiert donc une immense précision dans la manoeuvre mais surtout une certaine rapidité d'exécution de l'action. C'est pourquoi cette trace d'apprentissage garde en quelque sorte le meilleur des deux mondes et pourrait encore réduire sa longueur moyenne d'épisode à encore plus long terme.

Pour des learning rate pas trop agressifs, l'algorithme `PPO` est très stable en matière de récompense moyenne mais reste assez peu intéressante par rapport à la situation que nous venons de décrire à l'instant.

Pour conclure cette section, les résultats expérimentaux que nous venons de présenter confirment nos intuitions sur ces algorithmes. `DQN` peut converger mais nous n'avons aucune garantie de manière générale. Nous voyons également que sa méthode d'apprentissage peut parfois le bloquer dans des trajectoires sous-optimales voire catastrophiques. En effet, il prend en compte toute l'expérience accumulée jusqu'à présent et est, par conséquent, assez sensible au bruit. `A2C`, quant-à-lui, promet une convergence rapide mais assez risquée en début d'apprentissage avec une oscillation de la récompense moyenne non-négligeable. Enfin, `PPO` converge lentement de manière robuste à condition de ne pas sélectionner un taux d'apprentissage trop extrême.

2.1.2 Variante `CONTINUOUS`

La variante continue de ce problème consiste à modifier l'espace des actions en passant d'un ensemble fini à 4 éléments à une partie fermée de $\mathbb{R}^2$, plus précisément l'ensemble $[\![-1,+1]\!]^2$ où la première coordonnée correspond à l'actionnement du propulseur principal, activé que si la valeur est supérieure à 0. La seconde est associée aux propulseurs latéraux. Le gauche est actionné si la valeur est inférieure à -0.5. Le droit l'est si la valeur est supérieure à 0.5.



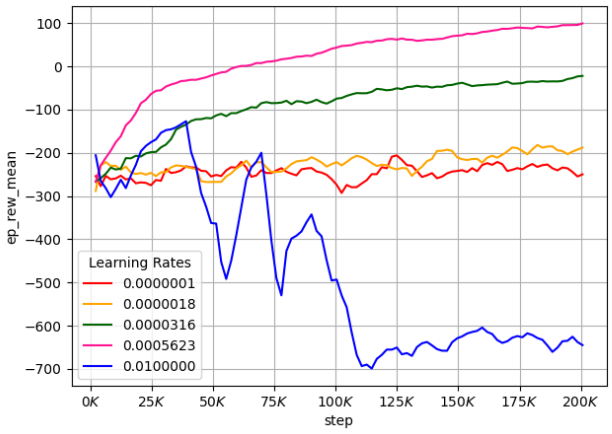
(a) A2C - Récompense moyenne

(b) A2C - Longueur moyenne d'épisode

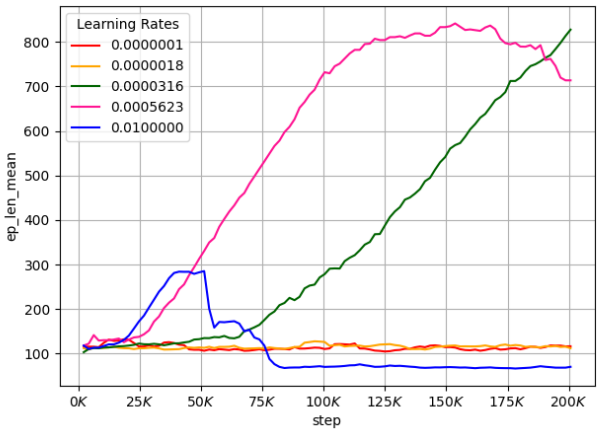
FIGURE 4 – Résultats comparatifs sur la variante `CLASSIC CONTINUOUS` du Lunar Lander pour `A2C`

Tout d'abord, `A2C` ne brille pas particulièrement sur les traces d'apprentissage collectées. En effet, la FIGURE 4A affiche pour trois taux d'apprentissage distincts une évolution de la valeur moyenne de récompense assez médiocre, constante pour deux d'entre eux ($1.0 \cdot 10^{-7}$ et $1.8 \cdot 10^{-6}$) et assez légèrement décroissante pour le troisième ($3.16 \cdot 10^{-5}$). En parallèle, la longueur moyenne d'épisode reste quasiment constante également. Il est probable que l'algorithme soit bloqué dans des plateaux sous-optimaux ce qui empêche l'apprentissage d'avoir lieu. Au contraire, le taux d'apprentissage le plus agressif, i.e. $1.0 \cdot 10^{-2}$ présente un tableau bien plus instable avec des montées conséquentes mais aussi des effondrements tout aussi conséquents avec une stabilisation de la longueur moyenne d'épisodes. Enfin, le taux d'apprentissage qui donne le résultat le plus satisfaisant ici est $5.623 \cdot 10^{-4}$. En effet, les courbes de récompense et de longueur de l'épisode moyennes au cours de l'apprentissage suivent la même trajectoire : une phase plutôt oscillatoire puis une croissance stable.

Ces résultats nous rappellent que `A2C` possède cette notion de convergence rapide. Mais, le prix à payer est une certaine variabilité et une instabilité entre les apprentissages.



(a) PPO - Récompense moyenne



(b) PPO - Longueur moyenne d'épisode

FIGURE 5 – Résultats comparatifs sur la variante `CLASSIC CONTINUOUS` du Lunar Lander pour `PPO`

Pour ce problème, `PPO` nous donne des résultats remarquables. En effet, nous saisissons parfaitement la robustesse de l'algorithme. Nous voyons notamment sur la FIGURE 5A que pour le taux d'apprentissage le plus agressif ($1.0 \cdot 10^{-2}$), la récompense moyenne s'effondre complètement à travers les pas de temps, avec des remontées significatives, mais pas suffisantes, jusqu'à une stabilisation dans un quasi-plateau catastrophique. La forme de cette courbe est corrélée à celle de la FIGURE 5B où la longueur moyenne d'épisode grandissait conjointement avec la récompense jusqu'au 50 000ème pas de temps. Néanmoins, la poursuite d'une stratégie qui a conduit l'algorithme sur des épisodes très courts n'a pas amélioré la situation pour l'entrainement.

Les deux taux d'apprentissage les plus calmes ($1.0 \cdot 10^{-7}$ et $1.8 \cdot 10^{-6}$) proposent une constance relative à la fois dans la récompense moyenne et dans la longueur moyenne d'un épisode. Nous pouvons tout de même remarquer que le second taux affiche une très légère croissance dans la rémunération qui confirme une convergence très lente de l'algorithme dans ce cas.

Enfin, les learning rate avec les meilleures résultats sont ceux des courbes rose ($5.623. \cdot 10^{-4}$) et verte ($3.16 \cdot 10^{-5}$). La seconde nous informe d'une croissance assez stable sur la phase d'apprentissage avec une augmentation de la longueur des épisodes en moyenne. La courbe rose, quant à elle, affiche les meilleurs résultats sur cette tâche. L'algorithme connaît en effet une croissance conjointe des courbes de récompense et de longueur d'épisode jusqu'au 150 000ème pas de temps. Vient ensuite une phase où la récompense moyenne continue de croître à la même vitesse alors que la longueur moyenne d'épisode décroît assez rapidement.

Pour conclure cette section sur le problème `VANILLA CONTINUOUS`, nous pouvons affirmer que `A2C` a des résultats assez médiocres ici, excepté pour une trace d'apprentissage qui semble prometteuse. Ces performances diffèrent grandement par rapport à celles que nous avions dans le cas discret, d'où l'instabilité à générer des conclusions générales pour cet algorithme. Par opposition, `PPO` affiche un comportement très similaire à celui lors de la première phase d'entrainement. A nouveau, nous trouvons une trace d'exécution excellente avec la croissance de la récompense moyenne et la chute de la longueur moyenne d'épisode dans le dernier quart de l'apprentissage.

2.1.3 Conclusion

Ainsi, le problème `VANILLA` a pu nous confirmer trois éléments.

Tout d'abord, l'algorithme `Deep Q-Learning` a un apprentissage assez instable et bruité. Il est assez difficile d'établir une heuristique robuste pour la sélection d'un taux d'apprentissage garantissant des résultats satisfaisants.

Concernant la méthode `Advantage Actor-Critic`, les performances rencontrées sont en demi-teinte. Le cas discret montre une certaine stabilité de la zone vers laquelle l'algorithme convergeait, une zone a priori sous-optimale mais non catastrophique. Au contraire, le cas continu présente plusieurs cas de convergence assez similaires mais avec deux traces d'apprentissage très différentes, l'une prometteuse et l'autre catastrophique.

Enfin, l'algorithme `Proximal Policy Optimization` est robuste sur les deux tâches, avec des garanties de convergence intéressantes mais aussi une distinction claire sur le choix du taux d'apprentissage.

2.2 Problème `WINDY`

Attaquons désormais de manière succincte le problème "venteux" où un effet de vent est appliqué au module lunaire. Deux forces affectent en conséquence l'aéronef : une qui est linéaire et l'autre rotationnelle.

2.2.1 Variante VANILLA

Les faibles taux d'apprentissage sont fortement affectés par le vent dans l'apprentissage de DQN comme le montre la Figure 6a. Auparavant, ces taux garantissaient une certaine constance dans la récompense obtenue, ce qui devient faux dans ce framework. Au contraire, les courbes rouge, bleue et rose démontrent une certaine robustesse face au vent malgré un apprentissage très lent, voire inefficace pour augmenter durablement la rémunération moyenne de l'agent sur le long terme.

Dans le cas CLASSIC VANILLA, l'algorithme A2C présentatit une stabilité intéressante pour les différents taux d'apprentissage. Dans notre cas, nous voyons à travers la Figure 6c des courbes très oscillatoires et sujettes à des grandes variations.

Enfin, PPO gère plutôt bien cet ajout de stochasticité dans l'apprentissage (Figure 6e). En effet, la courbe rose démontre à nouveau des résultats remarquables avec une croissance sur toute la durée d'apprentissage concernant la récompense moyenne de l'apprenti jusqu'à atteintre environ 60 comme rémunération. Concernant la longueur moyenne de l'épisode, encore une fois, elle grimpe puis amorce une descente autour du 175 000ème pas de temps.

2.2.2 Variante CONTINUOUS

A2C a une performance relativement étrange sur la version WINDY CONTINUOUS. En effet, la Figure 7a affiche des performances avec beaucoup d'oscillations, certes, et très instables pour la courbe bleue, mais on voit que cette dernière et la rose viennent croître et se rejoindre en fin d'apprentissage. Il est probable que cette trajectoire n'est pas durable pour la bleue. Néanmoins, c'est intéressant de voir que, pour ces taux d'apprentissage assez élevés, l'apprenant réussit à avoir des performances similaires au cas classique.

Concernant PPO (Figure 7c), nous constatons la même chose que dans le cas classique à la différence près que l'effondrement est deux fois plus dramatique ici pour le learning rate maximal $1.0 \cdot 10^{-2}$.

2.2.3 Conclusion

Pour conclure, dans le cas discret, l'ajout de vent vient perturber la stabilité que nous avions pue observer pour A2C de par les grandes disparités entre les courbes. DQN est aussi négativement impacté dans le sens où les effets d'effondrement sont bien plus violents pour la récompense et où la valeur moyenne est plus faible au global. PPO, lui, a une robustesse indéniable sur ce problème où le vent n'affecte pas plus que ça l'apprentissage tel que nous l'avons connu dans le cas CLASSIC VANILLA. Seules les valeurs moyennes de récompense sont légèrement plus faibles globalement.

Pour conclure, nous pouvons dire que la version "venteuse continue" ne démontre pas d'immenses différences par rapport au cas classique pour l'algorithme PPO. En effet, les changements vécus peuvent être vus comme des translations grossières des courbes qui conduisent à une diminution de la récompense moyenne à échelle globale. Pour A2C, ce même constat s'applique excepté pour la trajectoire de la courbe bleue qui semble avoir compensé l'effet du vent.

2.3 Problème ZERO-GRAVITY-WINDY

Terminons cette analyse par une version du problème où la constante gravitationnelle passée en entrée de l'environnement passe de -10 à -0.01. Concrètement, ce que ça signifie est que le module ne sera pas attiré vers la surface de la lune et sera uniquement mis en mouvement par les turbulences. C'est un moyen pour essayer de voir si l'apprenti a des chances de comprendre le monde qui l'entoure sans être attiré vers le sol. Nous comptons donc sur sa phase d'exploration pour comprendre la topologie du problème.

2.3.1 Variante VANILLA

Comme à l'accoutumée, commençons par le scénario avec l'ensemble d'actions discret.

La Figure 8a présente des récompenses très mauvaises pour DQN. Les courbes rouge et jaune sont en décroissance en fin d'apprentissage avec des récompenses moyennes en-dessous de -1000, ce qui est dramatique. Concernant les trois autres courbes, elles semblent être en phase de croissance très lente autour de -500. Ce test aurait vraisemblablement nécessité un bien plus grand nombre d'époques pour avoir un aperçu plus précis du comportement asymptotique de l'apprentissage.

A2C admet un taux d'apprentissage pour lequel l'effrondrement de performances est immense avec un minimum autour de -10 000 de récompense moyenne. Les quatre autres semblent avoir le même comportement que dans le cas CLASSIC VANILLA translaté d'environ 500 à 1000 points de récompense.

Les propriétés de stabilité que PPO pouvait afficher auparavant ont quelque peu disparu ici. En effet, la FIGURE 8E montre des courbes assez peu régulières avec de grandes oscillations. Dans ce cas encore, le learning rate $5.623 \cdot 10^{-4}$ admet un bon résultat par rapport aux autres.

2.3.2 Variante CONTINUOUS

La FIGURE 9A affiche des comportements chaotiques pour A2C. En effet, au-delà des deux taux d'apprentissage qui rendent la récompense moyenne stagnante autour de -1000, les trois autres taux connaissent des effondrements et des remontées successivement. Cela montre bien que à quel point l'algorithme est variable et est sujet aux variations.

PPO, comme les fois précédentes, conserve des allures de courbes très similaires. Néanmoins, il connaît un effondrement bien plus violent avec le taux d'apprentissage ($1.0 \cdot 10^{-2}$).

2.3.3 Conclusion

Cette section démontre principalement que l'apprenant nécessite bien plus de temps pour apprendre dans le cas où il n'y a pas de gravité qui l'attire vers le sol et lui indique donc une certaine direction. Les résultats des précédentes sections se ressentent encore légèrement mais dans une moindre mesure ici. C'est effectivement vrai pour PPO qui semble être très robuste peu importe la situation.

Pour le cas continu, comme pour le cas discret, un plus grand nombre de pas de temps aurait permis une analyse plus fine de ce scénario très exigeant pour l'apprenant. Malgré tout, nous retrouvons des comportements déjà observés pour PPO notamment.

3 Conclusion

Pour conclure cette analyse, nous pouvons tout d'abord dire que l'application dans un contexte d'apprentissage exigeant des algorithmes a permis une meilleure compréhension de leur comportement réel. Nous avons notamment pu constater l'instabilité de l'apprentissage de l'algorithme Deep Q-Learning, les qualités d'Advantage Actor-Critic en termes de rapidité de convergence vers un point et la robustesse de PPO dans une grande variété de contexte. La méthode d'évaluation est malgré tout critiquable dans le sens où, avec des capacités de calcul plus grandes, il aurait été possible d'affiner encore cette évaluation. Malgré tout, il est clair après cette étude que le taux d'apprentissage joue un rôle crucial en DeepRL. Nous avons pu le voir notamment dans les derniers cas, par exemple avec la FIGURE 9C, où des écarts entre la récompense moyenne de deux modèles sont à une hauteur d'environ 8 000 points.

4 Ouverture

Plusieurs pistes d'exploration connexes seraient intéressantes à mener avec ce problème :

- Utiliser des taux d'apprentissage non-constants : Comme nous l'avons vu durant l'analyse, certains taux d'apprentissage présentent des vertus qu'en début d'apprentissage avant que la récompense ne s'effondre à un instant donné, et vice versa. L'idée serait d'utiliser des techniques de schedulling sur le taux d'apprentissage pour rendre l'apprentissage évolutif (StepLR, ExponentialLR, CosineAnnealingLR, etc). Ces méthodes montrent déjà des résultats plus que probants en deep learning afin d'accélérer la convergence.
- Modifier le temps à partir duquel l'apprentissage commence vraiment : Le module Stable-Baselines3 permet de choisir un nombre de pas d'apprentissage pendant lesquels il ne fait qu'explorer et n'exploite pas sa connaissance pour choisir l'action suivante. Cette grandeur est fixée par défaut à 100, ce qui n'a pas été modifié ici. Ce paramètre pourrait avoir des vertus intéressantes à quantifier.
- Réimplémenter les algorithmes en adaptant les structures neuronales sous-jacentes : Stable-Baselines3 propose deux types d'architectures neuronales : un MLP (Perceptron Multi-Couches) ou un CNN (Réseau de Neurones Convolutionnel). La seconde option est préférée dans le cas d'observations sous la forme d'images. Il pourrait être utile d'explorer l'architecture avec originalité pour potentiellement améliorer les performances.
- Explorer des méthodes d'ensemble et de fusion : Nous savons que, en apprentissage supervisé, les méthodes d'ensemble permettent une réduction de la variance pour un même biais en général. En apprentissage par renforcement, il serait passionnant d'étudier à quel point l'agrégation de politiques, ou encore la fusion de réseaux de neurones, permet d'apprendre cette tâche.

Annexe

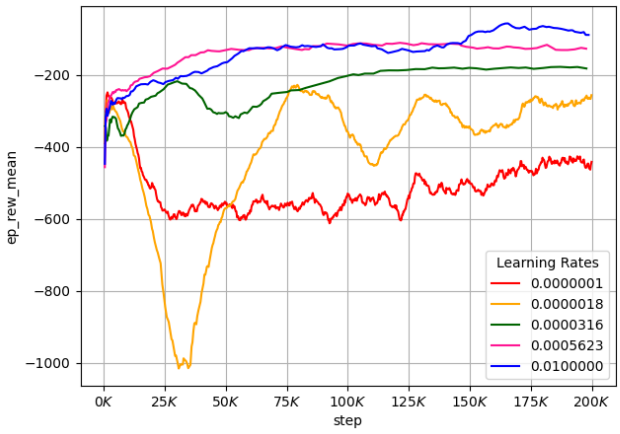
A Note sur l'utilisation de l'IA générative

L'IA générative, par l'intermédiaire de Gemini, a seulement servi pour de la recherche documentaire et pour améliorer la compréhension du module `Stable-Baselines3` par la génération d'un code minimal d'exemple, duquel nous nous sommes inspirés avec beaucoup de distance et de précaution pour construire la boucle d'entraînement. Il a également servi pour comprendre comment afficher des GIF dans le Readme.

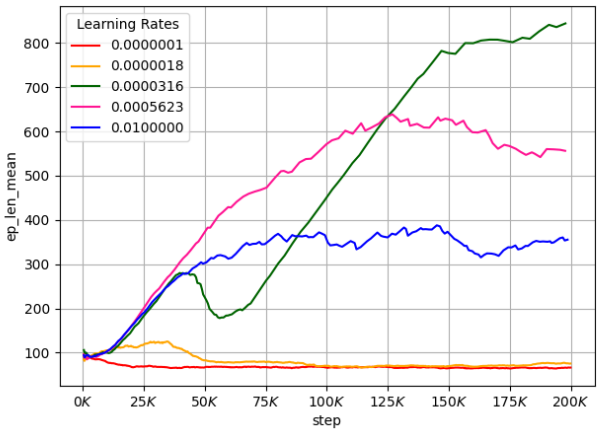
B Résultats graphiques issus de l'entrainement

B.1 Problème WINDY

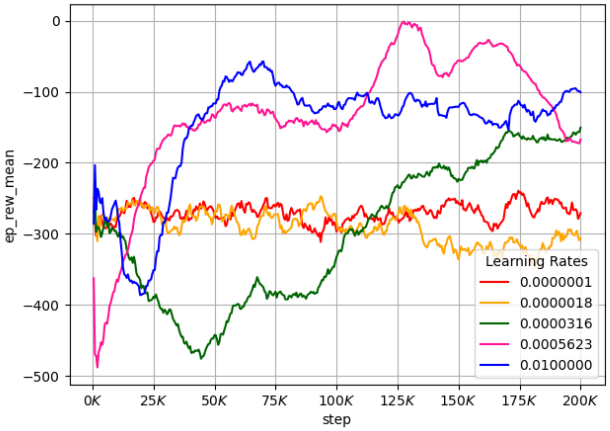
B.1.1 Variante VANILLA



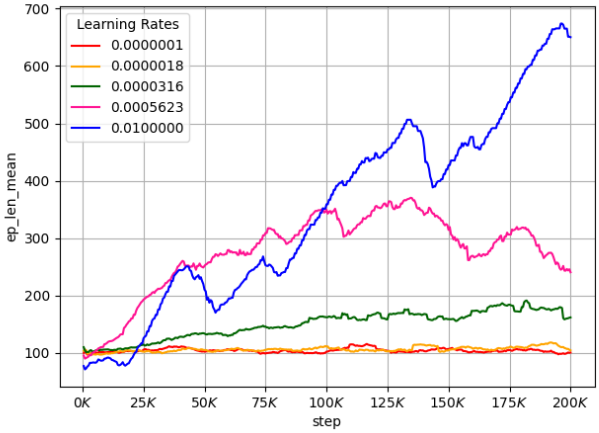
(a) DQN - Récompense moyenne



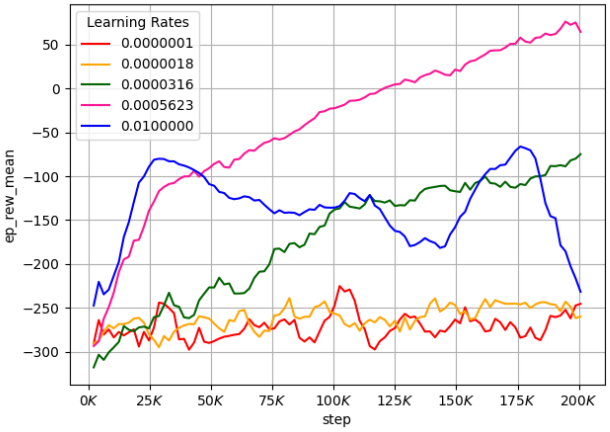
(b) DQN - Longueur moyenne d'épisode



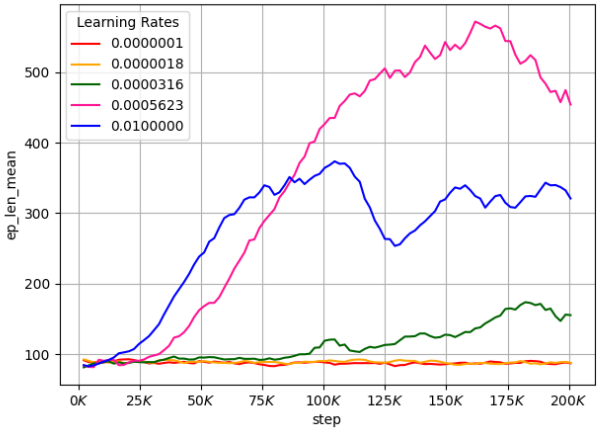
(c) A2C - Récompense moyenne



(d) A2C - Longueur moyenne d'épisode



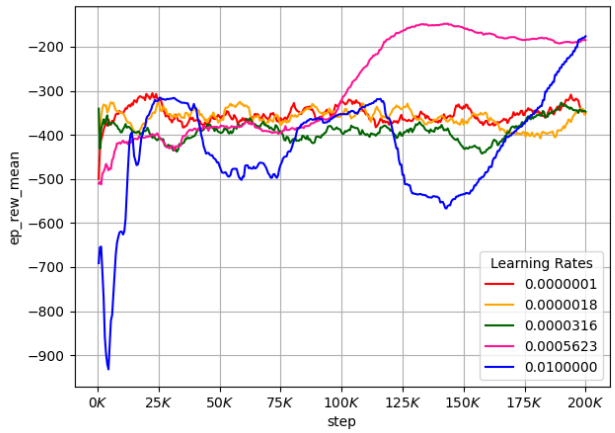
(e) PPO - Récompense moyenne



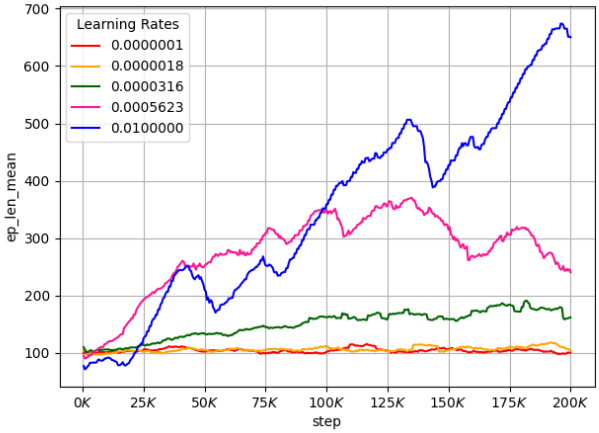
(f) PPO - Longueur moyenne d'épisode

FIGURE 6 – Résultats comparatifs sur la variante WINDY VANILLA du Lunar Lander

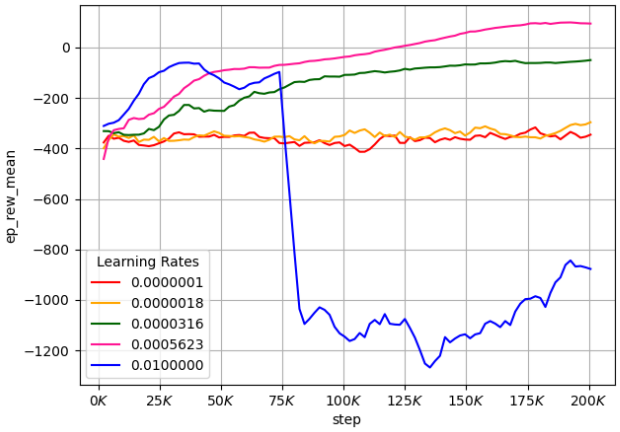
B.1.2 Variante `CONTINUOUS`



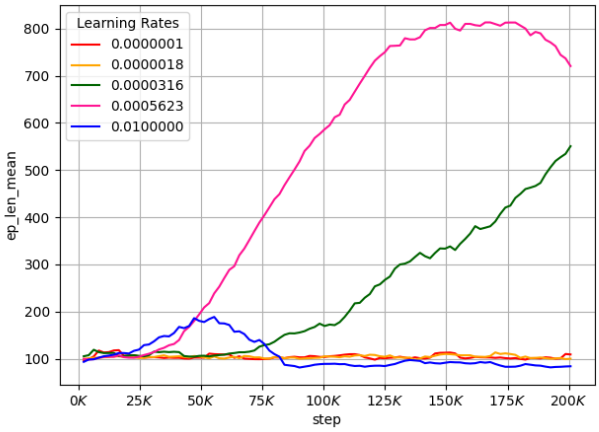
(a) A2C - Récompense moyenne



(b) A2C - Longueur moyenne d'épisode



(c) PPO - Récompense moyenne

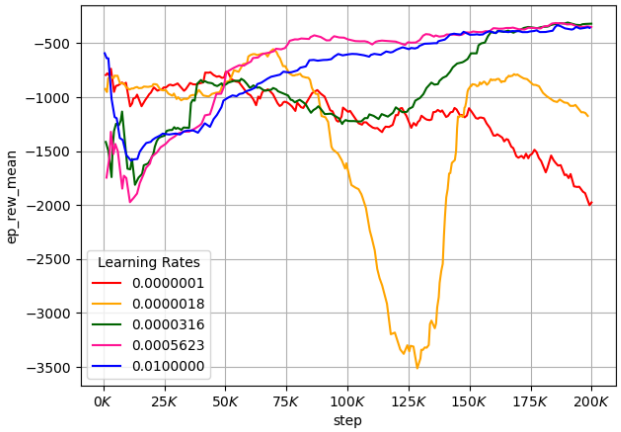


(d) PPO - Longueur moyenne d'épisode

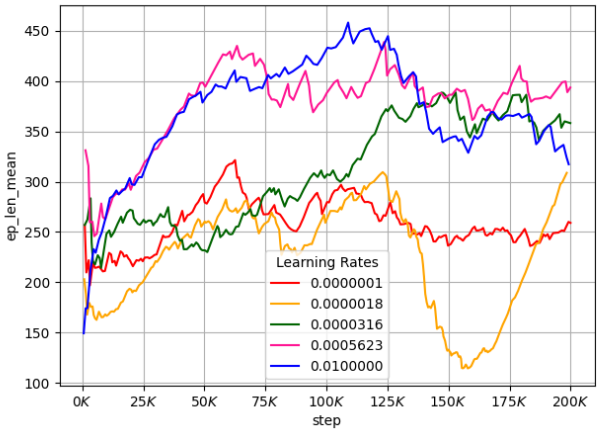
FIGURE 7 – Résultats comparatifs sur la variante `WINDY CONTINUOUS` du Lunar Lander

B.2 Problème ZERO-GRAVITY-WINDY

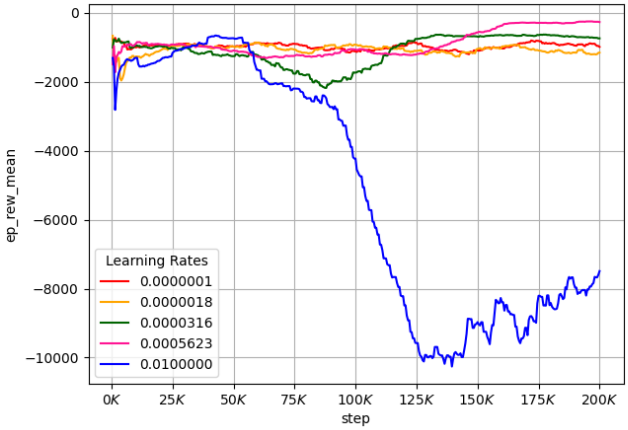
B.2.1 Variante VANILLA



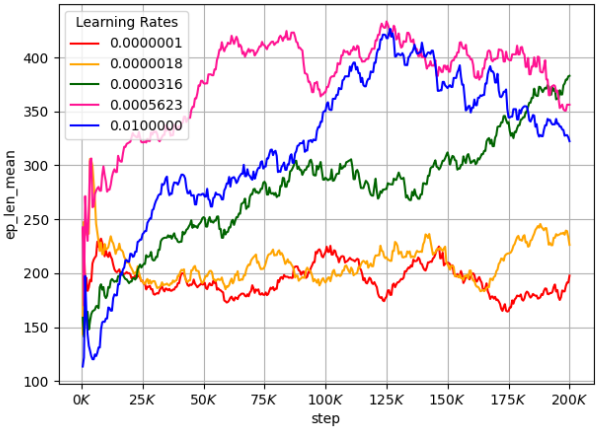
(a) DQN - Récompense moyenne



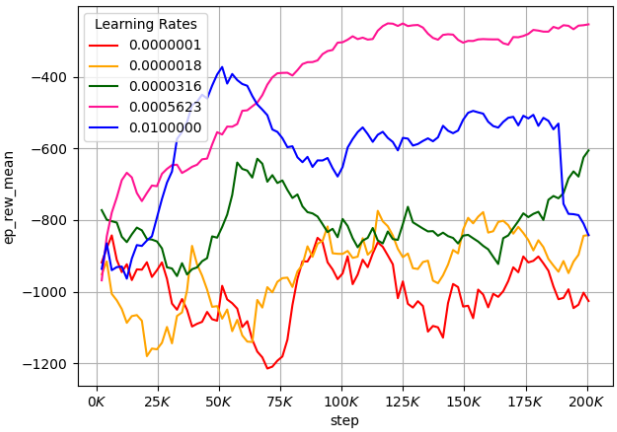
(b) DQN - Longueur moyenne d'épisode



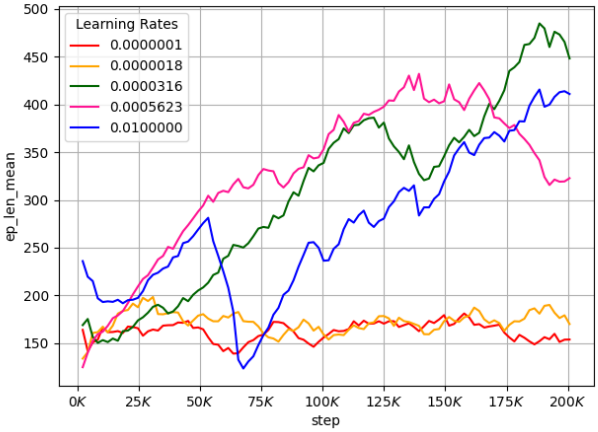
(c) A2C - Récompense moyenne



(d) A2C - Longueur moyenne d'épisode



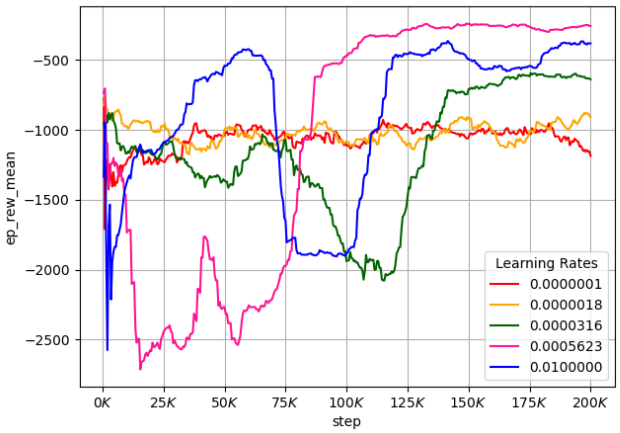
(e) PPO - Récompense moyenne



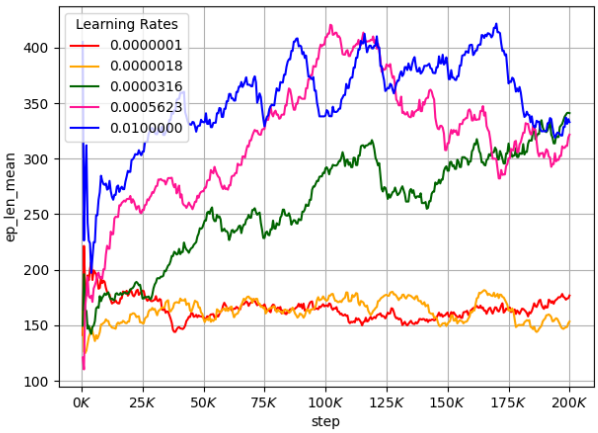
(f) PPO - Longueur moyenne d'épisode

FIGURE 8 – Résultats comparatifs sur la variante ZERO-GRAVITY-WINDY VANILLA du Lunar Lander

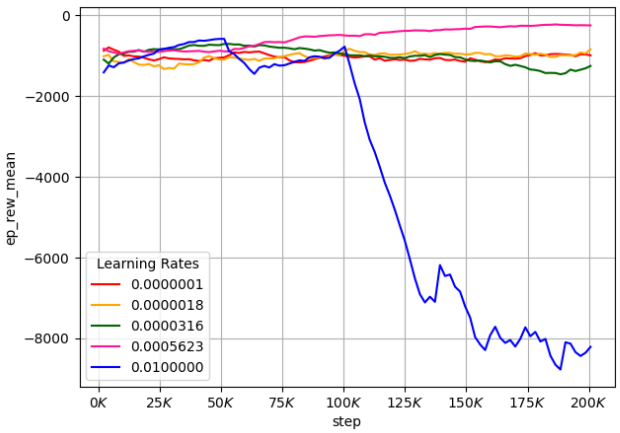
B.2.2 Variante `CONTINUOUS`



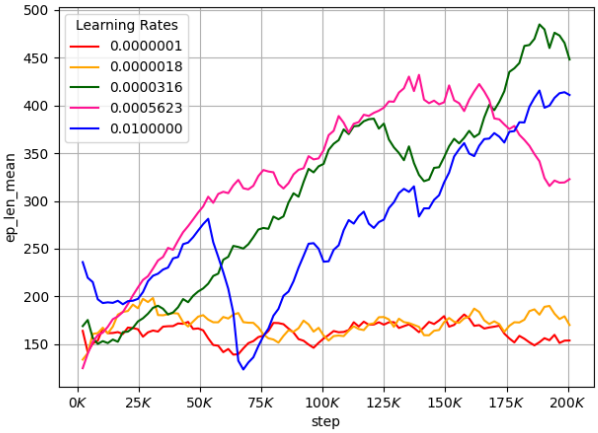
(a) A2C - Récompense moyenne



(b) A2C - Longueur moyenne d'épisode



(c) PPO - Récompense moyenne



(d) PPO - Longueur moyenne d'épisode

FIGURE 9 – Résultats comparatifs sur la variante `ZERO-GRAVITY-WINDY CONTINUOUS` du Lunar Lander

Références

[1] **Documentation de la librairie Stable-Baselines3**

[2] **Documentation de la librairie Gymnasium**

[3] **Module de cours d'apprentissage par renforcement**, Stéphane AIRIAU