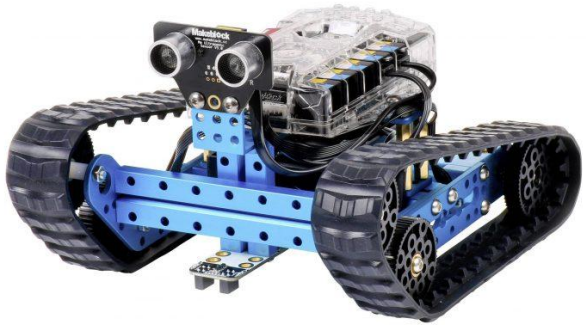
Soutenance d'IA

Présentation du projet Robot / Prédateur

DHENIN Alexandre
JURCZAK Benjamin
TROSSEVIN Florian



Lundi 10 Janvier 2022

Sommaire

1) Problèmes décisionnels de Markov
2) Modélisation du problème
3) Algorithme sous-jacent
4) Algorithme d’itération par valeur
5) Conclusion

I Problèmes décisionnels de Markov

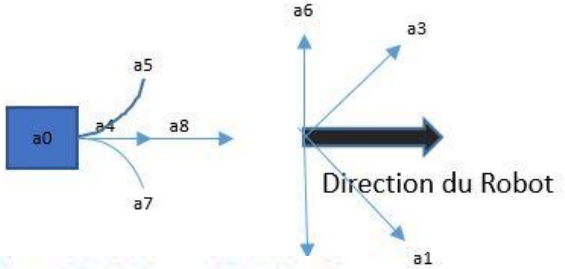
S (Ensemble des états) : info du carré perceptif à t et t-1

```
>>> etat[10]
[[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0,
0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]
, [[0, 0, 0, 0, 0], [0, 0, 0, 0, 1], [0, 0,
0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]
]

>>> etat[10][0]
[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0
, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]

>>> etat[10][1]
[[0, 0, 0, 0, 0], [0, 0, 0, 0, 1], [0, 0, 0
, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]
```

26 positions de la souris possible

donc 26*26 = 676 Etats

A (Ensemble des actions) : les 9 combinaisons possibles des roues




```
A = [ (0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2) ]
```

I Problèmes décisionnels de Markov

P matrice de transition entre les Etats

80 % de chance d'aller dans la direction souhaité ; 10 % *2 sur les côtés opposés

Fonction de récompense R

+5 si la proie est au centre du carré perceptif

-1 si la proie est sur une des bordures

T (Horizon), considéré infini

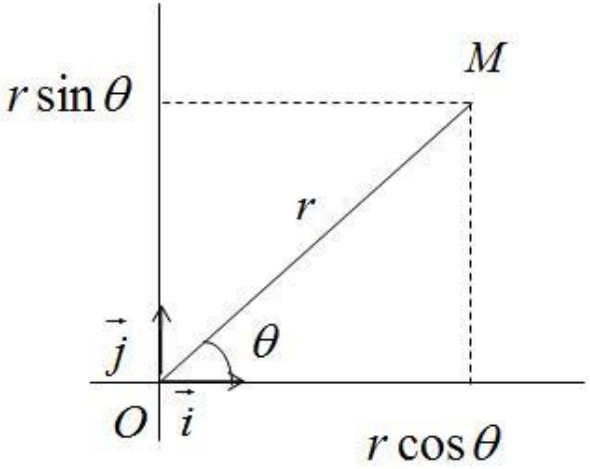
On supposera de plus les politiques déterministes

II Modélisation du problème

On travail dans un repère fixe et on va considérer un nouveau repère pour la souris

a / Déplacement de la souris

On suppose que le déplacement de la souris est de 1 à chaque itération (au sens de la norme 1)



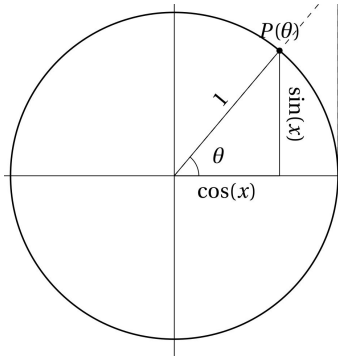
Déplacement linéaire :

On choisit l'origine dans le carré perceptif du robot, theta est fixe et on incrémente r au fur et à mesure

II Modélisation du problème

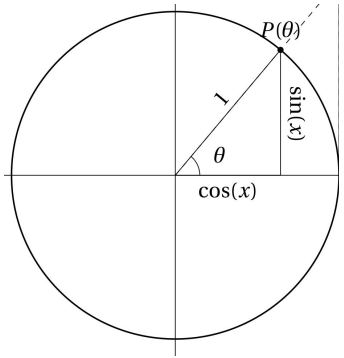
a / Déplacement de la souris

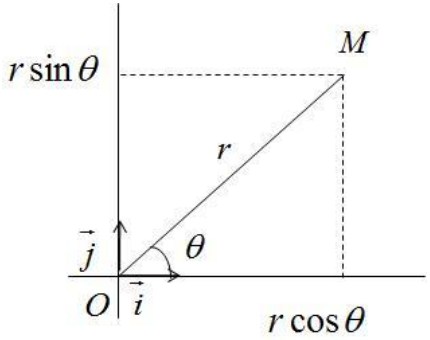
Déplacement circulaire



On choisit le rayon et un thêta initial et nous augmentons progressivement theta afin de nous déplacer d'une distance de 1 sur la corde

Déplacement aléatoire





Etant donné une position de la souris, on se déplace aléatoirement d'une distance de 1, sur le cercle autour du point en question

II Modélisation du problème

b / Robot et carré perceptif

Carré perceptif centré sur la position du robot

Orienté selon l'orientation du robot

0 degres initialement puis k*45degres

III Algorithme sous-jacent

a / Déplacement du robot

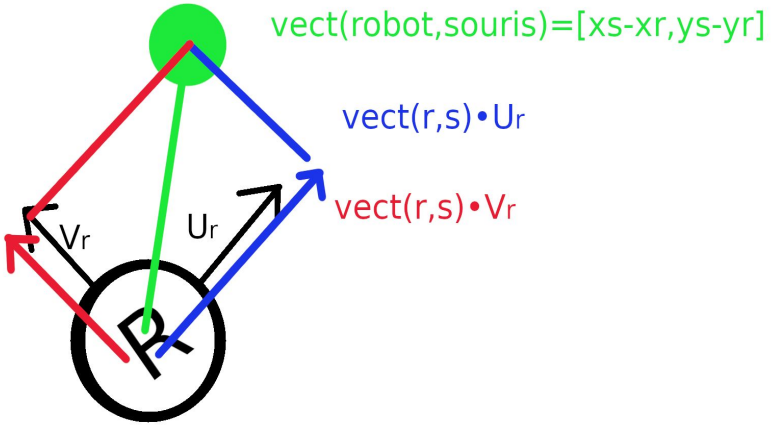
On établit la base orthonormé centré et orienté sur le robot

On fait les déplacements à l'aide des nouveaux vecteurs directeurs et on revient dans l'ancienne base

III Algorithme sous-jacent

b / Carré perceptif

La matrice perceptive est dans la base du robot, on obtient donc la position relative de la souris dans la nouvelle base en faisant vect(robot,souris) et en appliquant le produit scalaire avec les vecteurs orthonormés de la base du robot.



III Algorithme sous-jacent

b / Carré perceptif

On calcul la distance en norme infinie entre le robot et la souris pour vérifier que la souris se trouve dans le carré perceptif

On applique ensuite cette formule : i=np.floor(y+2.5),j=np.floor(x+2.5) afin d'obtenir la position de la souris dans le carré perceptif

IV Algorithme d'itération par valeurs (value iteration)

a / Objectif

→ Pour chaque état:

déterminer la politique à adopter

→ But: à chaque état, associer la meilleure des 9 actions possibles

→ Déplacement optimal du robot

1. Initialiser toutes les valeurs $V_0(i)$ pour état $i \in X = \{1, 2, \ldots, N\}$ avec une valeur arbitraire.

   $\varepsilon > 0$, $t = 0$, $\gamma \leq 1$
2. Pour chaque valeur d'état $i \in \{1, 2, \ldots, N\}$, calculer $V_{t+1}(i)$ à partir de :

$$V_{t+1}(i) = \min_{a_{ik} \in A_i} \left\{ c(i, a_{ik}) + \gamma \sum_{j=1}^{N} p_{ij}(a_{ik}) V_t(j) \right\}$$

3. Si $\|V_{t+1} - V_t\| < \varepsilon(1-\gamma)/2\gamma$ alors aller à l'étape 4 sinon $t = t+1$ et retourner à l'étape 2
4. Pour tous les états $i \in X = \{1, 2, \ldots, N\}$ et toutes les actions $a_{ik} \in A_i$, calculer les Q-valeurs

$$Q^*(i, a_{ik}) = c(i, a_{ik}) + \gamma \sum_{j=1}^{N} p_{ij}(a_{ik}) V^*(j)$$

5. Déterminer la politique optimale comme la politique gloutonne pour V*(i)

$$\mu^*(i) = \arg\min_{a_{ik} \in A_i} Q^*(i, a_{ik})$$

pour tous les états $i \in X = \{1, 2, \ldots, N\}$

IV Algorithme d'itération par valeurs (value iteration)

b / Implémentation

→ 4 paramètres pour représenter V car il prend comme valeur 2 position (t-1 et t)

→6 paramètres pour Q car il prends en plus la vitesse des roues

V Conclusion

→ Difficulté d'adaptation de l'algorithme d'itération sur les valeurs sur ce problème

→ Ambition sur le choix du sujet: se serait mieux traité avec plus de temps (meilleures bases)

→ Le robot préfère attendre une aide du vent lorsque la proie est derrière lui