École Polytechnique / Sorbonne Université

Apprentissage par Renforcement Profond pour le Market Making

M2 Probabilité et Finance (Ex DEA El Karoui)

Realisé par : Ameur Nadir, El Moubaraki Anass, Lemales Hamza

Résumé : *Notre travail porte sur l'application de l'apprentissage par renforcement profond à un problème de market making optimal en s'appuyant sur l'algorithme Soft Actor-Critic (SAC). Nous commençons par modéliser le problème sous la forme d'un Markov Decision Process (MDP), en définissant rigoureusement l'espace des états, l'espace des actions et la fonction de récompense. Nous détaillons ensuite les principes sous-jacents à la dynamique du mid-price, ainsi que la théorie derrière le SAC. Enfin, nous effectuons des simulations et évaluons l'impact des ordres adverses sur les performances du market maker avant de discuter des limites de notre approche.*

Mots clefs : *Market Making, Trading Algorithmique, Apprentissage par Renforcement, MDP, SAC, Mid-Price, Processus de Hawkes.*

Table des matières

1 Introduction

Ce rapport présente une étude de l'application de l'apprentissage par renforcement profond (Deep Reinforcement Learning, DRL) au market making, une activité essentielle des marchés financiers consistant à fournir de la liquidité sur un actif en plaçant des ordres limites à l'achat (ask) et à la vente (bid), tout en gérant les risques d'inventaire et d'exécution. Le market maker se rémunère alors en capturant le spread, c'est-à-dire la différence entre le best bid et le best ask.

L'objectif principal est de concevoir un agent capable d'optimiser seul ses décisions de placement d'ordres en s'adaptant aux dynamiques de marché. Ce type d'algorithme se rapproche des stratégies de trading haute fréquence, où des systèmes automatisés doivent réagir et exécuter des ordres à l'échelle de la microseconde. Pour ce faire, notre étude repose sur l'algorithme Soft Actor-Critic (SAC), une approche basée sur l'optimisation de la politique d'action à travers une maximisation conjointe de l'entropie et du gain attendu.

Notre problème est formalisé sous la forme d'un Markov Decision Process (MDP), où l'espace des états est défini par le mid-price et l'inventaire du market maker. L'espace des actions regroupe les décisions possibles de l'agent, à savoir poster un ordre limite au best bid, poster un ordre limite au best ask ou s'abstenir d'agir. La fonction de récompense est quant à elle construite afin de maximiser le P&L tout en intégrant une pénalisation sur l'inventaire.

Une partie essentielle du rapport est consacrée à la modélisation de la dynamique du mid-price, qui est décrite à l'aide d'un processus de Hawkes, une classe de processus de comptage souvent utilisée pour représenter les mouvements de prix à haute fréquence. Par ailleurs, un cadre expérimental est mis en place afin d'évaluer la pertinence de notre modèle. L'étude se concentre également sur l'analyse de l'impact des ordres adverses, du spread et de la volatilité sur les performances du market maker.

2 Markov Decision Process (MDP)

Nous posons la quadruplet (S_t, a_t, T_t, r_t) où S_t désigne l'état actuel, a_t l'action prise par l'algorithme (market maker) et tirée selon une politique stochastique $\pi(.|S_t)$, T_t la probabilité de transition de l'état S_t à l'état S_{t+1} en prenant en compte l'action a_t effectuée, et r_t la récompense obtenue par l'agent et qui est une conséquence directe du triplet (S_t, a_t, S_{t+1}) où $S_{t+1} \sim T_{(.|S_t, a_t)}$.

2.1 Description de l'espace des états

L'espace des états est défini par le couple (P_t, Q_t^A), où P_t représente le mid-price, calculé comme la moyenne entre le best bid et le best ask, et Q_t^A correspond à l'inventaire du market maker.

Le prix P_t suit une dynamique comme dans [4] :

$$dP_t = \eta dt + \sqrt{\sigma^2 + \tilde{\sigma}^2 + \xi^2} dW_t. \tag{1}$$

Nous justifierons soigneusement ce choix par la suite. Comme vous pouvez le voir, nous ne prenons pas en compte le market impact de nos ordres. De plus, nous supposons un spread constant.

Concernant l'inventaire, nous imposons que le market maker ne puisse être à découvert ni détenir plus de q unités d'actifs. Ainsi :

$$(P_t, Q_t^A) \in \mathbb{R} \times \{-q, \dots, q\}.$$

.

2.2 Description de l'espace des actions

Dans notre modèle, l'espace des actions est discret. En notant A_t l'ensemble des actions possibles de l'agent à l'instant t, nous obtenons :

$$A_t = \begin{cases} \{0,1\} & \text{si } Q_t^A = -q, \\ \{1,0,-1\} & \text{si } -q < Q_t^A < q, \\ \{-1,0\} & \text{si } Q_t^A = q. \end{cases}$$

1. $Q_t^A = -q$: Le market maker ne peut plus placer d'ordre limite à l'ask.
2. $Q_t^A = q$: Le market maker ne peut plus placer d'ordre limite au bid.
3. Entre les deux : Le market maker a le choix de placer un ordre limite (bid ou ask) ou ne rien faire.

De plus, Nous supposons que le market maker place toujours ses ordres soit au best bid/ask, soit il ne prend aucune action. De plus, tous ses ordres sont de taille un et ne peuvent être exécutés qu'à l'instant suivant leur émission. S'ils ne sont pas exécutés à cet instant, ils sont annulés.

2.3 Description des récompenses

À partir de deux états $S_t = (P_t, Q_t^A)$, $S_{t+dt} = (P_{t+dt}, Q_{t+dt}^A)$ et pour une action a_t, l'agent obtient une récompense à l'instant $t + dt$ qui s'exprime de la façon suivante :

$$R_{t+dt} = (W_{t+dt}^A - W_t^A) - \alpha \left(|Q_{t+dt}^A| - |Q_t^A|\right)dt,$$

où $W_t^A = Q_t^A P_t + C_t^A$ où C_t^A représente le cash du market maker. Cette expression est une conséquence de la définition de la récompense totale à l'horizon T. En effet, l'objectif est de maximiser :

$$\mathbb{E}_\pi \left[W_T^A - \alpha \int_0^T |Q_t^A| dt \right].$$

α étant un paramètre de pénalisation.

Nous présentons dans la partie suivante une justification de la dynamique du mid-price.

3 Modélisation de la dynamique du mid-price

3.1 Dynamique ultra haute fréquence

Pour modéliser le mid-price, nous commençons par décrire la dynamique haute fréquence puis nous utilisons une loi forte des grands nombres et un théorème central-limite pour établir la dynamique diffuse de ce dernier.

En notant le mid-price P_t, ce dernier suit une dynamique de la forme :

$$P_t = P_0 + \sum_{k=1}^{N(t)} X_k.$$

avec δ un demi-tick, $(X_k)_k \in \{-\delta, \delta\}^{\mathbb{N}}$ une chaîne de Markov ergodique ayant une probabilité stationnaire π^* et $N(t)$ un processus de comptage que nous fixons comme étant un processus de Hawkes [2] d'intensité :

$$\lambda(t) = \lambda + \int_0^t \mu(t-s)dN_s.$$

où μ est une fonction d'excitation.

Pour passer d'une échelle très haute fréquence à une échelle plus basse fréquence, nous notons P_{nt} avec $n \in \mathbb{N}^*$ ($n = 10, 1000, \dots$). Nous avons alors $P_{nt} = P_0 + \sum_{k=1}^{N(nt)} X_k$.

3.2 Théorème centrale limite et loi forte des grands nombres

Sous des conditions de régularité de la fonction d'excitation nous établissons le théorème central limite décrit dans [7], [5] et [6] :

$$\frac{P_{nt} - N(nt)a^*}{\sqrt{n}} \xrightarrow{d} \sigma^* \sqrt{\mathbb{E}[N(0,1)]} W(t), \quad \text{avec} \quad a^* = \lim_{n\to\infty} \frac{\sum_{k=1}^n X_k}{n}. \tag{2}$$

où $W(t)$ est un processus de Wiener et

$$(\sigma^*)^2 = 4\delta^2 \left[\frac{1 - p' + \pi^*(p' - p)}{(p + p' - 2)^2} - \pi^*(1 - \pi^*) \right].$$

Nous faisons l'hypothèse que $\pi^* \longleftrightarrow \delta$ et $1 - \pi^* \longleftrightarrow -\delta$. Les termes p et p' désignent les probabilités de transition dans la chaîne de Markov telles que :

$$p = \mathbb{P}\left(X_{t+1} = \delta \mid X_t = \delta\right) \quad \text{et} \quad p' = \mathbb{P}\left(X_{t+1} = -\delta \mid X_t = -\delta\right),$$

Esquisse de preuve

$$P_{nt} = P_0 + \sum_{k=1}^{N(nt)} X_k,$$

Ainsi, nous avons :

$$P_{nt} - N(nt)a^* = P_0 + \sum_{k=1}^{N(nt)} (X_k - a^*),$$

$$\frac{P_{nt} - N(nt)a^*}{\sqrt{n}} = \frac{P_0}{\sqrt{n}} + \frac{\sum_{k=1}^{N(nt)} (X_k - a^*)}{\sqrt{n}},$$

Sachant que :

$$\lim_{n\to\infty} \frac{P_0}{\sqrt{n}} = 0 \quad \text{et} \quad \frac{\sum_{k=1}^{N(nt)} (X_k - a^*)}{\sqrt{n}} = \frac{N(nt)}{n} \frac{\sqrt{n}}{N(nt)} \sum_{k=1}^{N(nt)} (X_k - a^*),$$

Si l'on suppose que $\int_0^\infty |\mu(s)| ds < \infty$ alors :

$$\frac{N(nt)}{n} \xrightarrow{p.s} t\mathbb{E}[N(0,1)].$$

En effet, le caractère L^1 de la fonction d'excitation garantit la stabilité du processus de Hawkes et sa stationnarité asymptotique [2]. C'est le cas, par exemple, pour :

$$\mu(s) = \alpha e^{-\beta s}, \quad \alpha, \beta > 0 \quad \text{et} \quad \mu(s) = \frac{\alpha}{(s+c)^{\beta}}.$$

Enfin, en utilisant un résultat de convergence faible dans la topologie de Skorokhod [2], nous obtenons :

$$\sqrt{n}\frac{1}{N(nt)}\sum_{k=1}^{N(nt)}(X_k - a^*) \xrightarrow{d} \frac{\sigma^* W(t)}{t\sqrt{\mathbb{E}[N(0,1)]}}.$$

Remarque 1. Nous supposons que le processus de comptage est indépendant de la chaîne de Markov. Ainsi, une approche pour démontrer ce résultat consiste à combiner le théorème central limite (TCL) ergodique classique avec l'hypothèse d'indépendance et la stabilité du processus de Hawkes.

Le TCL que nous avons énoncé permet d'approcher la volatilité du mid price et donc le terme de diffusion. Pour approcher le drift nous établissons une loi forte des grands nombre (équation (3)).

$$\frac{P_{nt}}{nt} \xrightarrow{p.s} a^*\mathbb{E}[N(0,1)]. \tag{3}$$

Esquisse de preuve
Nous avons :

$$\frac{P_{nt}}{nt} = \frac{P_0}{nt} + \frac{\sum_{k=1}^{N(nt)} X_k}{nt},$$

Décomposons le second terme :

$$\frac{\sum_{k=1}^{N(nt)} X_k}{nt} = \frac{N(nt)}{nt} \times \frac{1}{N(nt)}\sum_{k=1}^{N(nt)} X_k,$$

Or, nous savons que :

$$\frac{N(nt)}{nt} \xrightarrow{p.s} \mathbb{E}[N(0,1)].$$

De plus, par ergodicité de la chaîne de Markov :

$$\frac{1}{N(nt)}\sum_{k=1}^{N(nt)} X_k \xrightarrow{p.s} a^*,$$

Par conséquent :

$$\frac{P_{nt}}{nt} \xrightarrow{p.s} a^*\mathbb{E}[N(0,1)].$$

Nous avons établi deux résultats donnant une dynamique asymptotique du mid-price avec :

1. un drift égal a $a^*\mathbb{E}[N(0,1)]t$,
2. une volatilité égale a $\sigma^{*2}\mathbb{E}[N(0,1)] + \sigma^2$.

où σ^2 est la volatilité intrinsèque du processus de comptage (N_t). En explicitant $\mathbb{E}[N(0,1)]$ et a^*, nous obtenons (équation (4)) :

$$\begin{cases} \mathbb{E}[N(0,1)] = \frac{\lambda}{1-\mu^*}, \quad \text{où} \quad \mu^* = \int_0^\infty \mu(s)ds < 1, \\ a^* = \delta\pi^* - \delta(1-\pi^*) = \delta(2\pi^* - 1), \quad \text{mesure ergodique de la fonction identité.} \end{cases} \tag{4}$$

La dynamique du mid-price étant justifiée, nous allons maintenant présenter le cadre théorique de l'algorithme SAC.

4 Algorithme SAC

SAC est une méthode d'apprentissage par renforcement off-policy basée sur l'approche actor-critic, intégrant un terme d'entropie (d'où le 'soft') dans la fonction valeur afin de favoriser des politiques stochastiques et améliorer l'exploration. L'algorithme repose sur deux étapes : un acteur qui apprend la politique optimale et un critique qui apprend la fonction valeur état/action.

Tout d'abord, nous définissons la fonction valeur état et la fonction valeur état/action comme suit :

$$V_\psi(S_t) = \mathbb{E}_{\pi_\phi}\left[R_t^\gamma \middle| S_t = s\right] = \mathbb{E}_{\pi_\phi}\left[\sum_{k=0}^{T-t} \gamma^k R_{t+k} \middle| S_t = s\right], \tag{5}$$

$$Q_\theta(S_t, A_t) = \mathbb{E}_{\pi_\phi}\left[R_t^\gamma \middle| S_t = s, A_t = a\right] = \mathbb{E}_{\pi_\phi}\left[\sum_{k=t}^{T-t} \gamma^k R_{t+k} \middle| S_t = s, A_t = a\right]. \tag{6}$$

Les paramètres ϕ, θ, ψ sont les poids des réseaux de neurones caractérisant chacune des composantes de l'acteur et du critique, à savoir respectivement (V, Q, π).

Pour favoriser le caractère stochastique de la politique, SAC introduit un terme d'entropie dans les fonctions valeurs de telle façon à avoir :

$$Q_\theta(s,a) = \mathbb{E}_{\pi_\phi}\left[R_t^\gamma + \alpha H_{\pi_\phi}(\cdot|s)) \middle| S_t = s, A_t = a\right], \tag{7}$$

avec $H_{\pi_\phi}(\cdot|s) = -\int \log(\pi_\phi(a|s))\pi_\phi(a|s)\nu(da)$ l'entropie de Shannon de la politique.
Nous montrons alors que :

$$V_\psi(s_t) = \mathbb{E}_{a_t \sim \pi_\phi}\left[Q_\theta(s_t, a_t) - log(\pi_\phi(s_t, a_t))\right], \tag{8}$$

$$Q_\theta(s_t, a_t) = R_t + \gamma \mathbb{E}_{s_{t+1} \sim P}\left[V_\psi(s_{t+1})\right]. \tag{9}$$

Ainsi, nous devons entraîner nos réseaux à minimiser les termes d'erreur suivants :

1. $J_V(\psi) = \mathbb{E}_{s_t \sim \mathcal{D}}\left[\frac{1}{2}\left(V_\psi(s_t) - \mathbb{E}_{a_t \sim \pi_\phi}\left[Q_\theta(s_t, a_t) - \log \pi_\phi(a_t|s_t)\right]\right)^2\right]$.
2. $J_Q(\theta) = \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(s_t, a_t) - R_t - \gamma \mathbb{E}_{s_{t+1} \sim P}\left[V_\psi(s_{t+1})\right]\right)^2\right]$.
3. $J_\pi(\phi) = \mathbb{E}_{s_t \sim \mathcal{D}}\left[D_{KL}\left(\pi_\phi(\cdot|s_t) \middle\| \exp\left(Q_\theta(s_t, \cdot)\right)\right)\right]$.

Où $\mathcal{D}$ est une distribution d'états et d'actions observés (replay buffer) et D_{KL} est la mesure de dissimilarité de Kullback-Leibler.

Les deux premières équations découlent directement des résultats précédents. La troisième équation correspond à une projection de la politique optimale sur la famille de politique paramétrée par ϕ (généralement gaussienne).

Nous exploitons alors le gradient des trois fonctions de perte pour mettre à jour les paramètres du réseau de neurones via un algorithme de gradient stochastique.

1. $\hat{\nabla}_\psi J_V(\psi) = \nabla_\psi V_\psi(s_t) \left(V_\psi(s_t) - Q_\theta(s_t, a_t) + \log(\pi_\phi(a_t|s_t)) \right)$.
2. $\hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(a_t, s_t) \left[-r(s_t, a_t) - \gamma V_\psi(s_{t+1}) + Q_\theta(a_t, s_t) \right]$.
3. $\hat{\nabla}_\phi J_\pi(\phi) = \nabla_\phi \log \pi_\phi(a_t|s_t) + \nabla_\phi f_\phi(\varepsilon_t, s_t) \left(\nabla_{a_t} \log \pi_\phi(a_t|s_t) - \nabla_{a_t} Q_\theta(s_t, a_t) \right)$.

Le calcul des deux premiers gradients est immédiat. Pour la dernière équation, nous utilisons une reparamétrisation en considérant $a_t = f_\phi(\varepsilon_t, s_t)$ comme étant une fonction d'un bruit gaussien ε_t.

Nous obtenons ainsi l'algorithme SAC simplifié (sans réseau cible et double Q-learning) décrit dans [1].

Algorithm 1 Soft Actor-Critic (SAC)

Initialiser les paramètres du réseau θ, ϕ, et ψ
Initialiser la mémoire de replay $\mathcal{D}$
Pour chaque épisode **faire**
    Pour chaque étape de temps t **faire**
        Sélectionner une action $a_t \sim \pi_\phi(.|s_t)$
        Observer la récompense r_t et le nouvel état s_{t+1}
        Stocker (s_t, a_t, r_t, s_{t+1}) dans $\mathcal{D}$
        Échantillonner un mini-batch $\{(s_i, a_i, r_i, s_{i+1})\}$ de $\mathcal{D}$
        Mettre à jour Q_θ en minimisant l'erreur quadratique :

$$J_Q(\theta) = \mathbb{E}_{s,a,r,s'} \left[(Q_\theta(s, a) - (r + \gamma V_\psi(s')))^2 \right]$$

        Mettre à jour V_ψ en minimisant :

$$J_V(\psi) = \mathbb{E}_s \left[\frac{1}{2} (V_\psi(s) - (Q_\theta(s, a) - \log \pi_\phi(a|s)))^2 \right]$$

        Mettre à jour π_ϕ en minimisant :

$$J_\pi(\phi) = \mathbb{E}_{s \sim \mathcal{D}} \left[D_{KL}(\pi_\phi(\cdot|s) || \exp(Q_\theta(s, \cdot)/\tau)) \right]$$

    Fin Pour
Fin Pour

5 Environnement de Market Making

5.1 Choix des paramètres

En market making, un ordre est adverse si son exécution est suivie d'un mouvement de prix défavorable (souvent lié à une asymétrie d'information), entraînant une perte potentielle. À l'inverse,

une exécution non adverse ne subit pas ce biais directionnel. Dans nos simulations, un ordre est exécuté de manière adverse dès que le prix évolue défavorablement. Sinon, il a une probabilité $p = 0.2$ d'être exécuté de façon non adverse.

Nous choisissons de modéliser deux configurations : l'une tenant compte à la fois des exécutions adverses et non adverses (environnement adverse), et l'autre ne considérant que les exécutions non adverses (environnement non adverse).

Nous fixons un pas temporel $dt = 0.001$ (s), un horizon $T = 1$ (s), la contrainte d'inventaire $q = 5$, le mid-price initial $P_0 = 50$, les paramètres de volatilité $\sigma = \tilde{\sigma} = \xi = 0.1$ et un drift $\eta = 0$. Enfin, nous prenons un spread constant $\Delta = 0.01$, et une pénalité d'inventaire $\alpha = 0.001$.

5.2 Discrétisation de l'espace d'actions

Le SAC est compatible avec un espace d'actions continu, tandis que notre modèle impose des actions discrètes dans l'ensemble $\{-1, 0, 1\}$. Nous proposons donc un mapping de l'espace continu vers l'espace discret via un seuillage adapté.

Pour ce faire, nous appliquons la fonction tangente hyperbolique a notre politique. Cette dernière est gaussienne définie sur $\mathbb{R}$ et la fonction appliquée est une bijection croissante de $\mathbb{R}$ sur $[-1, 1]$. Afin d'assurer un mapping uniforme et une équipartition de la probabilité image sur $[-1, 1]$, nous définissons les seuils q_1 et q_2 comme étant l'image des quantiles d'ordre 0.33 et 0.66 d'une gaussienne centrée réduite par la fonction tangente hyperbolique. Après calcul, $q_2 = -q_1 = 0.41$.

Concrètement, si nous tirons une action a dans $[-1, 1]$, alors :

- Si $a \in [-1, q_1] \Rightarrow a = -1$,
- Si $a \in [q_1, q_2] \Rightarrow a = 0$,
- Si $a \in [q_2, 1] \Rightarrow a = 1$.

5.3 Mise à jour du mid-price

La variation du mid-price P dépend uniquement de sa valeur à l'instant précédent car nous négligeons notre market impact. En pratique, nous utilisons un schéma d'Euler-Maruyama pour le mettre à jour.

$$P_{t+dt} = P_t + \sqrt{\sigma^2 + \tilde{\sigma}^2 + \xi^2} \cdot \sqrt{dt} \cdot Z, \quad Z \sim \mathcal{N}(0, 1).$$

Remarque 2. Afin d'améliorer l'apprentissage, nous normalisons nos données en calculant un z-score pour le mid-price et en appliquant une normalisation min-max à l'inventaire :

$$\tilde{P}_t = \frac{P_t - P_0}{\sigma_{norm}}, \quad \tilde{Q}_t{}^A = \frac{Q_t^A + q}{2q}.$$

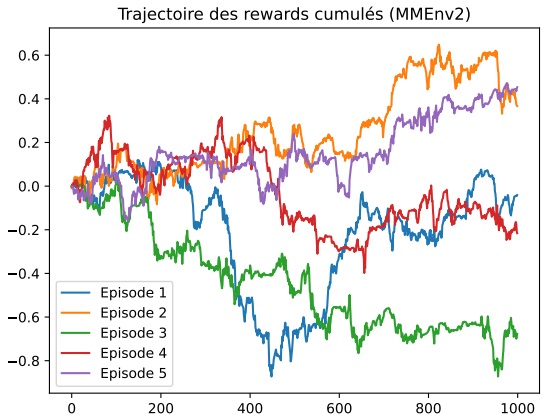
où $q = 5$, $P_0 = 50$, et $\sigma_{norm} = 0.124$.

6 Simulation de l'environnement

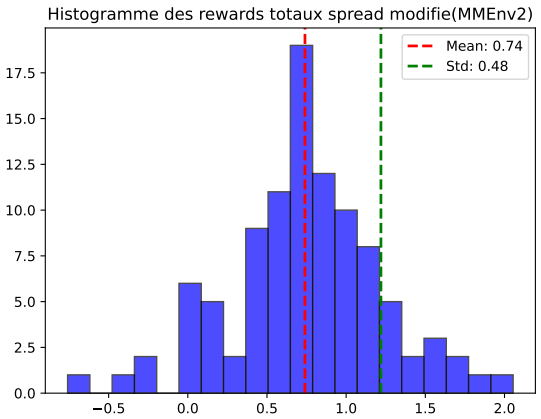
6.1 Principe et résultats préliminaires

Pour rappel, nous travaillons avec deux configurations comme dans [3]. Dans la première, nous prenons en compte les exécution adverses et non adverses. Dans la deuxième, nous prenons en compte uniquement les exécutions non adverses.

Afin de tester notre environnement et vérifier son bon fonctionnement, nous simulons 100 épisodes en appliquant une politique entièrement aléatoire, ce qui permet d'obtenir une première intuition sur son comportement. Nous conservons les paramètres définis dans le protocole expérimental et sélectionnons 5 épisodes parmi les 100 simulés afin de visualiser l'évolution des récompenses cumulés. Enfin, nous représentons la distribution des récompenses totales à l'aide d'histogrammes. Les figures 1 et 2 illustrent ces résultats préliminaires.

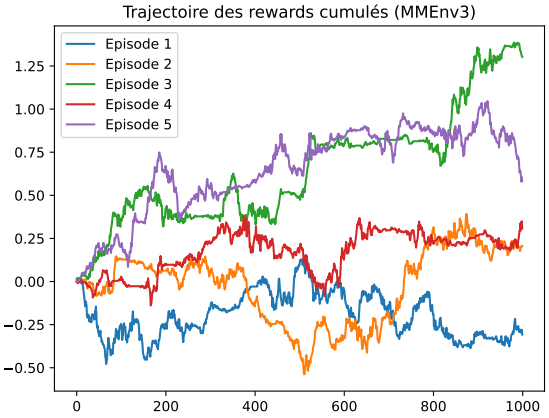


(a) Trajectoire des récompenses cumulées pour les cinq premiers épisodes simulés

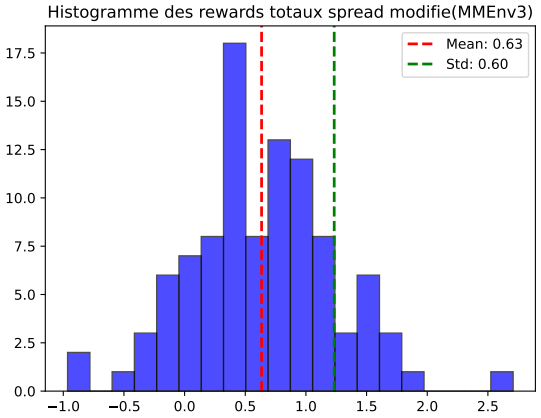


(b) Distribution des récompenses totales pour les 100 trajectoires simulées

FIGURE 1 – Récompenses cumulées et totales pour des simulations de l'environnement avec adversité



(a) Trajectoire des récompenses cumulées pour les cinq premiers épisodes simulés



(b) Distribution des récompenses totales pour les 100 trajectoires simulées

FIGURE 2 – Récompenses cumulées et totales pour des simulations de l'environnement sans adversité

Commentaire :

Nous remarquons que la distribution des récompenses finales est plus avantageuse dans le cas où l'adversité est prise en compte, avec une moyenne de 0.74 contre 0.63 pour le cas non adverse. Nos résultats divergent de ceux de [3], qui observent quant à eux un effet nettement pénalisant de l'adversité dans l'apprentissage.

À première vue, notre constat peut sembler contre-intuitif. Cependant, une explication réside dans le fait que le mid-price suit un mouvement brownien, ce qui implique que le nombre d'exécutions adverses est environ cinq fois supérieur au nombre d'exécutions non adverses, compte tenu de nos paramètres. Or, si le spread est suffisamment grand par rapport à la volatilité du mid-price (voir la prochaine sous-section), le market maker ne subit pas seulement aucune perte, mais réalise également un gain net. Certes, une exécution non adverse reste plus rentable individuellement qu'une exécution adverse, mais cette rentabilité est compensée par leur faible fréquence.

De plus, la pénalité d'inventaire, bornée par $5 \cdot 10^{-3}$, est négligeable et n'a quasiment aucun impact sur la richesse finale du market maker, qui est 100 fois supérieure en ordre de grandeur. Ce choix de paramètre est d'autant plus surprenant que les auteurs de [3] lui attribuent un rôle significatif dans l'apprentissage avec SAC, entraînant des inventaires proches de 1, une situation que nous n'observons jamais dans notre cadre expérimental.

Enfin, en prenant du recul sur l'environnement, nous pouvons commencer à entrevoir certaines limitations, suggérant que l'apprentissage risque de n'avoir qu'un intérêt limité. En effet, non seulement l'espace d'actions est très restreint, mais le mid-price est surtout totalement imprévisible, empêchant le modèle d'exploiter des patterns pertinents.

6.2 Lien entre exécution adverse et les paramètres du modèle

Pour illustrer nos propos, nous allons exprimer la récompense du market maker quand une exécution adverse a lieu. Afin de fixer les idées, disons que nous nous plaçons dans le cas où il a envoyé un ordre au best ask à l'instant t et il a été exécuté à l'instant $t + dt$. Le mid-price a donc augmenté ($P_{t+dt} > P_t$). En négligeant la pénalité d'inventaire (déjà négligeable en soi) et en prenant $Q_t^A = 0$, la récompense s'écrit :

$$R_{t+dt} = W_{t+dt} - W_t = P_{t+dt} \cdot Q_{t+dt}^A + C_{t+dt} - P_t \cdot Q_t^A - C_t,$$

$$R_{t+dt} = \frac{\Delta}{2} - |\Delta P_t|,$$

Ainsi,

$$\mathbb{E}(R_{t+dt}) = \frac{\Delta}{2} - \mathbb{E}(|\Delta P|_t),$$

$(P_t)_t$ un processus de diffusion avec drift nul. Ainsi, $|\Delta P_t| \sim \sigma_{eff}\sqrt{dt}|Z|$ où σ_{eff} est la volatilité effective de P.

$$\text{`}\mathbb{E}(R_{t+dt}) = \frac{\Delta}{2} - \mathbb{E}(|\Delta P|_t),$$

$$\mathbb{E}(R_{t+dt}) = \frac{\Delta}{2} - \sigma_{eff}\sqrt{\frac{2}{\pi}dt},$$

Donc,

$$\mathbb{E}(R_{t+dt}) > 0 \Leftrightarrow \frac{\Delta}{2} > \sigma_{eff}\sqrt{\frac{2}{\pi}dt} \Leftrightarrow \frac{\Delta}{\sigma_{\text{eff}}} > 2\sqrt{\frac{2}{\pi}dt}.$$

Nous pouvons ainsi voir que le ratio volatilité/spread est une grandeur clé dans l'interprétation de l'impact de l'adversité sur la richesse du market maker. Nous notons également que l'inégalité ci-dessus est largement vérifiée avec nos valeurs de paramètres. Il est donc cohérent que les exécutions adverses soient bénéfiques.

7 Entraînement et test du SAC

Dans cette partie, nous présentons l'architecture neuronale de notre implémentation de SAC [3], ainsi que les résultats d'entraînement et de test pour les deux configurations. Dans un premier temps, nous entraînons notre modèle avec les paramètres initiaux, puis nous réduisons la valeur du spread et augmentons la pénalité sur l'inventaire afin d'étudier les performances du modèle dans un environnement de market making plus agressif.

7.1 Architecture SAC

- Actor Neural Network : Étant donné notre espace d'états, le réseau de neurones est conçu pour prédire la moyenne et le logarithme de l'écart-type d'une politique gaussienne. La couche d'entrée du réseau correspond au vecteur d'état $s \in \mathbb{R}^2$. Il comporte deux couches cachées, chacune composée de 256 neurones. La première couche cachée est définie par :

  $$h_1 = \text{ReLU}(W_1 s + b_1),$$

  et la seconde par :

  $$h_2 = \text{ReLU}(W_2 h_1 + b_2),$$

  où W_1 et W_2 sont les matrices de poids du réseau de neurones, et b_1 et b_2 sont les biais associés. L'activation utilisée, ReLU (Rectified Linear Unit), est un choix populaire pour éviter le problème des gradients évanescents. La couche de sortie se compose de deux parties : l'une pour la moyenne et l'autre pour le logarithme de l'écart-type de la distribution des actions. La sortie de la moyenne est définie par $\mu = W_\mu h_2 + b_\mu$, tandis que la sortie du logarithme de l'écart-type est donnée par $\log \sigma = W_{\log(\sigma)} h_2 + b_{\log(\sigma)}$.
- Critic Neural Network : Pour estimer les fonctions de valeur état-action, la méthode SAC utilise deux réseaux de neurones critiques, Q_1 et Q_2, afin de réduire le biais de surestimation souvent observé dans les méthodes à critique unique. Chaque réseau critique possède une architecture presque identique à celle du réseau acteur décrit précédemment. L'entrée de ces réseaux est la paire état-action $[s, a] \in \mathbb{R}^3$. Les couches cachées sont définies de la même manière que pour le réseau acteur, soit :

  $$h_1 = \text{ReLU}(W_1[s, a] + b_1) \quad \text{et} \quad h_2 = \text{ReLU}(W_2 h_1 + b_2).$$

  Enfin, la couche de sortie, représentant les Q-valeurs d'état-action $Q(s, a)$, est définie par :

  $$Q(s, a) = W_Q h_2 + b_Q.$$

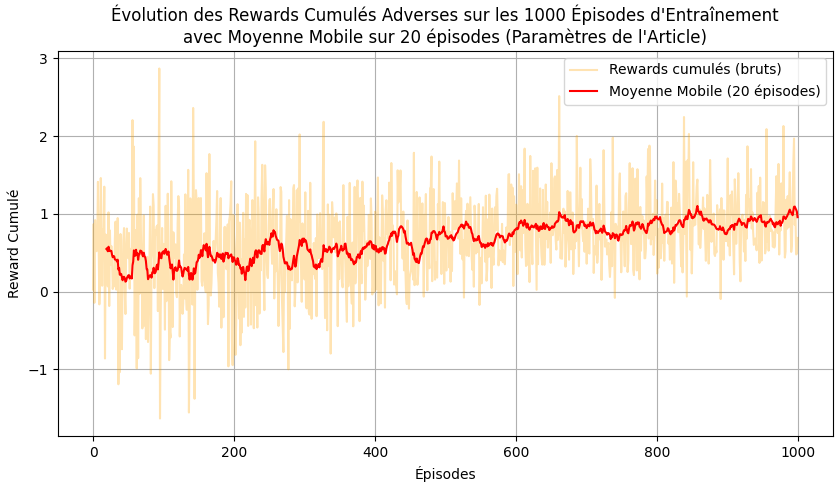
- Paramètres des réseau de neurones : Pour extraire des représentations optimisées des états avant leur traitement par les réseaux acteur et critique, un réseau de neurones presque identique est utilisé ici. L'entrée de ce réseau est le vecteur d'état s, les couches cachées sont définies par :

  $$h_1 = \text{ReLU}(W_{h_1} s + b_{h_1}) \quad \text{et} \quad h_2 = \text{ReLU}(W_{h_2} h_1 + b_{h_2}).$$

7.2 Résultats numériques

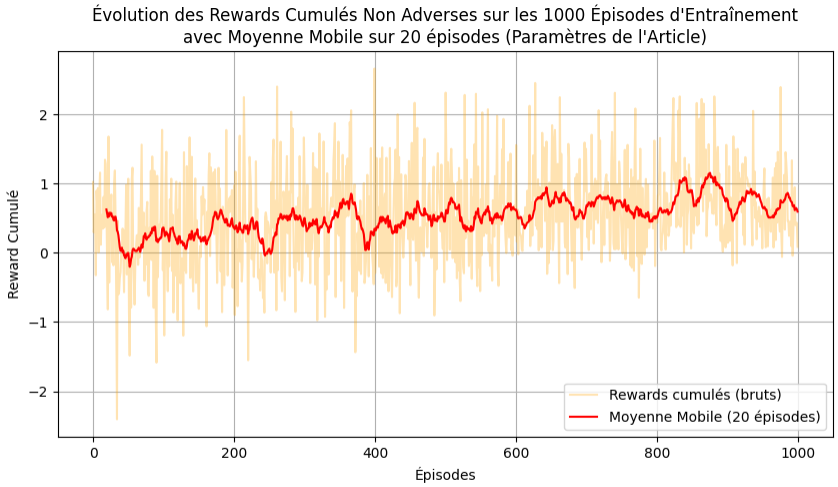
7.2.1 Entraînement

Nous utilisons initialement un spread de 0.01 et une pénalité d'inventaire de 0.001. Les figures 3 et 4 montrent respectivement l'évolution des récompenses cumulées pour les cas adverse (inclus les deux types d'exécution) et non adverse sur 1000 épisodes d'entraînement.



Moyenne: 0.661 | Écart-type: 0.569

FIGURE 3 – Trajectoire des récompenses cumulées pour le cas adverse sur les données d'entraînement



Moyenne: 0.526 | Écart-type: 0.678

FIGURE 4 – Trajectoire des récompenses cumulées pour le cas non adverse sur les données d'entraînement

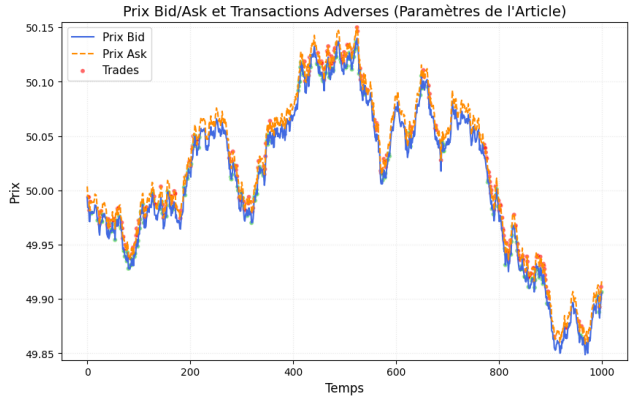
Commentaire :

L'analyse des courbes d'entraînement met en évidence une amélioration progressive de la prise de décision de l'agent, confirmant ainsi l'apprentissage de notre algorithme SAC. Cette progression se traduit par une tendance croissante de la moyenne mobile sur 20 épisodes. Par ailleurs, et comme

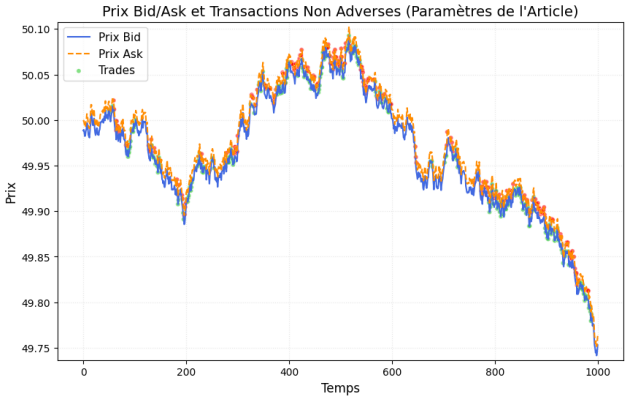
nous en avions longuement discuté, nous avons encore de meilleurs résultats avec la configuration adverse.

7.2.2 Test

Pour tester notre modèle, nous simulons 200 épisodes et analysons la distribution des récompenses cumulées à la fin de ces épisodes (figure 8). De plus, pour un épisode spécifique, nous représentons l'évolution du best bid/ask et la position des trades (figure 5), la trajectoire des récompenses cumulées (figure 6) ainsi que l'évolution de l'inventaire (figure 7) dans les deux configurations.

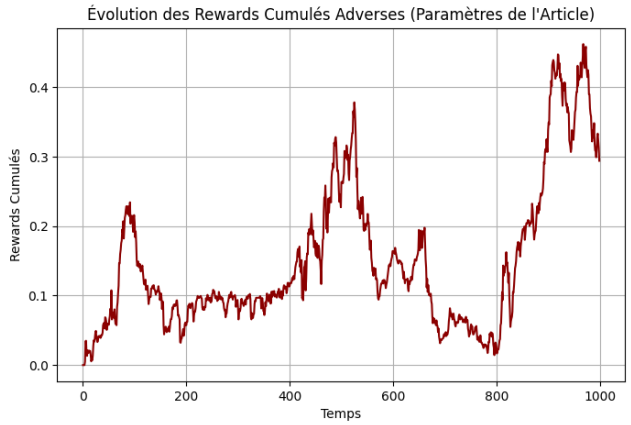


(a) Trajectoire du best bid/ask et position des trades pour un seul épisode des données de test pour l'environnement adverse

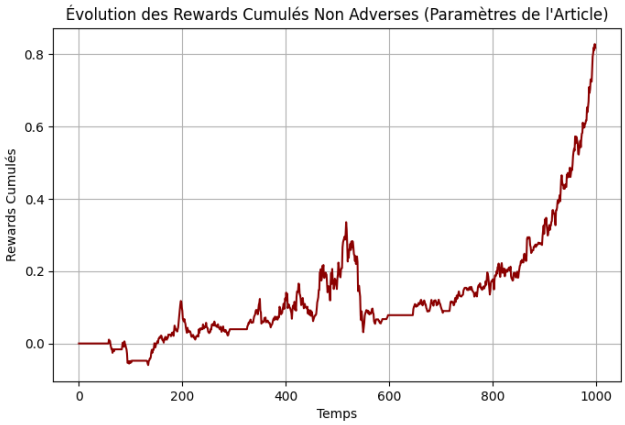


(b) Trajectoire du best bid/ask et position des trades pour un seul épisode des données de test pour l'environnement non adverse

FIGURE 5 – Trajectoire du best bid/ask et position des trades pour un seul épisode des données de test

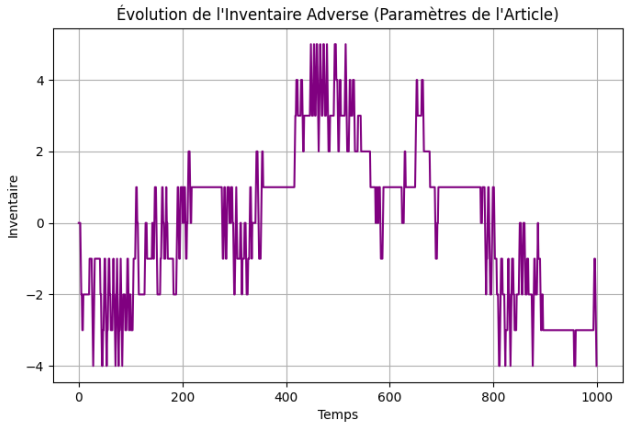


(a) Trajectoire des récompenses cumulées pour un seul épisode des données de test pour l'environnement adverse

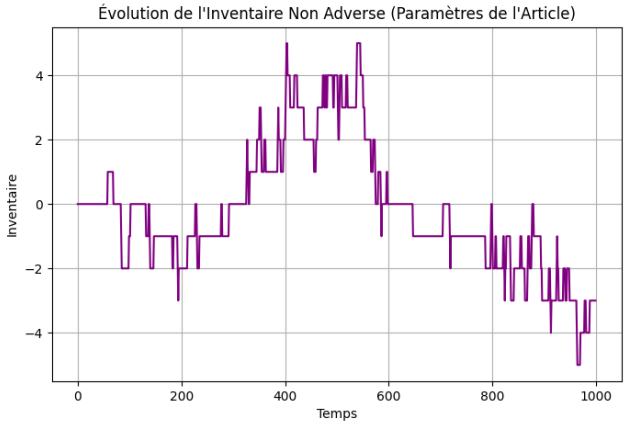


(b) Trajectoire des récompenses cumulées pour un seul épisode des données de test pour l'environnement non adverse

FIGURE 6 – Trajectoire des récompenses cumulées pour un seul épisode des données de test

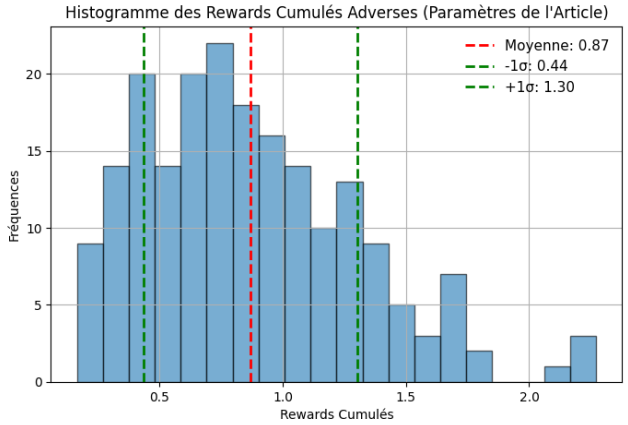


(a) évolution de l'inventaire pour un seul épisode des données de test pour le cas adverse

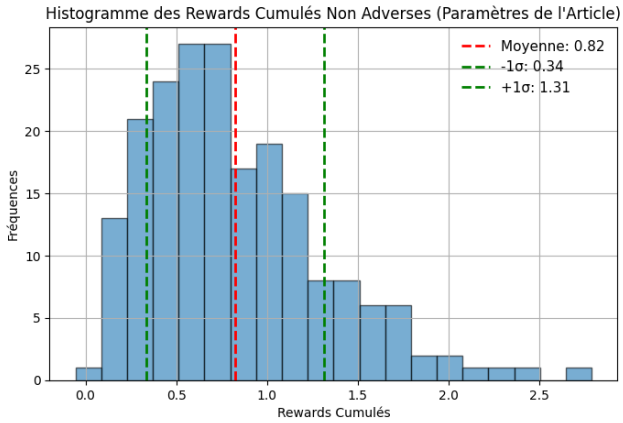


(b) évolution de l'inventaire pour un seul épisode des données de test pour le cas non adverse

FIGURE 7 – évolution de l'inventaire pour un seul épisode des données de test



(a) Histogramme des récompenses cumulées pour le cas adverse sur les données de test



(b) Histogramme des récompenses cumulées pour le cas non adverse sur les données de test

FIGURE 8 – Histogramme des récompenses cumulées sur les données de test

Commentaire :

L'analyse des courbes nous permet de mieux comprendre la stratégie apprise par SAC. L'algorithme tente de détecter des tendances dans l'évolution du mid-price afin de les exploiter. Concrètement, il vend à découvert lorsque le prix chute sur plusieurs pas de temps et achète lorsque celui-ci monte, tout en respectant la contrainte d'inventaire.

Ce comportement est gagnant si la dynamique du prix se poursuit dans la même direction, mais perdant en cas de retournement. Or, puisque le mid-price suit un mouvement brownien, chaque mouvement a une chance sur deux de se prolonger ou de s'inverser. Par conséquent, le modèle sur-apprend.

Toutefois, comme nous l'avons remarqué précédemment, le modèle dispose de peu de leviers pour optimiser ses performances. En l'absence d'autres signaux exploitables, la seule stratégie rationnelle qu'il peut adopter consiste à augmenter la fréquence de ses ordres, ce qu'il fait. Par conséquent, nous observons bien des résultats légèrement meilleurs que dans le cas aléatoire.

7.3 Effet du spread et de la pénalisation sur l'inventaire

À présent, nous souhaitons tester notre modèle avec des paramètres ajustés afin d'accentuer la pénalisation de l'inventaire et de rendre les ordres adverses perdants en moyenne, dans le but d'observer l'impact de ces contraintes sur la politique apprise.

7.3.1 Entraînement

Pour étudier l'effet d'une diminution du spread et d'une augmentation de la pénalité nous choisissons un spread égal a 0.005 et une pénalité sur l'inventaire égal à 0.1. Nous visualisons dans les figures 9 et 10 l'évolution des récompenses cumulées pour les deux cas adverse et non adverse sur les 1000 épisodes d'entraînement.

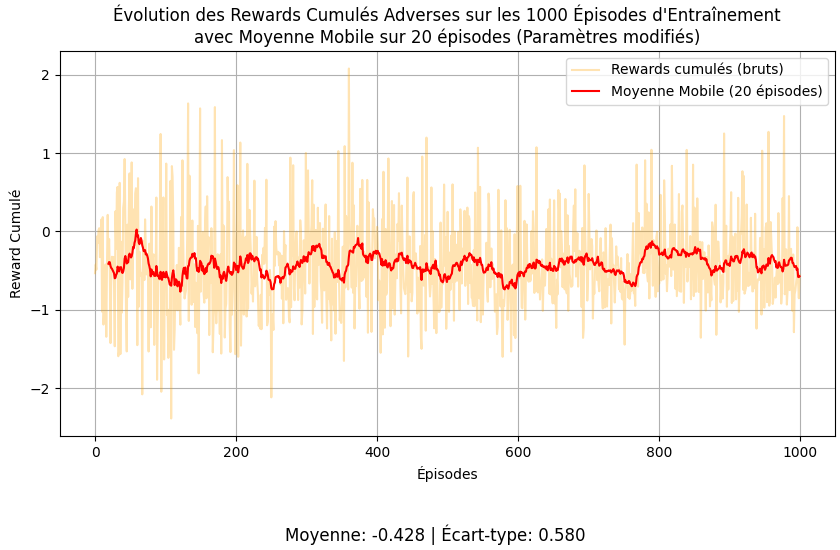


FIGURE 9 – Trajectoire des récompenses cumulées pour le cas adverse sur les données d'entraînement

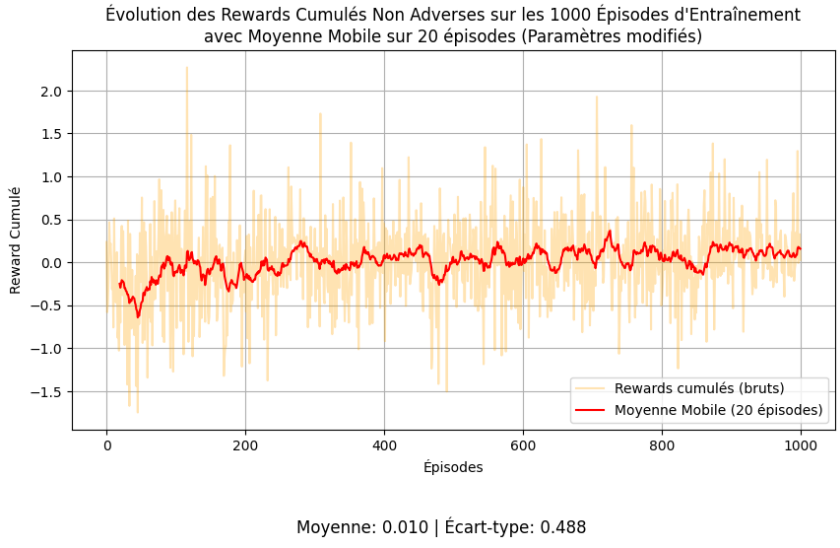


FIGURE 10 – Trajectoire des récompenses cumulées pour le cas non adverse sur les données d'entraînement

Commentaire :

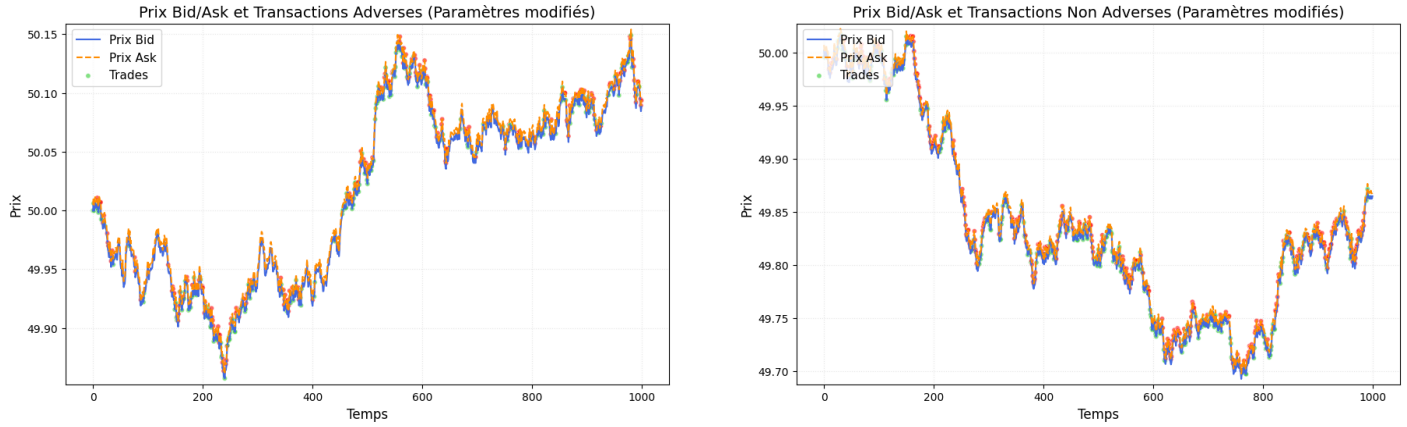
Nous observons sur les courbes d'apprentissage que les récompenses restent négatives tout au long de l'entraînement dans le cas adverse. Autrement dit, il vaudrait mieux ne rien faire plutôt que d'utiliser cette politique. Cela est logique : désormais, les exécutions adverses sont déficitaires en moyenne, et l'agent ne peut ni prédire l'évolution du mid-price, ni contrôler le type d'exécution qui se produira. Dans ces conditions, aucune stratégie viable ne peut émerger, rendant l'apprentissage inefficace.

Pour le cas non-adverse, qui subit également la réduction du spread mais ne souffre pas du biais directionnel des exécutions adverses, nous parvenons à obtenir des récompenses positives en fin d'entraînement.

7.3.2 Tests

Concernant la partie test, nous représentons les mêmes types de courbes que dans la section précédente.

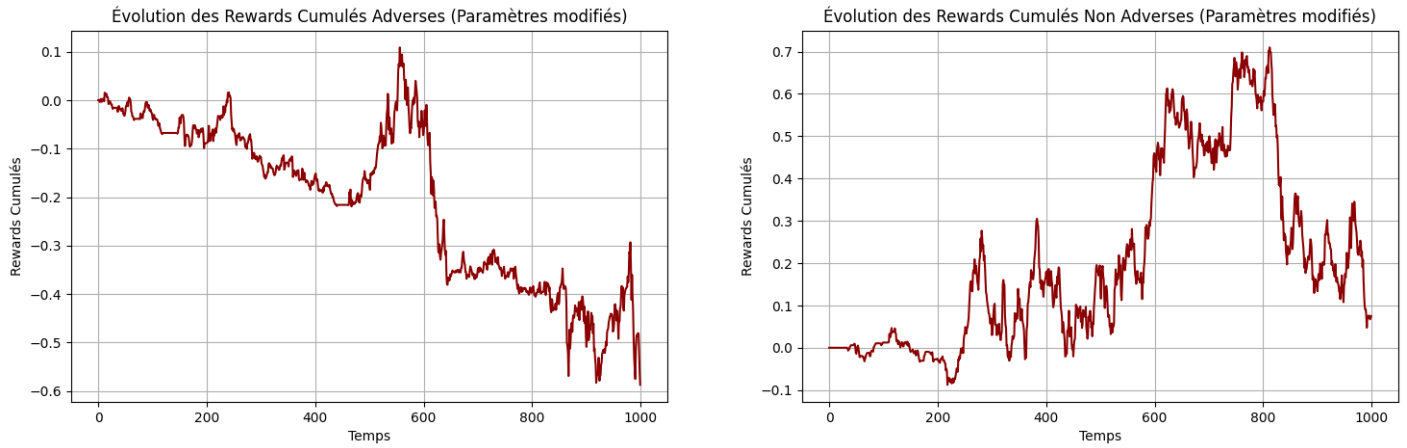
FIGURE 11 – Évolution du best bid/ask et des positions de trade sur les données de test



(a) Environnement adverse

(b) Environnement non adverse

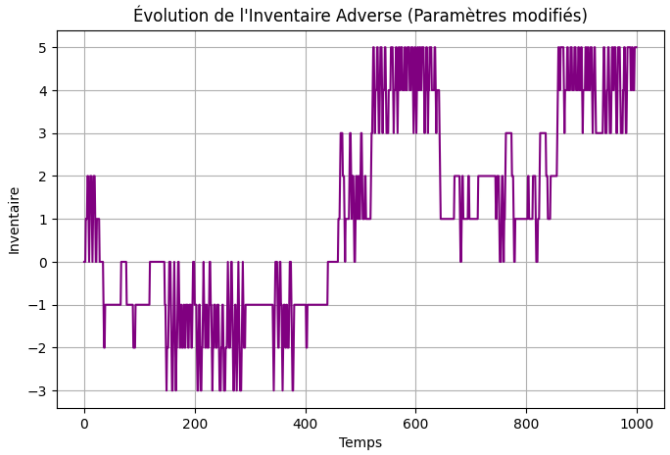
FIGURE 12 – Évolution des récompenses cumulées sur les données de test



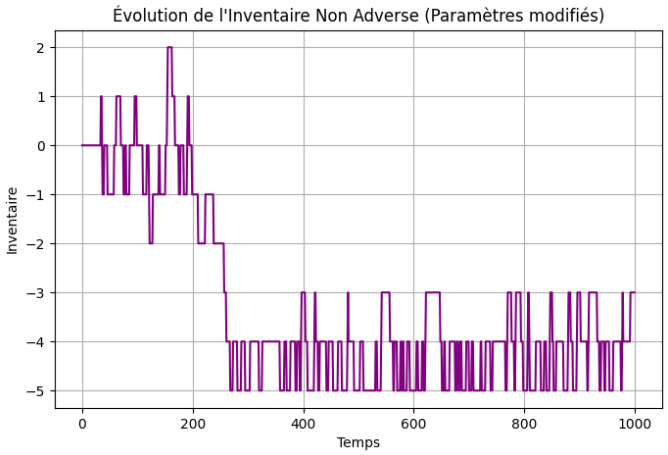
(a) Environnement adverse

(b) Environnement non adverse

FIGURE 13 – Évolution de l'inventaire pour un seul épisode des données de test

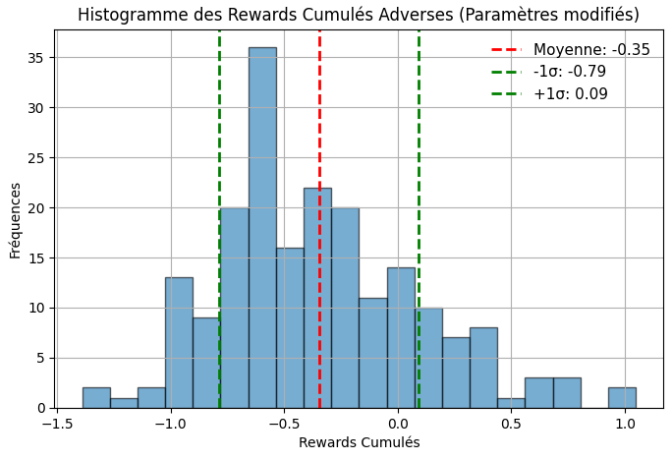


(a) Environnement adverse

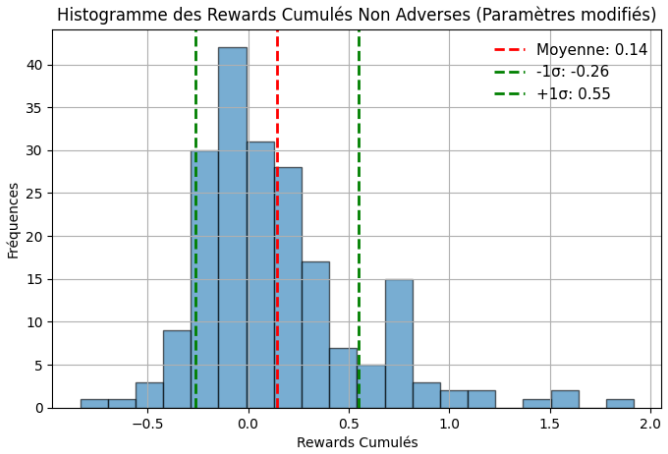


(b) Environnement non adverse

FIGURE 14 – Histogramme des récompenses cumulées sur les données de test



(a) Environnement adverse



(b) Environnement non adverse

Commentaire :

Dans la continuité de nos observations sur l'apprentissage, nous constatons que la configuration adverse produit des résultats inférieurs à ceux du cas non adverse. De plus, l'entraînement s'est révélé totalement inefficace dans le premier cas, tandis qu'il a permis une amélioration notable dans l'autre. Enfin, bien que la pénalité soit très élevée, elle n'a pas empêché les agents de maintenir des inventaires importants, ceux-ci représentant leur seule opportunité de générer des gains.

8 Conclusion

Pour conclure, nous avons étudié le problème du market making optimal en appliquant l'algorithme SAC, une méthode d'apprentissage par renforcement off-policy, particulièrement adaptée aux environnements complexes et à haute dimension. Afin de mieux modéliser la dynamique des carnets d'ordres, nous avons exploré les dynamiques de prix non markoviennes basées sur les processus de Hawkes et semi-Markoviens, tout en intégrant les exécutions adverses et les contraintes d'inventaire afin de mieux refléter les conditions réelles du marché.

Cependant, l'analyse des résultats numériques met en évidence certaines limites de notre modèle. L'hypothèse d'un mid-price suivant une diffusion brownienne et d'un spread constant, l'absence de market impact et la simplification des dynamiques d'exécution des ordres ne capturent pas fidèlement la microstructure des marchés haute fréquence. Ces simplifications réduisent l'efficacité de la politique apprise dans des conditions de marché plus complexes.

Parmi les pistes d'amélioration, une modélisation plus réaliste du carnet d'ordres, inspirée du modèle multiniveau Queue Reactive (Rosenbaum et al.), permettrait de mieux intégrer la profondeur du carnet, la formation des files d'attente et l'impact des ordres sur la dynamique des prix. De plus, diversifier les types d'ordres disponibles (ex. ordres iceberg, ordres stop, ordres conditionnels) offrirait une plus grande flexibilité à la politique apprise, améliorant ainsi les performances globales du market maker.

Bibliographie

[1] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic : Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[2] Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of applied probability*, 11(3) :493–503, 1974.

[3] Luca Lalor and Anatoliy Swishchuk. Algorithmic and high-frequency trading problems for semi-markov and hawkes jump-diffusion models. *arXiv preprint arXiv :2409.12776*, 2024.

[4] Luca Lalor and Anatoliy Swishchuk. Reinforcement learning in non-markov market-making. *arXiv preprint arXiv :2410.14504*, 2024.

[5] Ana Karen Roldan Contreras. Stochastic optimal control problems in lob for various stochastic models. 2023.

[6] Myles Sjogren and Timothy DeLise. General compound hawkes processes for mid-price prediction. *arXiv preprint arXiv :2110.07075*, 2021.

[7] Anatoliy Swishchuk and Aiden Huffman. General compound hawkes processes in limit order books. *Risks*, 8(1) :28, 2020.