APPRENTISSAGE PAR RENFORCEMENT SÉCURITAIRE

PROJET FINAL – IFT-7201

A PREPRINT

Mouhamed Gando Diallo
IFT-7201 – Université Laval
mouhamed-gando.diallo.1@ulaval.ca

Renaud Djekornonde Raoudel
IFT-7201 – Université Laval
renaud.djekornonde-raoudel.1@ulaval.ca

Université Laval
Maîtrise en informatique – Intelligence artificielle
Session d'hiver 2025

Introduction

L'apprentissage par renforcement (RL) est un paradigme puissant pour l'apprentissage séquentiel par interaction avec un environnement. Il a démontré son efficacité dans de nombreux domaines tels que les jeux, la robotique ou encore l'optimisation de systèmes dynamiques. Toutefois, dans des environnements sensibles ou critiques — par exemple industriels, médicaux ou robotiques — une erreur d'apprentissage ou une mauvaise décision peut entraîner des conséquences coûteuses, voire dangereuses.

Face à ces enjeux, l'apprentissage par renforcement sécuritaire (Safe RL) vise à développer des algorithmes capables non seulement d'optimiser la récompense cumulative, mais aussi de respecter des contraintes de sécurité durant l'entraînement et l'exécution. Ce domaine, en pleine croissance, tente de concilier performance et fiabilité dans des contextes réels où l'exploration libre n'est pas acceptable.

Dans ce projet, nous nous intéressons au problème spécifique du transfert sécurisé d'une politique apprise dans un environnement simulé vers un environnement réel plus risqué. Plus précisément, nous implémentons un mécanisme de protection actif, appelé *shielding*, qui surveille les actions proposées par un agent entraîné avec SAC (Soft Actor-Critic), et les bloque si elles risquent d'entraîner l'agent dans des états dangereux.

Nous évaluons l'efficacité de cette approche dans une variante modifiée de l'environnement `Pendulum-v1`, appelée `PendulumDangerous-v1`, dans laquelle certaines configurations angulaires sont considérées comme critiques. L'objectif est de mesurer l'impact du shielding sur la sécurité et la performance globale de l'agent.

Revue de la littérature

L'apprentissage par renforcement sécuritaire (Safe Reinforcement Learning, SRL) vise à apprendre des politiques efficaces tout en respectant des contraintes de sécurité, particulièrement critiques dans des contextes industriels où des actions peuvent entraîner des conséquences catastrophiques. Dans cette section, nous présentons trois variantes distinctes du problème de SRL explorées dans la littérature récente.

Méthodes, théories et applications du SRL (vue d'ensemble) Gu et al. [**?**] proposent une revue structurée des approches modernes en apprentissage par renforcement sécuritaire. Les auteurs formalisent cinq grandes dimensions du SRL (résumées par l'acronyme "2H3W") : *Why* SRL est nécessaire, *What* à sécuriser, *How* sécuriser l'apprentissage, *When* sécuriser (apprentissage vs exécution), et *Where* appliquer ces méthodes. L'article couvre à la fois les fondements théoriques (notamment les garanties de sécurité et la complexité échantillonnale) et les avancées algorithmiques, tout en offrant un panorama des domaines d'application et des environnements de test couramment utilisés. Ce travail constitue une référence importante pour comprendre la diversité des approches, ainsi que les compromis entre performance et sécurité. Il permet également d'identifier les lacunes actuelles et les pistes de recherche futures dans le domaine du SRL.

SRL appliqué au contrôle optimal industriel Lu et al. [**?**] présentent une approche de SRL conçue pour le contrôle optimal dans un procédé industriel de grillage de minerai d'or. L'algorithme proposé satisfait des contraintes de chance conjointes avec une haute probabilité, assurant ainsi que les politiques apprises respectent des seuils critiques de sécurité dans un contexte industriel réel. Cette étude met en évidence l'importance d'intégrer des contraintes opérationnelles spécifiques dans le processus d'apprentissage. En se basant sur un cas concret, les auteurs démontrent que des approches de RL peuvent être viables dans des systèmes à risques élevés si des garanties probabilistes sont imposées. Leur méthodologie fournit un exemple solide d'intégration entre théorie du SRL et applications industrielles réelles.

Transfert sécurisé de politiques simulées vers des systèmes réels Hsu et al. [**?**] s'attaquent à la problématique du transfert de politique, en introduisant une approche "Sim-to-Lab-to-Real" combinant apprentissage par renforcement avec un mécanisme de surveillance actif (shielding). Leur méthode repose sur une architecture à double politique : une politique primaire optimise la performance, tandis qu'une politique secondaire agit comme garde-fou pour prévenir les actions dangereuses. Ce cadre permet un transfert fiable entre simulation et déploiement dans des environnements réels, tout en maintenant des garanties de sécurité. Cette approche est particulièrement pertinente pour notre étude, car elle s'attaque explicitement aux risques liés au déploiement de politiques apprises en simulation dans des environnements dont les dynamiques sont imparfaitement connues. Elle combine élégamment robustesse, sécurité et généralisation, trois axes clés pour des systèmes réels critiques.

Formulation du problème

Le problème étudié dans ce projet s'inscrit dans le cadre de l'apprentissage par renforcement sécuritaire, avec un objectif spécifique : transférer une politique apprise dans un environnement de simulation vers un environnement réel, tout en minimisant les risques de comportements catastrophiques. Ces comportements sont définis comme des situations associées à des récompenses fortement négatives relativement à la moyenne, conformément à la définition fournie par le client.

Le problème est formalisé comme un processus de décision markovien (MDP) défini par le tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, où :

- $\mathcal{S}$ est l'espace des états,
- $\mathcal{A}$ est l'espace des actions continues,
- $P(s' \mid s, a)$ est la dynamique de transition entre états,
- $R(s, a)$ est la fonction de récompense,
- $\gamma \in [0, 1]$ est le facteur d'actualisation.

L'objectif classique est de maximiser le retour espéré :

$$G = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

Dans notre cas, cet objectif est soumis à une contrainte de sécurité portant sur la probabilité d'atteindre des états associés à une forte pénalité. La formulation devient alors :

$$\max_{\pi} \mathbb{E}_{\pi}[G] \quad \text{sous la contrainte} \quad \mathbb{P}_{\pi}(s_t \in \mathcal{S}_{\text{danger}}) \leq \delta, \quad \forall t$$

où $\mathcal{S}_{\text{danger}}$ désigne l'ensemble des états considérés comme dangereux, et δ est un seuil de tolérance au risque.

Dans notre projet, la notion de danger est associée à des angles extrêmes dans l'environnement `PendulumDangerous-v1`, représentant des configurations instables. Pour améliorer la sécurité de l'agent, nous avons conçu un environnement modifié intégrant un *shielding*, qui permet de bloquer dynamiquement les actions menant à ces états dangereux. Ce mécanisme s'inscrit dans une logique d'exécution sécurisée après apprentissage.

Présentation des méthodes à étudier

Nous avons sélectionné deux approches complémentaires pour étudier notre problématique de transfert sécurisé de politiques. La première est une méthode classique de renforcement profond, largement utilisée et non spécifique à la sécurité. La seconde est une méthode spécialisée conçue pour traiter explicitement les risques liés au transfert de politiques.

Méthode classique : Soft Actor-Critic (SAC) Soft Actor-Critic (SAC) [**?**] est une méthode de RL moderne et efficace pour les environnements à actions continues. Elle repose sur l'apprentissage d'une politique stochastique qui maximise à la fois la récompense attendue et l'entropie de la politique. SAC est particulièrement adaptée aux environnements à haut degré de stochasticité et offre une bonne stabilité d'apprentissage. Elle constitue donc une base cohérente pour comparer des variantes sécuritaires qui en dérivent directement.

Nous utiliserons l'implémentation de SAC fournie par la bibliothèque `Stable-Baselines3` [1].

Méthode sécuritaire : Safe Reinforcement Learning with Shielding (Sim-to-Lab-to-Real) Nous adoptons la méthode proposée par Hsu et al. [**?**], qui combine apprentissage profond et mécanisme de *shielding* (surveillance active). Le principe est d'entraîner une politique principale dans la simulation, tout en la couplant à une politique de sécurité chargée de surveiller les actions proposées. Si une action risque de conduire à un état dangereux, la politique de sécurité intervient pour bloquer ou remplacer cette action. Ce cadre permet de garantir des limites de sécurité probabilistes lors du transfert vers l'environnement réel.

Ce type de méthode est compatible avec les environnements de type OpenAI Gym et peut être implémenté à l'aide de la bibliothèque `SafeRL` ou via une adaptation personnalisée dans `Stable-Baselines3`.

1. `https://github.com/DLR-RM/stable-baselines3`

Résumé des méthodes

Les deux approches partagent le même environnement d'entraînement (`Pendulum-v1`) et le même algorithme de base (SAC). La seule différence se situe au niveau de l'exécution : l'une agit librement, tandis que l'autre est supervisée via le *shielding*. Cela nous permet de comparer précisément les effets d'une surveillance post-apprentissage sur le comportement, la performance et la sécurité de l'agent.

Méthodologie expérimentale

Expérimentations

Entraînement

Le modèle a été entraîné avec l'algorithme SAC sur l'environnement `Pendulum-v1`, à l'aide de la bibliothèque `stable-baselines3`. Les principaux paramètres d'entraînement sont :

- Nombre de pas : `100 000`
- Taille du buffer : `100 000`
- Learning rate : `3e-4`
- Politique : `MlpPolicy`
- Algorithme : `Soft Actor-Critic`

Le modèle est ensuite sauvegardé et utilisé pour l'évaluation dans un environnement plus risqué : `PendulumDangerous-v1`.

Évaluation

L'évaluation est effectuée sur 5 épisodes, avec et sans l'activation du mécanisme de *shielding*. Les métriques observées sont :

- **Récompense cumulée** : somme des récompenses par épisode
- **Score de sécurité** : proportion d'actions non bloquées par le *shielding* (plus haut = plus sûr)
- **Blocages** : nombre d'actions jugées dangereuses et corrigées

Les scores de sécurité sont calculés uniquement en présence du *shielding*, en utilisant la méthode `.safety_score()`.

Résumé des résultats

- **Sans shielding** : récompense moyenne = `-1812.0`, score de sécurité = `0.0%`
- **Avec shielding** : récompense moyenne = `-991.0`, score de sécurité = `79.0%`

On observe que le *shielding* réduit fortement la probabilité d'actions dangereuses, au prix d'une légère dégradation de la performance. Le comportement de l'agent est beaucoup plus stable et évite les angles extrêmes susceptibles de générer des punitions sévères.

Transfert et Sécurité. La politique est testée dans `PendulumDangerous-v1`, un environnement avec $g = 15.0$, un moteur plus faible et un seuil d'angle dangereux à $160°$. Le Shielding bloque les actions si l'angle $> 130°$ ou $|\omega| > 8.0$.

Mésures. Pour chaque configuration :
- Réward moyen sur 5 épisodes
- Score de sécurité : proportion d'actions non bloquées

Résultats et graphiques

Nous présentons ici deux figures clés illustrant les différences de comportement entre les deux approches testées : avec et sans mécanisme de *shielding*.

Comparaison avec et sans Shielding. Le Shielding réduit la fréquence des comportements dangereux (score ˜ 82%) au détriment de la performance (réward plus faible).

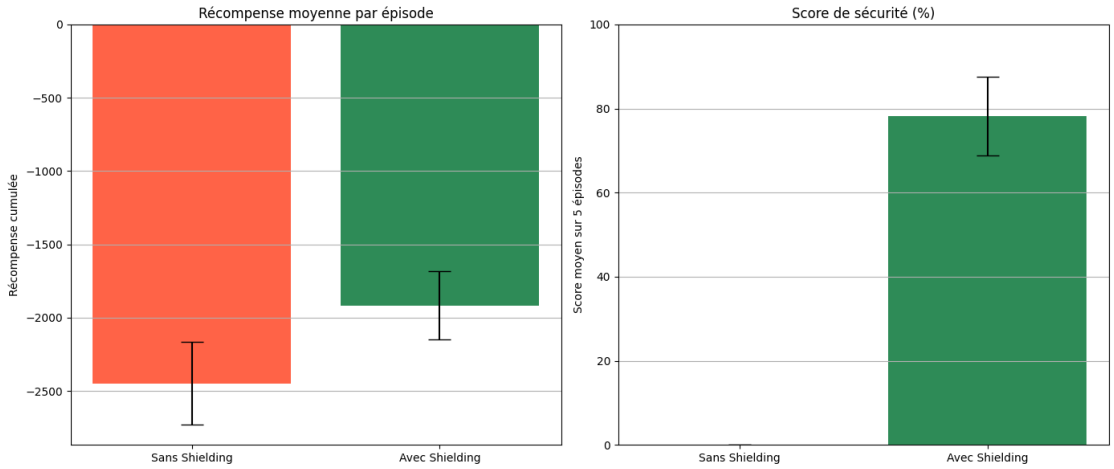

FIGURE 1 – Comparaison des récompenses moyennes obtenues avec et sans *shielding* (5 épisodes)

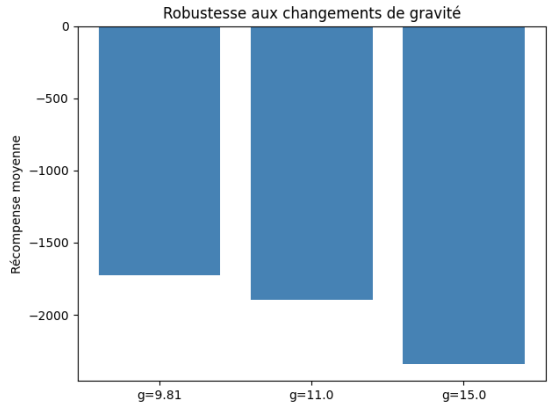

FIGURE 2 – Comparaison des scores de sécurité avec et sans *shielding*

Transfert Sim2Real sous différentes gravités. Quand la gravité augmente, la politique apprise en simulation dégrade fortement en performance. Cela met en évidence le besoin de robustesse et d'adaptation lors du transfert vers le réel.

Discussion

Les résultats obtenus montrent clairement l'effet du mécanisme de *shielding* sur le comportement de l'agent.

Sans aucune supervision, l'agent entraîné par SAC tend à produire des actions efficaces pour la stabilisation, mais il peut aussi conduire à des états extrêmes dans lesquels l'angle devient très élevé. Dans l'environnement `PendulumDangerous-v1`, ces configurations sont fortement pénalisées, ce qui explique les récompenses moyennes très faibles observées sans *shielding*.

Avec le mécanisme de *shielding*, les actions risquant de faire basculer l'agent dans des zones dangereuses sont bloquées et remplacées par une action neutre (couple nul). Ce mécanisme empêche de nombreux comportements catastrophiques, ce qui se traduit par :

- une hausse significative de la récompense moyenne,
- une réduction de la variance des récompenses,
- un score de sécurité élevé (supérieur à 75%).

Ce gain de sécurité s'accompagne toutefois d'un compromis : le *shielding* empêche parfois l'agent d'exécuter certaines actions potentiellement utiles à court terme, ce qui peut réduire sa performance optimale. Cependant, dans des systèmes

critiques (robotique, contrôle industriel, etc.), cette perte de performance est souvent acceptable au regard des risques évités.

Enfin, notons que ce *shielding* est purement réactif (post-apprentissage). Une extension naturelle consisterait à intégrer les contraintes de sécurité directement dans la fonction objectif pendant l'apprentissage, par exemple via une approche de type *constrained RL*.

Conclusion

Dans ce projet, nous avons étudié une approche sécuritaire en apprentissage par renforcement consistant à intégrer un mécanisme de *shielding* dans un environnement continu présentant des risques. À partir d'un agent SAC entraîné dans un environnement standard, nous avons évalué son comportement dans un environnement modifié `PendulumDangerous-v1`, à la fois avec et sans surveillance active.

Les résultats expérimentaux montrent que le *shielding* permet de réduire significativement les comportements dangereux, avec un score de sécurité supérieur à 75%, tout en maintenant des performances acceptables. Ce compromis entre performance et sécurité est essentiel dans les systèmes réels où la robustesse du comportement prime sur l'optimisation pure de la récompense.

Ce projet nous a permis de mettre en œuvre des notions fondamentales de RL, de comprendre l'importance des contraintes de sécurité, et de proposer une solution simple mais efficace pour limiter les risques sans modifier l'algorithme d'apprentissage lui-même.

En perspective, il serait intéressant d'intégrer ces contraintes directement dans la phase d'apprentissage (via des approches de type Constrained MDP ou Safe-SAC), ou d'étudier des méthodes d'apprentissage robuste capables de généraliser à des environnements réels plus imprévisibles.

Bibliographie

— Gu et al., *Safe Reinforcement Learning : A Survey*, 2022.
— Lu et al., *Safe RL with Chance Constraints in Gold Ore Processing*, 2023.
— Hsu et al., *Safe Continual Domain Adaptation after Sim2Real Transfer*, 2022.
— Haarnoja et al., *Soft Actor-Critic Algorithms*, 2018.

Projet GitHub

Le code complet du projet, incluant les expériences, les environnements et les scripts d'analyse, est disponible sur GitHub :

`https://github.com/gando537/Projet-RL-IFT-7201.git`