

Stabilisation de Quadcopter par apprentissage artificiel

Minh Tuan VO et Ramy CHEMAK

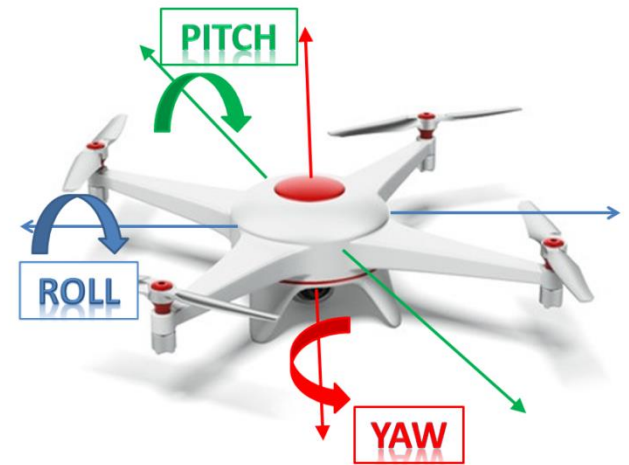


INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
CENTRE VAL DE LOIRE

Contexte physique

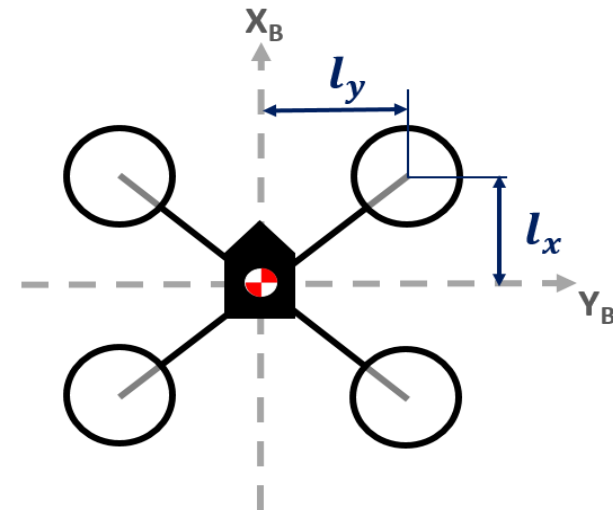
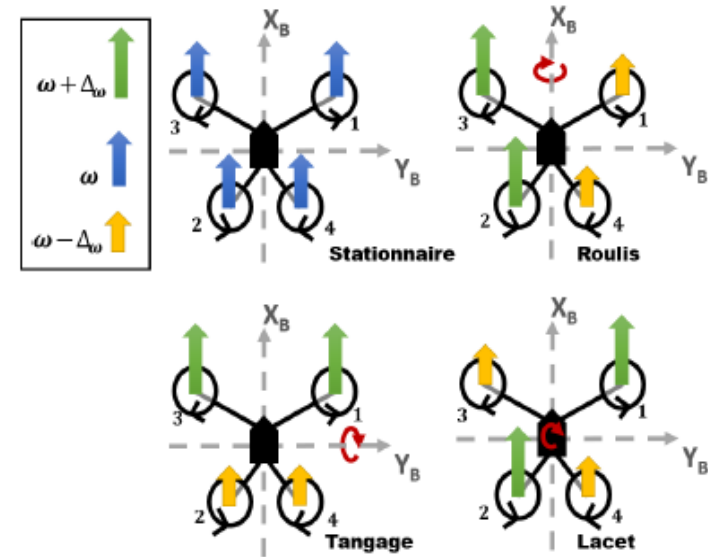
Quatre variables aérodynamiques :

- La force de poussée T
- Le mouvement de roulis (angle θ)
- Le mouvement de tangage (angle ϕ)
- Le mouvement de lacet (angle Ψ)



Contexte physique

- Mouvements associés à une modulation des vitesses des 4 moteurs des hélices
- Constantes aérodynamiques (l_x , l_y , k_f , k_c)
- Formules de la force de poussée et des forces de trainée



Contexte physique

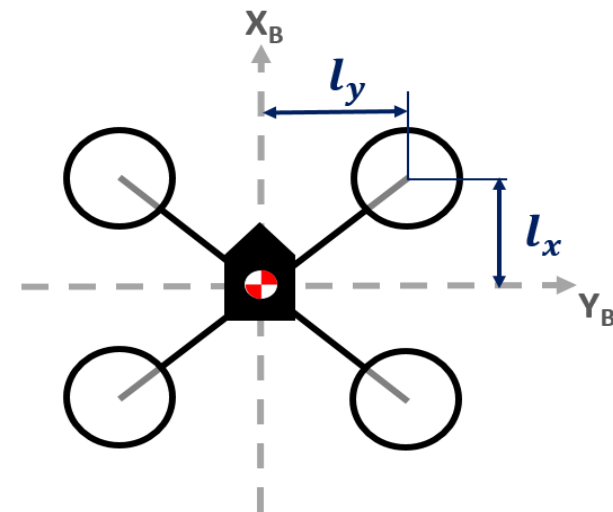
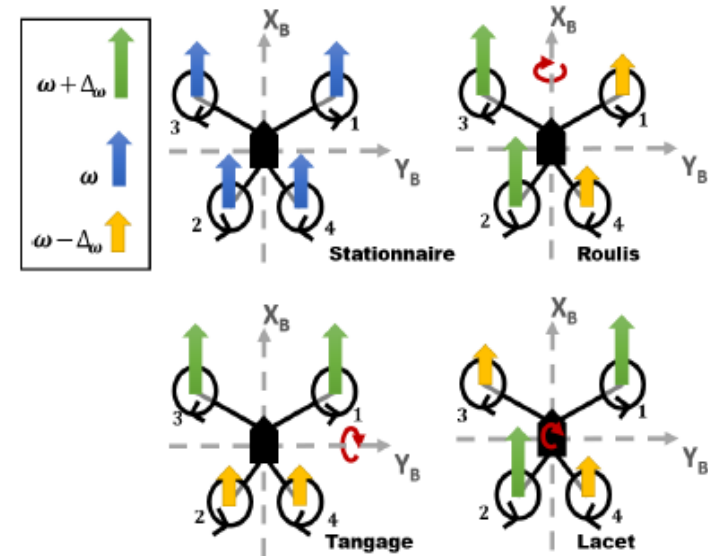
- Mouvements associés à une modulation des vitesses des 4 moteurs des hélices
- Constantes aérodynamiques (l_x , l_y , k_f , k_c)
- Formules de la force de poussée et des forces de trainée

$$T = k_f (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$$

$$\tau_\varphi = k_f l_y (-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2)$$

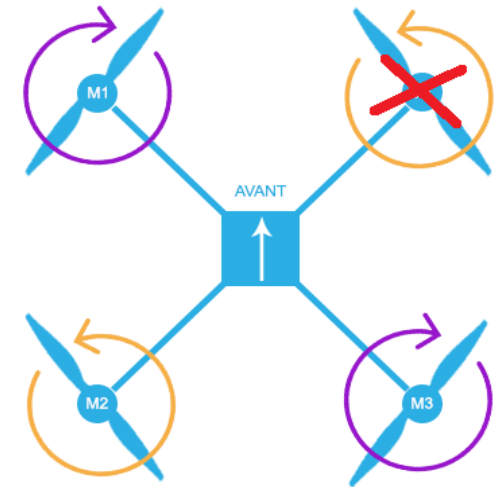
$$\tau_\theta = k_f l_x (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$$

$$\tau_\psi = k_c (\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2)$$



Problématique

- Perte de stabilité
- Difficulté de contrôle des 4 variables aérodynamique
- Absence de modèle ou formule physique



Définition du problème

- Espace des états : tous les tuples $(\Psi, \theta, \varphi, T)$
- Etat initial : n'importe quel tuple $(\Psi, \theta, \varphi, T)$
- Etat but : un tuple de valeurs fixes $(\Psi_{\text{ref}}, \theta_{\text{ref}}, \varphi_{\text{ref}}, T_{\text{ref}})$

Définition du problème

- Espace des états : tous les tuples $(\Psi, \theta, \varphi, T)$
- Etat initial : n'importe quel tuple $(\Psi, \theta, \varphi, T)$
- Etat but : un tuple de valeurs fixes $(\Psi_{\text{ref}}, \theta_{\text{ref}}, \varphi_{\text{ref}}, T_{\text{ref}})$

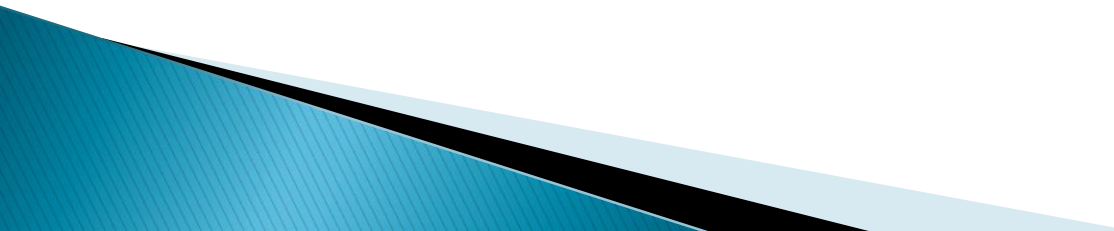
$$S = a^*(\theta - \theta_{\text{ref}}) + b^*(\varphi - \varphi_{\text{ref}}) + c^*(\Psi - \Psi_{\text{ref}}) + d^*(T - T_{\text{ref}})$$

Définition de l'environnement

- Entièrement observable
- Déterministe
- Mono-agent
- Discret
- Séquentiel
- Semi-dynamique



Solution proposée

- Un modèle de l'environnement connu, pas de solution analytique
 - Une simulation du modèle est possible
 - Le seul moyen de recueillir des données est l'interaction avec l'environnement
- 

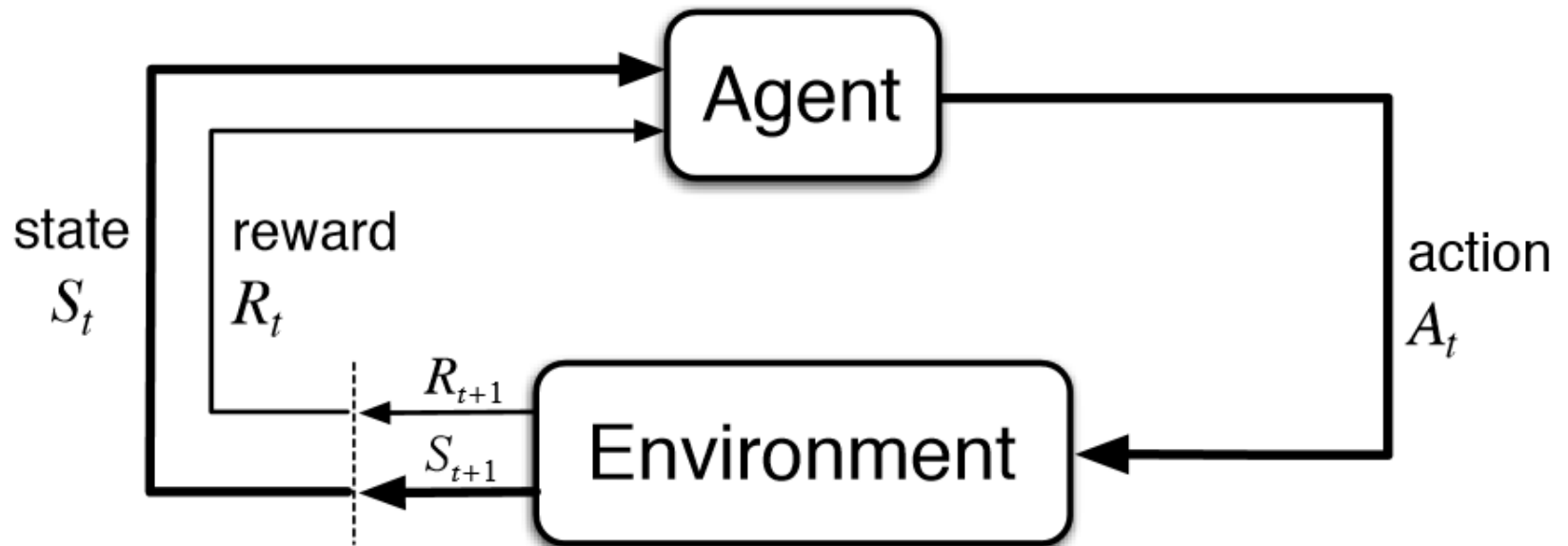
Solution proposée

- Un modèle de l'environnement connu, pas de solution analytique
- Une simulation du modèle est possible
- Le seul moyen de recueillir des données est l'interaction avec l'environnement

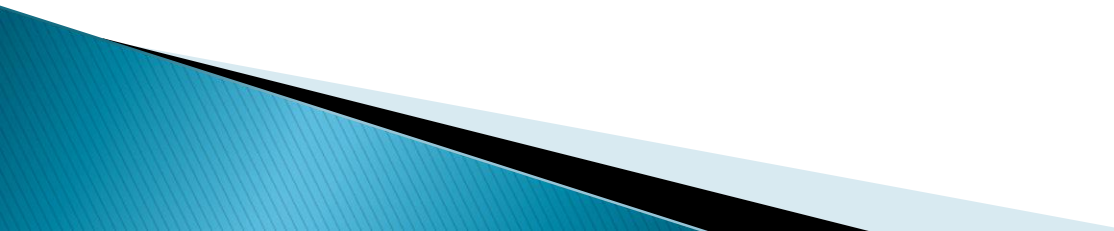
➔ Algorithme d'apprentissage par renforcement



Principe de l'algorithme

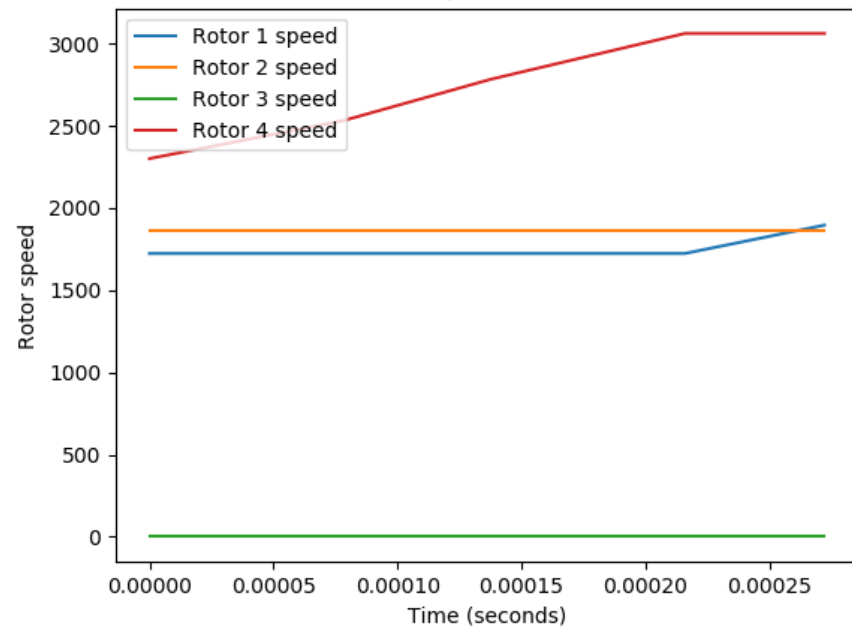


Algorithme orienté mécanique

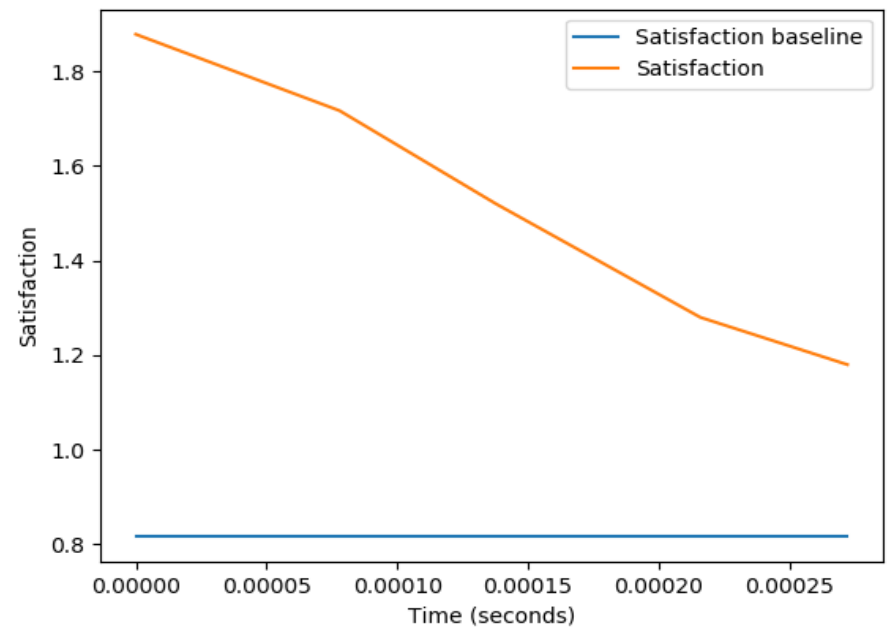
- Actions possibles
 - Récompenses
 - Politique de choix de l'action :
 1. Modulation aléatoire
 2. Maximisation de la récompense
 3. Maximisation de la récompense + Mouvements souhaités
- 

Simulation

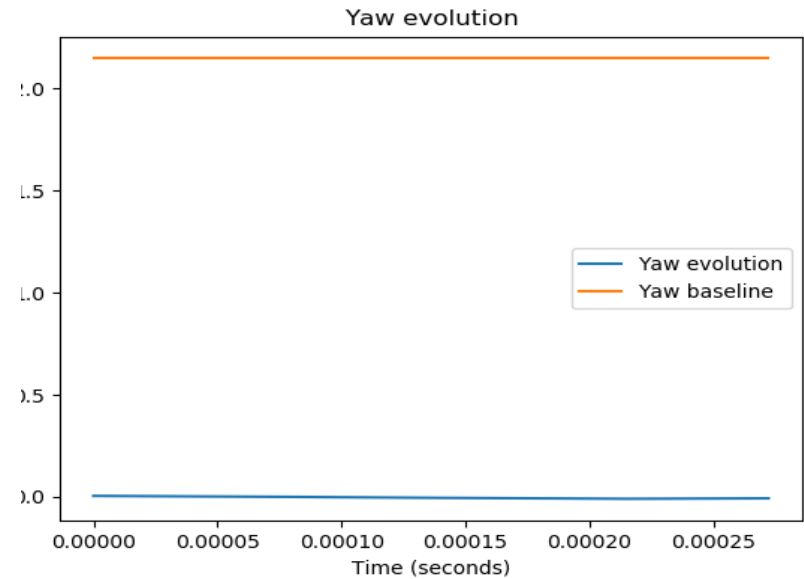
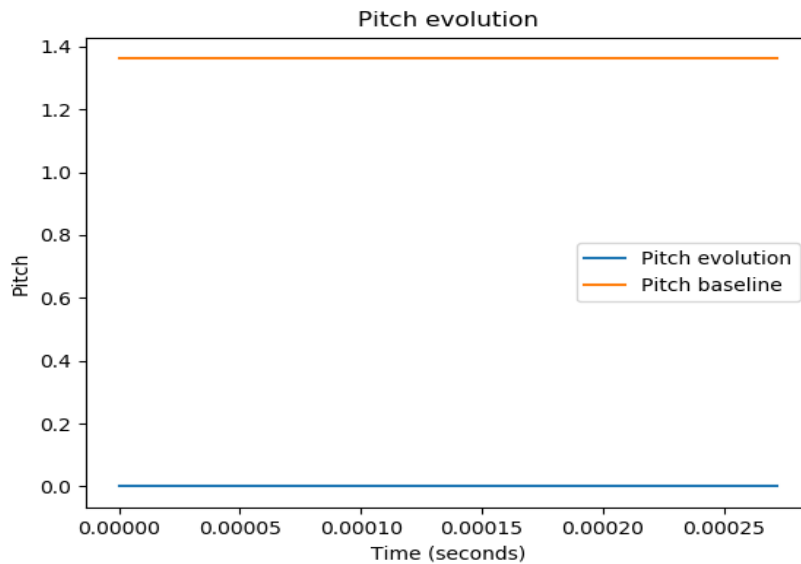
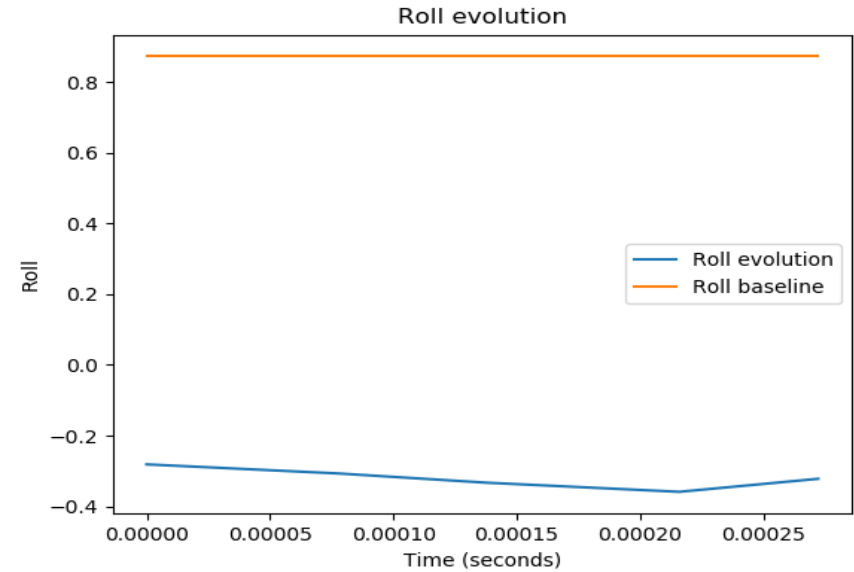
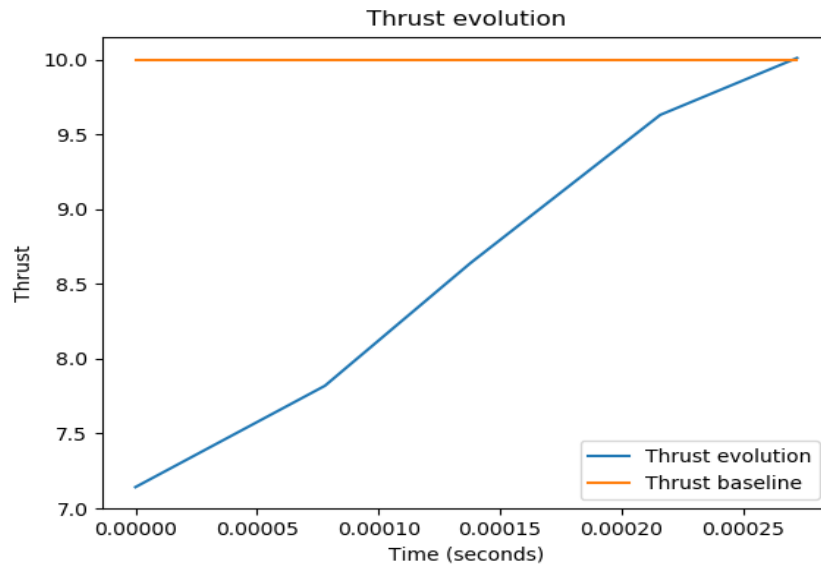
Rotors speed evolution



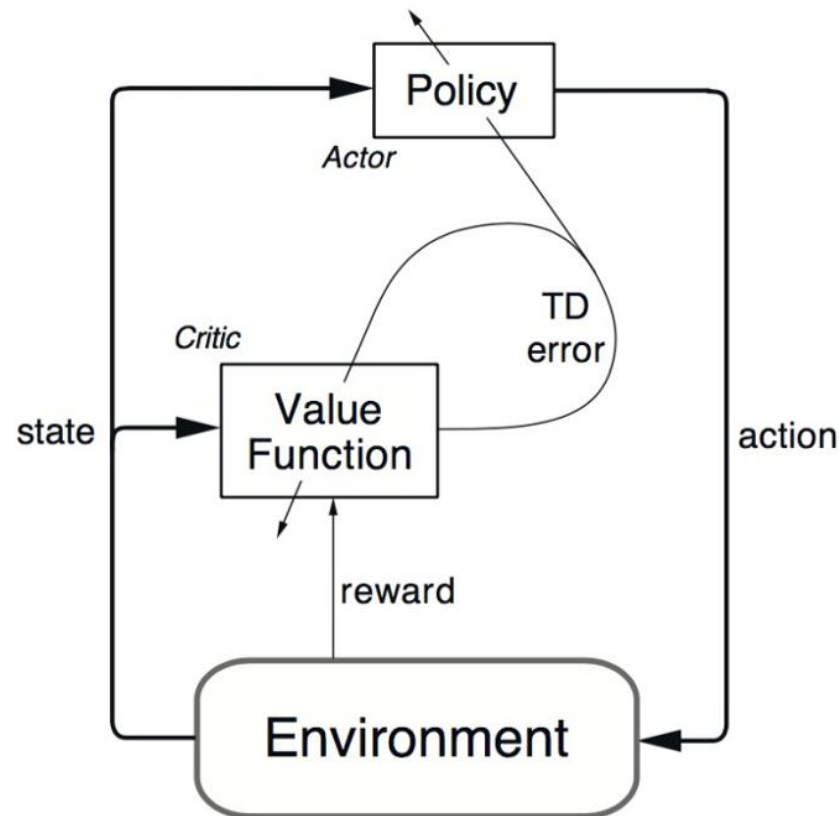
Satisfaction evolution



Simulation



Deep Deterministic Policy Gradients



Principe de l'algorithme

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ .

Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$

Initialize replay buffer R

for episode = 1, M **do**

 Initialize a random process \mathcal{N} for action exploration

 Receive initial observation state s_1

for t = 1, T **do**

 Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise

 Execute action a_t and observe reward r_t and observe new state s_{t+1}

 Store transition (s_t, a_t, r_t, s_{t+1}) in R

 Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R

 Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'}))|\theta^{Q'}$

 Update critic by minimizing the loss: $L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$

 Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

 Update the target networks:

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$$

end for

end for

Merci pour votre attention

