

MASTER 2 INTELLIGENCE ARTIFICIELLE

Compte rendu du TP 2 : Modélisation d'un Robot Nettoyeur avec un Processus Décisionnel de Markov

Membres :

Nouh CHELGHAM
Hocine Ouassim DJEKOUN
Mamady DJIGUINE

Chargé du cours :

Mme FARGIER HELÈNE

Table des matières

1	Introduction	2
2	Description du Modèle MDP	2
2.1	États :	2
2.2	Actions :	3
2.3	Probabilités de Transition	3
2.4	Récompenses	3
3	Résultats et Analyse	3
3.1	Politiques Optimales	3
3.2	Influence des paramètres	4
3.3	Politique Optimale quand pDust augmente	4
3.4	Amélioration du modèle	4
4	Conclusion	5

1 Introduction

Ce rapport présente la modélisation d'un robot ménager autonome utilisant un Processus Décisionnel de Markov (MDP). L'objectif est de déterminer la meilleure stratégie de nettoyage pour le robot afin de maximiser l'efficacité tout en minimisant les risques, tels que la panne de batterie et le maintien d'un salon propre.

Le robot évolue dans un environnement simple, composé du salon (qui peut être propre ou sale) et d'une station de charge. Il peut effectuer deux actions : nettoyer ou charger. Le choix de l'action optimale à chaque instant est crucial pour garantir un fonctionnement efficace du robot.

Dans ce rapport nous détaillons le modèle MDP utilisé, présentons les résultats obtenus pour différents scénarios et discutons de l'influence des paramètres du modèle sur la politique optimale du robot.

2 Description du Modèle MDP

Le modèle MDP du robot nettoyeur est défini par les éléments suivants :

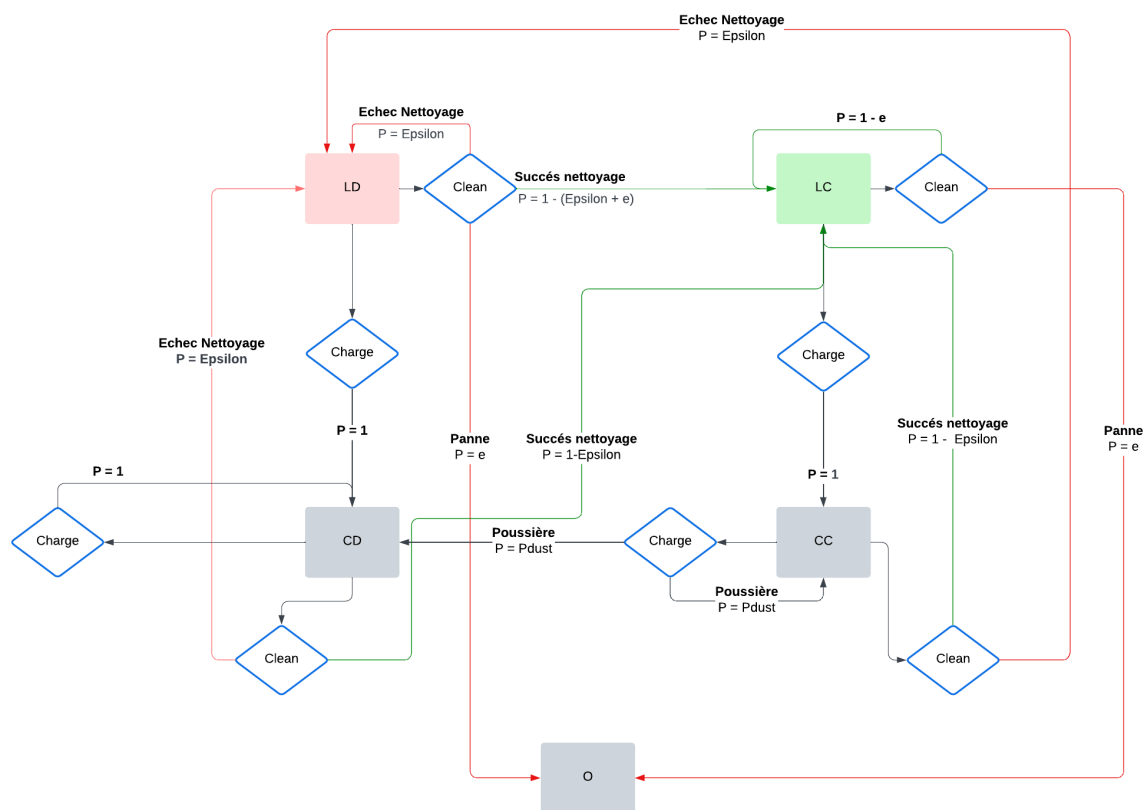


FIGURE 1 –

2.1 États :

- **LD** : Le robot est dans le salon et le salon est sale.
- **LC** : Le robot est dans le salon et le salon est propre.
- **CD** : Le robot est à la station de charge et le salon est sale.
- **CC** : Le robot est à la station de charge et le salon est propre.
- **O** : Le robot est hors tension (panne de batterie).

2.2 Actions :

- **Nettoyer(clean)** : Le robot tente de nettoyer le salon. Cette action peut entraîner une panne de batterie avec une probabilité 'e'.
- **Charger(charge)** : Le robot se déplace vers la station de charge pour recharger sa batterie. Pendant le chargement, le salon peut se salir avec une probabilité 'pDust'.

2.3 Probabilités de Transition

Les probabilités de transition entre les états dépendent de l'action choisie et des paramètres 'e', 'eps' (probabilité de panne et d'échec du nettoyage) et 'pDust'. Par exemple, si le robot est dans l'état **LD** et choisit de nettoyer, il peut : Rester dans l'état **LD** si le nettoyage échoue (**probabilité 'eps'**). Passer à l'état **LC** si le nettoyage réussit (**probabilité $1 - (eps + e)$**). Passer à l'état **O** si la batterie tombe en panne (**probabilité 'e'**).

2.4 Récompenses

- **rc** : Récompense pour être dans un état propre (LC ou CC).
- **rd** : Récompense pour être dans un état sale (LD ou CD). $rd < rc$
- **costcrash** : Récompense pour être hors tension (O). $costcrash \leq rd$

L'objectif du robot est de maximiser la somme des récompenses à long terme en choisissant la meilleure action à chaque instant.

3 Résultats et Analyse

3.1 Politiques Optimales

- **Probabilité de panne élevée (e = 0.95)** : Le robot privilégie le chargement dans la plupart des situations pour éviter la panne. Il ne nettoie que lorsqu'il est à la station de charge et que le salon est sale (CD).

```
Lorsque la probabilité d'être à court de batterie est élevée
rd = -10, rc = 10, costcrash = -10, e = 0.95, eps = 0.05, pDust = 0.5,
Policy:
LD: charger
LC: charger
CD: nettoyer
CC: nettoyer
O: nettoyer
```

FIGURE 2 –

- **Probabilité de panne faible (e = 0.3)** : La politique choisit de charger le robot dans les états "LD" (dans le salon, sale) et "LC" (dans le salon, propre). Ensuite, elle choisit de nettoyer trois fois consécutives dans les états "CD"(au chargeur, Sale), "CC"(au chargeur, propre) et "O"(Hors-Tension).

```
La proba Court batterie Faible
rd = -10, rc = 10, costcrash = -10, e = 0.3, eps = 0.05, pDust = 0.5,
Policy:
LD: charger
LC: charger
CD: nettoyer
CC: nettoyer
O: nettoyer
```

FIGURE 3 –

On explique ça par le fait que la politique évite le nettoyage dans le salon lorsqu'il est sale, pour minimiser le risque de panne d'électricité pendant le nettoyage."

- **Probabilité de panne faible ($e = 0.05$)** La politique choisit de nettoyer le robot dans les états "LD" (dans le salon, sale) et charger "LC" (dans le salon, propre). Ensuite, elle choisit de nettoyer trois fois consécutives dans les états "CD"(au chargeur, Sale), "CC"(au chargeur, propre) et "O"(Hors-Tension). La politique décide de nettoyer le robot en "LD" car la probabilité de panne est assez faible.

```
La proba Court de charge est très faible
rd = -10, rc = 10, costcrash = -10, e = 0.05, eps = 0.05, pDust = 0.3,
Policy:
LD: nettoyer
LC: charger
CD: nettoyer
CC: nettoyer
O: nettoyer
```

FIGURE 4 –

3.2 Influence des paramètres

- **e (probabilité de panne)** : Plus 'e' est élevé, plus le robot privilégie le chargement pour éviter la panne.
- **pDust (probabilité de poussière)** : Une valeur élevée de 'pDust' incite le robot à nettoyer plus fréquemment pour maintenir le salon propre.
- **rc (récompense pour un état propre)** : Augmenter 'rc' rend le nettoyage plus attractif pour le robot.

3.3 Politique Optimale quand pDust augmente

Après essai de plusieurs configurations de pDust nous remarquons que l'augmentation de pDust n'influence quasiment pas la politique optimale.

3.4 Amélioration du modèle

Le modèle est amélioré pour prendre en compte plusieurs niveaux de batterie (élevé, moyen, faible). La fonction `create_transition_matrices_batterie_levels` définit les matrices de probabilité de transition et de récompense pour le modèle amélioré. Les politiques optimales pour le modèle amélioré sont trouvées à l'aide d'itérations de politiques et dépendent des niveaux de batterie et des probabilités d'événements.

```

rd = -10, rc = 10, costcrash = -10, e = 0.4, eps = 0.05, pDust = 0.8,
Policy:
LD_L: charger
LD_M: nettoyer
LD_H: nettoyer
LC_L: charger
LC_M: charger
LC_H: charger
CD_L: charger
CD_M: nettoyer
CD_H: nettoyer
CC_L: nettoyer
CC_M: nettoyer
CC_H: nettoyer
0: nettoyer

```

FIGURE 5 – Résultat de l’amélioration du modèle

4 Conclusion

Ce rapport présente la modélisation d’un robot nettoyeur à l’aide d’un MDP. Les résultats obtenus montrent que la politique optimale du robot dépend fortement de la probabilité de panne de batterie. Lorsque cette probabilité est élevée, le robot privilégie le chargement pour éviter la panne. Plustard le modèle amélioré prend en compte plusieurs niveaux de batterie, ce qui rend le modèle plus réaliste.

Références

- [1] Markov Decision Process (MDP) Toolbox <https://pymdptoolbox.readthedocs.io/en/latest/api/mdp.html>
- [2] Processus de décision markovien https://fr.wikipedia.org/wiki/Processus_de_d%C3%A9cision_markovien