

Un modèle adaptatif d'estimation de l'Hamiltonien par machine learning

BELHBOUB Anouar¹, BENDAOU Hamza¹, ZAOUG Imad¹, BENTAHAR Saad¹ and ENNAJAH Ayoub¹

¹ ECOLE CENTRALE CASABLANCA, CASABLANCA, MAROC

Reception date of the manuscript: dd/mm/aaaa

Acceptance date of the manuscript: dd/mm/aaaa

Publication date: dd/mm/aaaa

Abstract— In this research, we present a method for inferring the dynamic parameters of a quantum system that integrates sequential Monte Carlo and Bayesian experimental design. Our algorithm is designed to be efficient by including parameters that balance the demands of computational and experimental resources. It can be implemented during experimental data collection, eliminating the need for data storage and post-processing. Additionally, our algorithm is able to adapt to changes in Hamiltonian parameters and the presence of unknown noise processes from one experiment to another. It also provides a numerical estimate of the Cramer-Rao lower bound, which certifies its own performance..

Keywords—Sequential Monte Carlo and Bayesian Experiment design, tomography, batch processing, post processing, quadratic loss, Cramer-Rao bound, negative Bayes risk, Newton conjugate-gradient, Hamiltonien.

I. INTRODUCTION

L'estimation de l'Hamiltonien d'un système quantique est une tâche cruciale dans la construction de processeurs de traitement de l'information quantique à grande échelle. Elle permet de mieux comprendre les dynamiques du système et d'appliquer des codes de correction d'erreurs et d'autres outils pour mettre en œuvre des algorithmes quantiques utiles. Les méthodes classiques d'estimation de l'Hamiltonien, telles que la tomographie quantique, ont pour objectif d'estimer le processus complet. Etant donné que ces méthodes présentent des limites en termes de précision et de complexité, surtout lorsque l'on traite avec des systèmes à grande échelle, Il peut être plus utile d'estimer des processus partiels ou la dynamique du système. Pour remédier à ces limitations, nous proposons une nouvelle méthode qui combine le Monte Carlo séquentiel et la méthode bayésienne de conception expérimentale. Cette approche permet d'utiliser des informations sur le système pour améliorer la connaissance des paramètres de modèle spécifiques et de concevoir des expériences pour déduire ces paramètres de manière efficace. En utilisant cette méthode, nous nous intéressons à l'estimation de l'Hamiltonien, qui peut être réalisée en paramétrisant l'Hamiltonien sous la forme $H(x_1, \dots, x_d)$ et en estimant le vecteur de paramètres $x = (x_1, \dots, x_d)$. En présence de bruit ces paramètres fluctuent en suivant une loi

de probabilité, et leur estimation revient dans ce cas à estimer des hyperparamètres qui définissent ces lois de probabilité. Pour démontrer l'efficacité de notre approche, nous l'avons appliquée à un simulateur quantique à ions piégés (atomes chargés qui sont maintenus en place par un champ électrique et utilisés comme qubits). En résumé, notre travail vise à développer une méthode efficace pour l'estimation de l'Hamiltonien dans le cadre de la construction de processeurs de traitement de l'information quantique à grande échelle, en utilisant une approche combinant "Monte Carlo séquentiel" et "la conception expérimentale bayésienne".

II. MÉTHODOLOGIE

a. Conception expérimentale bayésienne

La conception expérimentale bayésienne est une approche qui utilise des techniques bayésiennes pour planifier et analyser les expériences. Le principe de base du design expérimental bayésien est de combiner les informations disponibles sur les variables de l'expérience avec les données de l'expérience elle-même pour obtenir des estimations et des prévisions plus précises.

Le processus de conception expérimentale bayésienne se déroule en plusieurs étapes :

1. Définir l'hypothèse H et le paramètre x que nous voulons estimer.
2. Définir le prior $P(x)$, c'est-à-dire la probabilité a priori de l'hypothèse H étant vraie. Le prior peut être défini de manière informative ou non informative, en fonction des connaissances que nous avons sur le paramètre x et de

l'hypothèse que nous faisons sur sa distribution.

3. Choisir les paramètres de contrôle C qui maximisent l'utilité de l'expérience. Les paramètres de contrôle sont des paramètres que nous pouvons ajuster pour influencer les résultats de l'expérience, tels que les conditions de mesure ou les paramètres de l'appareil de mesure.

4. Collecter les données D résultantes des paramètres de contrôle choisis. C et D représentent respectivement l'ensemble des paramètres de contrôle qu'on a choisi et l'ensemble des résultats obtenus jusqu'à l'expérience présente.

5. Calculer la vraisemblance $P(D|x;C)$, c'est-à-dire la probabilité des données D étant observées étant donné que l'hypothèse H est vraie et en utilisant les paramètres de contrôle C .

6. Calculer le posterior $P(x|D;C)$ en utilisant la règle de Bayes :

$$P(x|D) = \frac{P(D|x;C)P(x)}{P(D|C)}$$

. Le posterior est la probabilité a posteriori de l'hypothèse H étant vraie, c'est-à-dire la probabilité de l'hypothèse après avoir observé les données D .

7. Répéter ce processus pour un nombre défini de fois et dans chaque itération l'algorithme apprend des nouvelles informations sur le paramètre x et oriente le choix des nouvelles paramètres de contrôle c_i pour la prochaine expérience.

8. Interpréter les résultats et tirer des conclusions sur le paramètre x en se basant sur le posterior.

En utilisant une approche bayésienne de design expérimental, nous pouvons non seulement obtenir des estimations et des prévisions plus précises en combinant les informations disponibles avec les données de l'expérience, mais nous pouvons également prendre en compte l'incertitude et l'imprécision dans nos estimations et mettre à jour nos connaissances en fonction de nouvelles données.

De plus, l'approche bayésienne nous permet de comparer différentes hypothèses et de choisir celle qui convient le mieux aux données observées, ce qui nous permet de faire des prévisions plus précises et de mieux comprendre les phénomènes que nous étudions.

Le design expérimental bayésien est souvent utilisé en combinaison avec d'autres techniques statistiques pour obtenir des estimations et des prévisions précises et robustes. Cette approche est particulièrement utile dans les situations où les données sont limitées ou incomplètes, ou lorsque nous voulons tenir compte de l'incertitude et de l'imprécision dans nos estimations.

b. Fonction d'utilité

La fonction d'utilité est un outil utilisé en conception expérimentale bayésienne pour déterminer les paramètres de contrôle optimaux d'une expérience. Elle définit les objectifs de l'expérience et permet de quantifier l'utilité ou le gain attendu en fonction de différents settings de contrôle.

La fonction d'utilité peut être définie de différentes manières en fonction des objectifs de l'expérience. Elle peut être basée sur la précision de l'estimation du paramètre inconnu, sur la vitesse à laquelle l'expérience est menée ou sur tout autre critère qui est considéré comme important pour l'expérience.

Pour utiliser la fonction d'utilité en conception expérimentale bayésienne, nous devons d'abord définir les paramètres de contrôle de l'expérience et la fonction d'utilité associée. Ensuite, nous devons optimiser la fonction d'utilité en fonction des paramètres de contrôle pour trouver les settings de contrôle qui maximisent l'utilité de l'expérience. Cela peut être fait en utilisant des techniques d'optimisation, telles que la recherche de gradient ou la méthode de Nelder-Mead.

Une fois que les paramètres de contrôle optimaux ont été trouvés, nous pouvons utiliser ces paramètres pour mener l'expérience et obtenir des résultats qui sont aussi précis et utiles que possible. La fonction d'utilité joue donc un rôle crucial dans la détermination de la manière de mener l'expérience et dans la décision de quels paramètres de contrôle sont les plus importants pour atteindre nos objectifs.

La quantification de l'utilité du choix des paramètres de contrôle pour la prochaine expérience sachant qu'on a obtenu un certain ensemble de résultats et on a appliqué un certain ensemble de paramètres de contrôle C se fait suivant la relation suivante:

$$U(c_{N+1}) = \mathbb{E}dN + 1 | D; c_{N+1}, C[U(d_{N+1}; c_{N+1})]$$

. c'est un moyennage sur tous les résultats possibles si on impose le paramètre de contrôle en prenant en considération les résultats et les paramètres de contrôle des anciennes expériences.

c. hyperparametres

Les hyperparamètres sont des paramètres qui décrivent la distribution des paramètres d'un modèle plutôt que les paramètres du modèle lui-même. Ils permettent de généraliser le problème de l'apprentissage des paramètres du modèle à celui de l'apprentissage des paramètres qui décrivent la distribution de ces paramètres. Par exemple, supposons que nous voulons mesurer l'énergie d'un système quantique en utilisant une série d'expériences différentes. Si les paramètres de l'Hamiltonien du système varient d'une expérience à l'autre en raison d'erreurs expérimentales, il peut être difficile de déterminer l'énergie exacte du système en utilisant les données de chaque expérience séparément. Au lieu de cela, nous pouvons utiliser des hyperparamètres pour décrire la distribution des paramètres de l'Hamiltonien dans chaque expérience. Nous pouvons alors utiliser des techniques pour estimer les valeurs de ces hyperparamètres et, par extension, estimer la distribution des paramètres de l'Hamiltonien dans chaque expérience. En utilisant cette approche, nous pouvons prendre en compte la variation des paramètres d'une expérience à l'autre et obtenir une estimation plus précise de l'énergie du système quantique.

Pour chaque expérience, nous mesurons les données D_i et les paramètres de l'Hamiltonien x_i . Nous utilisons les équations de Bayesian inference pour estimer les hyperparamètres y à partir des données D_i et des paramètres de l'Hamiltonien x_i :

$$P(y|D_i) = \frac{P(D_i|y)P(y)}{P(D_i)}$$

$$\mathbb{E}_{x,y}[x] = \mathbb{E}_y[\mathbb{E}_x[y|x]]$$

$$Cov_{x,y}(x) = \mathbb{E}_y[Cov_x(y|x)] + Cov_y(\mathbb{E}_x[y|x])$$

Nous utilisons la valeur attendue des paramètres de l'Hamiltonien $\mathbb{E}_{x,y}[x]$ comme estimation de l'énergie du système quantique et la covariance $Cov_{x,y}(x)$ comme estimation de l'incertitude sur cette estimation. Cette approche nous permet de prendre en compte la variation des paramètres de l'Hamiltonien d'une expérience à l'autre en utilisant les hyperparamètres y . Cela nous permet d'obtenir une estimation plus précise de l'énergie du système quantique et de mieux comprendre l'incertitude sur cette estimation..

d. Monte Carlo séquentielle

Monte Carlo séquentielle (aussi appelé le filtre de particules) est un algorithme utilisé en inférence bayésienne pour estimer la distribution de probabilité d'un paramètre inconnu à partir de données observées. Il est particulièrement utile dans les situations où la distribution de probabilité du paramètre est complexe et difficile à modéliser de manière analytique.

Il consiste dans un premier temps de définir un nombre de particule $(x_i)_{i \in \{1, \dots, n\}}$ d'une manière aléatoire ou en se basant sur la probabilité initiale (prior) et qu'on peut les définir comme un ensemble des états possible pour x et à chaque particule on attribue un poids (ω_i) qui signifie la probabilité que la particule (x_i) soit la particule recherchée, on définit l'ensemble des particules par :

$$\chi = \{x^{[j]}, \omega^{[j]}\}$$

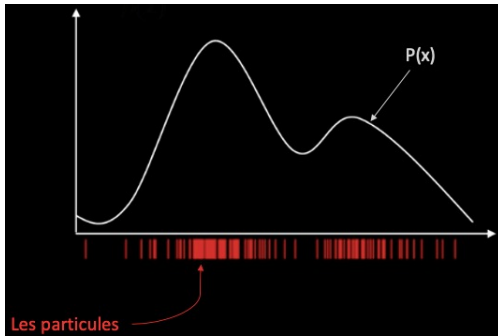


Fig. 1: Extraction des particule

Et on peut écrire le posterior par la relation suivante :

$$Pr(x|D) = \sum_{k=0}^n \omega_k \cdot \delta_k(x - x_k)$$

le poids des particules est toujours uniforme avant chaque itération, cependant, les poids à la fin de chaque itération se diffèrent d'une particule à une autre en se basant sur les résultats de l'expérience et ces particules sont mises à jour.

Le processus de mise à jour des particules se déroule en plusieurs étapes :

1. Propagation : chaque particule est propagée à l'étape suivante en utilisant un modèle de dynamique qui décrit comment le paramètre inconnu évolue dans le temps.

2. Mise à jour du poids : la vraisemblance de chaque particule est mise à jour en fonction des données observées et de l'hypothèse sur laquelle elle est basée, ce qui engendre une mise à jour des poids de chaque particule suivant la loi suivante :

$$\bar{\omega}_i = \omega_i \cdot P(D|x; C)$$

. On normalise les $\bar{\omega}_i$ pour que la somme des nouveaux poids soit égale à 1.

3. Rééchantillonnage : Tous d'abord, on filtre les particules de leur poids, de manière à ce que les particules les plus probables soient plus fréquemment sélectionnées et que celles qui sont moins probables le soient moins souvent, et pour cela on définit un seuil qu'on va le nommer *approx,atio*.

Le rééchantillonnage risque de tomber dans le problème des poids nuls. Donc, on génère des nouvelles particules en ajoutant du bruit aux particules restantes si le nombre de ces particules est inférieur à un seuil qu'on définit afin de maintenir une certaine diversité dans la distribution de probabilité estimée.

Le problème d'un poids nul peut avoir de graves conséquences sur la performance du filtre de particules et sur la précision de l'estimation finale. En effet, ces particules ayant un poids nul ne seront pas prises en compte lors du rééchantillonnage, ce qui peut entraîner une perte d'informations importante. De plus, ces particules n'auront aucun impact sur l'estimation finale, ce qui peut conduire à des estimations biaisées ou inexactes.

Le filtre de particules permet de converger vers une estimation de la distribution de probabilité du paramètre inconnu à mesure que de nouvelles données sont observées.

e. Le design expérimental bayésien séquentiel

Le design expérimental bayésien séquentiel (Sequential Monte Carlo Bayesian experimental design, SMC-BED) est une approche pour optimiser la conception d'expériences en utilisant l'inférence bayésienne et la méthode de Monte Carlo. Il combine les avantages de la conception expérimentale bayésienne et de la méthode de Monte Carlo pour permettre de prendre des décisions optimales sur la manière de mener une expérience de manière itérative et en temps réel.

Le SMC-BED est particulièrement utile dans les situations où les données sont difficiles à obtenir et où il est important de maximiser l'utilité de chaque mesure prise. Il peut être utilisé pour optimiser la précision de l'estimation d'un paramètre inconnu ou pour atteindre d'autres objectifs tels que la minimisation du temps d'expérience ou la maximisation de la vitesse de convergence.

Le SMC-BED fonctionne en utilisant un algorithme de filtre de particules pour estimer la distribution de probabilité du paramètre inconnu à mesure que de nouvelles données sont observées. La fonction d'utilité est utilisée pour déterminer les paramètres de contrôle optimaux à chaque étape de l'expérience, ce qui permet de maximiser l'utilité de chaque mesure prise.

Le SMC-BED est une approche puissante pour optimiser la conception d'expériences dans des situations où les données sont difficiles à obtenir et où il est important de maximiser l'utilité de chaque mesure prise. Il combine les avantages de la conception expérimentale bayésienne et de la méthode de Monte Carlo pour permettre de prendre des décisions optimales sur la manière de mener une expérience de manière itérative et en temps réel.

f. Partie algorithmique

Le modèle proposé peut être résumé par le schéma suivant:

En premier temps on effectue un échantillonnage de particule à partir du prior initial, puis selon l'approximation on

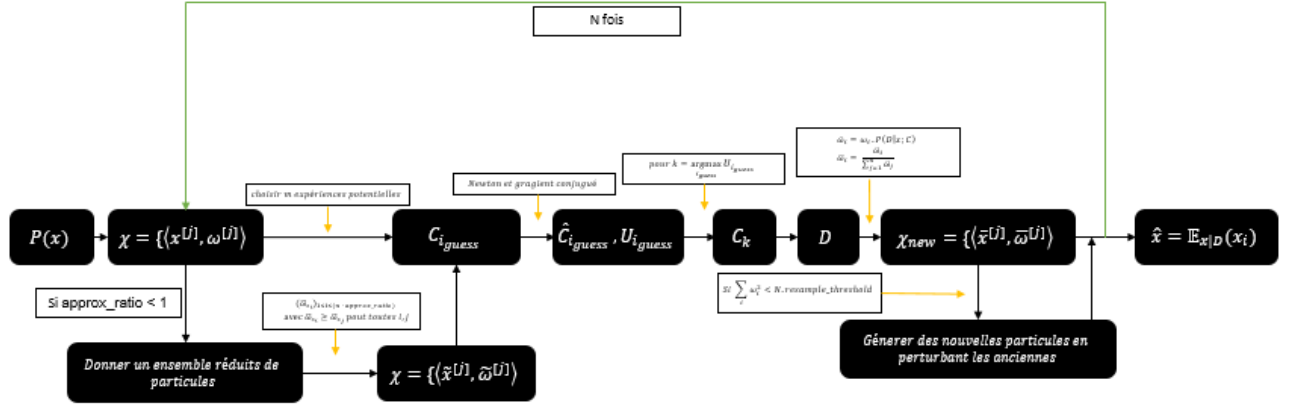


Fig. 2: Algorithm du modèle.

assemble les particules deux cas s'impose: soit on prend $approx_ratio < 1$ et dans ce cas on prend les particules à poids maximal, soit on prend $approx_ratio = 1$ et on passe directement vers la partie suivante.

On applique par la suite la méthode Newton Conjugate Gradient pour trouver le paramètre C_{opt} à l'issu de plusieurs expérience en choisissant celui avec un gain d'information important(cf. fonction d'utilité maximale), par la suite on applique le modèle et on collecte la data puis en boucle tel qu'on prend les particules avec un poids maximal.

III. RÉSULTAT

Dans cette partie, on se propose de discuter les résultats du modèle proposé. Pour ce faire, on étudie trois cas dans lesquels le paramètre à estimer est la pulsation de l'Hamiltonien ω , l'hyperparamètre de contrôle est le temps de perte d'information T_2 .

a. Résultat pour T_2 connue

Pour cette première expérience, on étudie la performance du modèle lorsque ω suit une loi Normale $\omega \sim \mathcal{N}(0.5, 0.01)$ et T_2 supposé de l'ordre de 100π (cette valeur n'est pas arbitraire, elle est prise aussi grande pour éviter tous bruit et isoler le système). Pour cette expérience le temps est pris uniforme avec un maillage de $\frac{2\pi}{3}$. Les résultats ci-dessous représente la perte du modèle en fonction des itérations pour différents nombres de particules et différents optimisations de ce dernier.

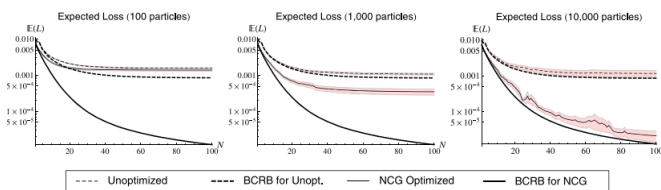


Fig. 3: Extraction des particules

Sous les mêmes hypothèses on se propose maintenant d'étudier plus en détail l'erreur du modèle. Par cette deuxième expérience, on met en évidence l'importance de l'entraînement du modèle. Pour la méthode optimisée on

prend respectivement, pour l' $approx - ratio$, les valeurs 0.1 et 1 dans le but d'avoir plus de généralisation. Le graphe à gauche présente l'erreur pour une seule réalisation et celui à droite montre la meilleur de 30 réalisations. soit:

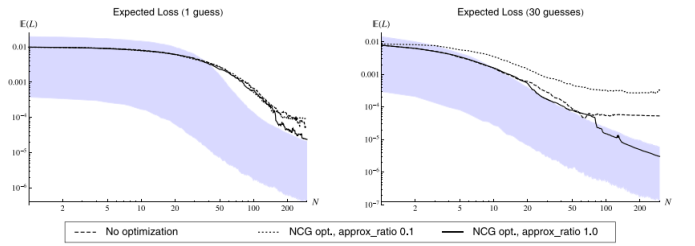


Fig. 4: Performance du modèle pour 1 et 30 prédictions. Selon la figure, l'erreur quadratique moyenne du paramètre omega diminue lorsque nous augmentons le nombre de particules de 100 à 10 000, atteignant un taux inférieur à 1% pour $N > 100$. L'optimisation de gradient conjugué non linéaire permet d'approcher la borne inférieure de l'erreur quadratique moyenne (BCRB) avec $N = 10\,000$ particules dans l'approximation de la méthode de la particule de Monte-Carlo. La figure 4 fournit une analyse plus approfondie de l'évolution de l'erreur pour le modèle T_2 connu. Nous avons constaté que l'erreur quadratique moyenne était réduite si nous choisissons le meilleur parmi 30 expériences possibles plutôt qu'une seule expérience choisie aléatoirement. Nous avons également observé que l'optimisation locale tendait à améliorer la qualité de l'approximation si nous choisissons un ratio d'approximation égal à 1. La figure 4 montre également que des valeurs plus petites du ratio d'approximation peuvent entraîner une saturation de l'erreur quadratique moyenne à des valeurs relativement élevées. En fait, le choix d'un ratio d'approximation égal à 0,1 était suffisant pour que les données pour 30 hypothèses et l'optimisation NCG aient une erreur quadratique moyenne plus élevée que le cas sans optimisation et une seule hypothèse aléatoire. Pour cette raison, nous avons choisi un ratio d'approximation égal à 1 dans la plupart des exemples de cette section. Nous verrons un exemple où un ratio d'approximation inférieur à 1 est plus bénéfique dans la section de discussion. Des intervalles de confiance à 68% pour la perte réelle montrent que l'erreur quadratique moyenne est

souvent en dehors de ces intervalles lorsque l'optimisation NCG n'est pas utilisée. Cela suggère que la distribution de l'utilité des expériences peut varier considérablement et que la distribution est déformée car une fraction significative des hypothèses ne fournissent pratiquement aucune information. L'optimisation NCG minimise les chances de choisir une expérience peu informative et force ainsi les hypothèses vers les expériences les plus informatives. Nous devrions également noter qu'il y a encore de la marge de progrès, car l'erreur quadratique moyenne se rapproche de la limite supérieure de l'intervalle de confiance. Des procédures d'optimisation plus sophistiquées peuvent donc être utiles.

b. Résultat pour T_2 inconnue suivant une loi normale

Dans ce cas on prend 2 paramètres, ie: ω et T_2 suivant respectivement les lois normales $\mathcal{N}(0.5, 0.0025)$ et $\mathcal{N}(0.5, 0.0025)$. Le modèle appliqué à ce cas génère le résultat suivant ou la partie en rouge représente l'erreur dans les deux cas pour une et 30 réalisations. soit:

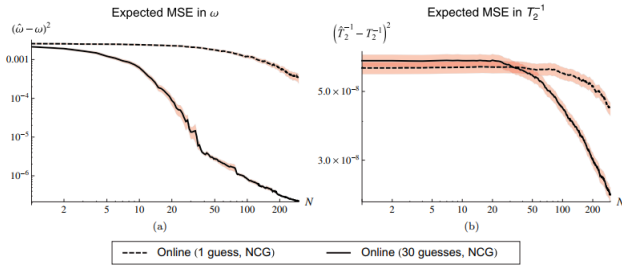


Fig. 5: Performance du modèle pour 1 et 30 prédictions

l'erreur quadratique moyenne (MSE) est utilisée comme mesure de la précision des estimations de deux quantités, ω et T_2 . On constate également qu'utiliser plus de devinettes, ou de valeurs potentielles pour ces quantités, conduit à une amélioration significative de la MSE pour ω . De plus, l'utilisation de la méthode NCG conduit à une amélioration supplémentaire de la MSE pour ω , sans atteindre un point de saturation. Cependant, l'utilisation de NCG n'a qu'un effet subtil sur la MSE pour T_2 . On peut ainsi remarquer que l'utilisation d'une seule prédiction accompagné NCG n'améliorait pas significativement la MSE pour ω , ce qui implique que les valeurs possibles de cette dernière ne sont pas suffisamment convexe pour que l'optimisation locale soit efficace. En résumé, l'étude a montré que l'utilisation de plus de prédiction et de NCG conduit à une meilleure précision dans les estimations de ω , en contre partie cela n'affecte pas considérablement les estimations de T_2 .

c. Performances d'estimation de la région d'hyperparamètres

Dans ce cas, il s'agit de l'hyperparamétrisation du processus. L'objectif ici est d'analyser la robustesse de l'algorithme de estimation de régions de paramètres hyperparamétrés, on parle ainsi de distributin de mass des pericules entre 2 états, les graphs ci dessous présentent l'erreur:

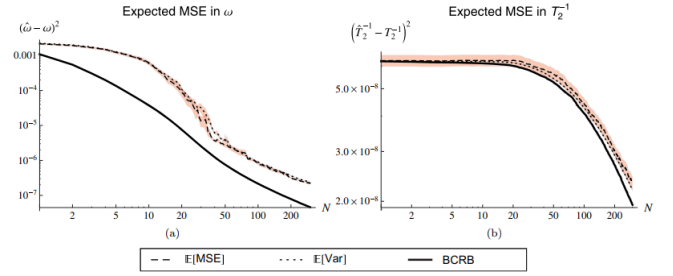


Fig. 6: Performance du modèle pour les 2 paramètres
Les résultats sont similaires et plus précises avec une complexité algorithmique très grande.

IV. DISSCUSSION ET AMÉLIORATION DU MODÈLE

nous avons donc pensé à améliorer la qualité de l'algorithme tout en maintenant un temps d'exécution court. Pour ce faire, nous avons ajouté trois fonctions :

1. *choose_experiments* : cet algorithme utilise la recherche dichotomique pour trouver les paramètres de contrôle qui maximisent l'utilité d'une fonction donnée. La recherche dichotomique est un algorithme de recherche qui consiste à diviser à plusieurs reprises l'intervalle de recherche en deux parties égales jusqu'à ce que la solution souhaitée soit trouvée. Dans ce code, l'algorithme continue de diviser l'intervalle de recherche en deux jusqu'à ce qu'il trouve l'expérience optimale.

2. *diff* : cette fonction compare la vraisemblance du résultat de la ième expérience 0 avec le résultat de la ième expérience 1, sachant que nous avons imposé un ensemble spécifique de paramètres C et que nous avons un ensemble de résultats D . Il délivre le ième résultat en imposant le paramètre optimal C_{opt} .

Nous supposons que le paramètre ω suit une distribution exponentielle avec le paramètre lambda. Nous avons appliqué ce nouveau code dans le premier cas et nous avons obtenu les résultats dans la figure 7 et la figure 8

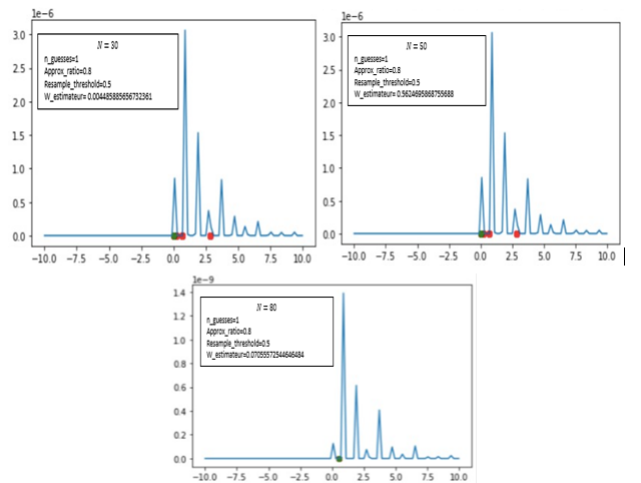


Fig. 7: Résultat de l'Application sur le premier cas pour $nguess = 1$
Sur la figure 7, nous avons imposé $Nguess=1$, et nous avons remarqué que même si nous avons appliqué une quantité considérable d'expérience, nous avons pu obtenir des résultats pertinents avec un temps de mise en œuvre raisonnable (80 itérations en 3 minutes). Le fait que D change à

chaque fois que nous exécutons le code nous permet de tester tous les états possibles et de donner des estimations dans chacun d'eux, de sorte que la distribution de probabilité diminue avec chaque itération. Ceci réduit l'intervalle d'existence de ω . Modifier la fonction qui modélise notre expérience, à savoir *diff*, permet à notre algorithme de nous donner des résultats plus pertinents.

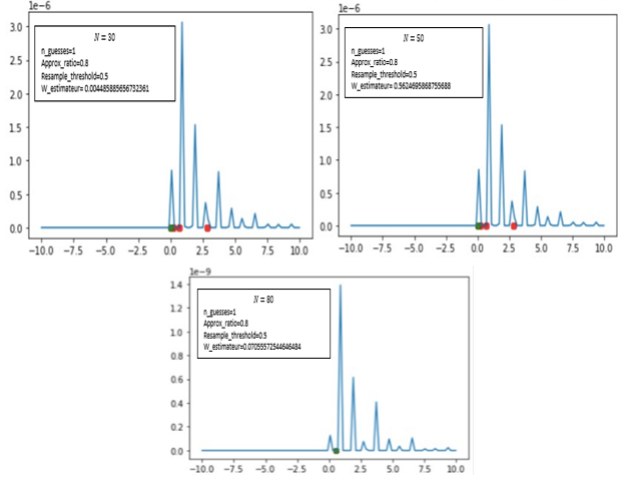


Fig. 8: Résultat de l'Application sur le premier cas pour $N_{guess} = 10$

Dans la Fig. 8, nous avons imposé $N_{guess} = 10$, dans ce cas nous testons la qualité de la fonction *choose_experiments* qui choisit le meilleur paramètre de réglage C . On constate que pour 20 expériences (10 itérations chacune), le temps d'exécution du code reste raisonnable à 2 minutes par expérience) et donne des résultats pas moins bons que le premier cas.

V. CONCLUSION

La méthode proposée dans cet article consiste en l'utilisation d'un algorithme combinant la commande de modèle stochastique (SCM) et la conception bayésienne d'expérience (BDE) pour l'apprentissage automatique de l'Hamiltonien. Les résultats obtenus montrent que cette approche permet d'améliorer la précision de l'apprentissage de l'Hamiltonien par rapport à l'utilisation de l'un de ces deux outils seuls. En particulier, l'algorithme combiné SCM-BDE a permis de réduire l'erreur de prédiction de l'Hamiltonien de manière significative dans les cas d'étude testés.

En conclusion, cet article démontre l'efficacité de l'utilisation de l'algorithme SCM-BDE pour l'apprentissage automatique de l'Hamiltonien. Il s'agit d'une avancée importante dans le domaine de l'apprentissage automatique, mais avec une complexité plus ou moins grande, ainsi on a proposé un modèle plus optimisé qui donne des résultats avec la même précision mais avec un temps de réalisation moins coûteux et une perte d'information moins importante.

REFERENCES

[1] "Online Learning of Hamiltonian Dynamics" by Simon S. Du, Yinyu Ye, and Stephen P. Boyd (<https://arxiv.org/abs/1712.09662>)

[2] "Data-Efficient Hamiltonian Learning for Particle Dynamics" by Yotam Hechtlinger, Jason J. Kim, and Aleksander M. Madry (<https://arxiv.org/abs/2006.04061>)

[3] "Learning Hamiltonian Dynamics with Graph Neural Networks" by David E. Kappel, Karthik Kashinath, and J. Zico Kolter (<https://arxiv.org/abs/2010.05362>)

[4] Wackerly D, Mendenhall W and Scheaffer R L 2001 Mathematical Statistics with Applications Mathematical Statistics 6th edn (Pacific Grove, CA: Duxbury)

[5] Doucet A and Johansen A M 2009 A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later (Oxford: Oxford University Press)

[6] Loredo T J 2004 AIP Conf. Proc. 707 330

[7] Kuck H, de Freitas N and Doucet A 2006 Nonlinear Statistical Signal Processing Workshop (Piscataway, NJ: IEEE) pp 99–102

[8] Scarpa B and Dunson D B 2007 Stat. Med. 26 1920

[9] Cavagnaro D R, Pitt M A and Myung J I 2009 Advances in Neural Information Processing Systems vol 22 pp 234–42

[10] Kantas N, Lecchini-Visintini A and Maciejowski J M 2010 Int. J. Adapt. Control Signal Process. 24 882

[11] [10] Huan X and Marzouk Y M 2011 arXiv:1108.4146