Transporte de estados coherentes usando aprendizaje reforzado profundo

Andrés F. Guerrero, Kennet J. Rueda ⋆
Universidad Nacional de Colombia, Facultad de Ciencias
Introducción a la investigación teórica

Junio 2022

Abstract
El transporte coherente de estados a través de un paso adiabático entre puntos cuánticos es un problema muy estudiado y de este se conoce una solución de ansatz para sus casos más simples. Sin embargo, cuando el sistema es perturbado, como en el caso de detuning en el que se tienen energías diferentes para cada estado base, estas soluciones ya no son efectivas. El aprendizaje profundo por refuerzo (DRL por sus siglas en ingles) es una herramienta poderosa capaz de resolver problemas de toma de decisiones secuenciales sin tener conocimiento previo del problema. En este trabajo se hace uso de un agente de DRL para solucionar el problema de transporte adiabático de estados coherentes con el fin de probar su eficacia para resolver el problema donde no se conocen soluciones de anzats. Inicialmente, se prueba el agente en el sistema más simple de tres puntos cuánticos sin perturbaciones. Se encontró que el agente es capaz de solucionar el problema aplicando pulsos contra-intuitivos tal como se esperaba por las soluciones anzats. Seguido, se adapto agente y el entorno para conseguir una solución del transporte cuando se tiene una perturbación tipo detuning. Se comprobó que el agente realiza transporte a través de los puntos cuánticos con un aumento en la densidad de población del estado intermedio. Se concluye que es posible hacer ajustes finos en el modelo para conseguir un transporte limpio. Finalmente, se pone a prueba la eficacia del agente cuando se realiza transporte entre 5 y 7 puntos cuánticos.

1 Introducción

La inteligencia artificial ha probado ser capaz de encontrar reglas y soluciones a problemas de alta complejidad en un amplio rango de aplicaciones que pasan tanto por las ciencias médicas Williamson et al. (2020),Wong & Yip (2018), como por el diseño de materiales mecánicos Guo et al. (2020) y optimización en ingeniería En particular, el paradigma del aprendizaje reforzado Sutton & Barto (2018) presenta un potencial notable por su capacidad de abordar problemas sin contar con datos previos acerca del mismo. Aunque este paradigma es ya bien conocido en aplicaciones de ingeniería que requieren de sistemas de control o en el campo de los videojuegos, en problemas de física y en particular, en sistemas cuánticos recién se conocen aplicaciones en años recientesMoro et al. (2021), Fösel et al. (2021). En este trabajo se busca hacer uso de esta herramienta para producir soluciones al problema de transporte de estados entre puntos cuánticos, en particular, para transportar estados de espín entre puntos cuánticos acoplados a primeros vecinos. Este problema ya ha sido abordado teóricamente por varios autores Greentree et al. (2004)Ferraro et al. (2015), Gullans & Petta (2020) y se han encontrado soluciones a partir de propuestas o Ansatz que funcionan en los casos mas simples donde los estados de cada punto tienen niveles de energía iguales, sin embargo, la carencia de un desarrollo general impide encontrar soluciones directas al problema cuando los estados tienen niveles de energía distintos. Acá proponemos el uso de un agente de aprendizaje reforzado, que a partir de interacciones con simulaciones del sistema, aprenda a generar secuencias de pulsos que permitan realizar transporte. A continuación explicaremos el modelo del sistema físico de puntos cuánticos, seguidamente discutiremos los métodos numéricos usados para plantear la dinámica del mismo, así mismo, se explica de manera somera el paradigma de aprendizaje reforzado haciendo enfasis en el agente construido. Por último presentamos un análisis de los resultados que finaliza con una corta discusión de las problemáticas presentadas y reflexiones para trabajos futuros.

2 Sistema físico

2.1 Modelo y hamiltoniano del sistema

El sistema físico acá estudiado corresponde a un arreglo de puntos cuánticos en cuyos estados de mínima energía se encuentra ubicado un electrón, los espines de estos electrones se acoplan a través de una interacción de intercambio que se modela a partir de un hamiltoniano efectivo de heisenberg. El hamiltoniano completo para este problema puede escribirse como Gullans & Petta (2020):

$$H = \sum_i g\mu_B \boldsymbol{B}_i^{\text{tot}} \cdot \boldsymbol{s}_i + \sum_{i,j} J_{ij}(t)\left(\boldsymbol{s}_i \cdot \boldsymbol{s}_j - 1/4\right) \qquad (1)$$

donde B_i es un campo magnético total para cada punto, resultado de campos globales y locales, por otra parte J_{ij} es el termino que modula el acople entre espines. El campo magnético anterior tiene efecto sobre los estados de mínima energía de cada punto y a partir de consideraciones necesarias sobre su valor y su frecuencia de oscilación Gullans & Petta (2020), puede asumirse constante respecto al término de intercambio de forma tal que podemos centrarnos únicamente en este último término. Así, el hamiltoniano que nos interesa es el que contiene el termino de intercambio entre espines.
Podemos escribir nuestro sistema en la base de los estados de espín de cada punto de forma tal que si consideramos que uno de los puntos tiene un espín antiparalelo a los otros dos, tenemos que nuestra base serán los estados $\{|\uparrow\downarrow\downarrow\rangle, |\downarrow\uparrow\downarrow\rangle, |\downarrow\downarrow\uparrow\rangle\}$, de donde el hamiltoniano puede escribirse como:

$$H_0 = \begin{pmatrix} \Delta_{12} & j_{12}(t) & 0 \\ j_{12}^*(t) & \Delta_{13} & j_{23}(t) \\ 0 & j_{23}^*(t) & 0 \end{pmatrix} \quad (2)$$

los terminos j_{12} y j_{23} también son llamados Ω_{12} y Ω_{23} y son los coeficientes del acople entre los puntos $1-2$ y $2-3$.

2.2 Problema de transporte

El problema de transporte que acá nos concierne implica pasar de un estado $|\uparrow\downarrow\downarrow\rangle$ a uno $|\downarrow\downarrow\uparrow\rangle$ pasando un tiempo aproximadamente cero en el estado $|\downarrow\uparrow\downarrow\rangle$. Para esto, es necesario encontrar la secuencia de valores correctos de Ω_{12} y Ω_{23}, que cumpla los resultados deseados.
Como se comentó en la primer sección, existen soluciones a este problema para cuando los niveles de energía correspondientes a estos estados es el mismo, esto es, que los elementos de la diagonal del hamiltoniano sean $\Delta_{12} = \Delta_{13} = 0$, estas soluciones se muestran y discuten en la sección de resultados. Sin embargo, este trabajo busca encontrar soluciones a los casos en los que $\Delta_{12} \neq 0$ y $\Delta_{13} \neq 0$.

3 Metodología

A fin de encontrar la secuencia de pulsos que permita realizar transporte, se plantea la creación de un entorno y un agente de aprendizaje reforzado que interactúe con el primero para aprender a generar los pulsos.

3.1 Aprendizaje reforzado

El aprendizaje reforzado es un área del aprendizaje automático inspirado en la psicología conductual, en el que se espera que a través de la interacción de una máquina virtual con un entorno, esta aprenda a tomar secuencias de series de decisiones que maximicen el valor esperado del refuerzo, que son las recompensas otorgadas por el entorno según las acciones tomadas por la máquina.
De manera formal, decimos que un modelo de aprendizaje reforzado está dado si se tiene: Un conjunto de acciones $a_{t_i} \in A$ que un agente puede tomar, un conjunto de estados $s_{t_i} \in \mathcal{S}$ que describen al sistema en un tiempo t, una función de refuerzo $r : \mathcal{S} \longrightarrow R$ que otorga a cada estado un valor real y un agente y un entorno que se encargan de tomar una acción $a_{t_{i+1}}$ a partir de un estado s_{t_i} y de producir un estado $s_{t_{i+1}}$ a partir de una acción $a_{t_{i+1}}$ y un estado s_{t_i}, respectivamenteSutton & Barto (2018). A continuación se mostrará como se construyó el entorno para este problema.

3.2 Construcción del entorno

Las acciones y estados acá definidos están inspirados en el sistema físico. Diremos que las acciones que nuestro agente puede tomar son los coeficientes de acople Ω_{12} y Ω_{23} entre los espines. Luego $a_{t_i} = [\Omega_{12}(t_i), \Omega_{23}(t_i)]$, de manera semejante, el estado de nuestro sistema estará descrito por los elementos de la matriz densidad del sistema cuántico para un instante de tiempo t_i. Adicionalmente, agregaremos al estado $s(t_i)$, las acciones tomadas en el instante de tiempo anterior, así tenemos que Porotti et al. (2019):

$$s_{t_i} = [|\rho_{nm}(t_i)| \text{ para } m, n \in \{1, 2, 3\}, \Omega_{12}(t_{i-1}), \Omega_{23}(t_{i-1})]$$

Una vez definidas nuestras acciones y estados, el entorno debe poder generar estados para un tiempo posterior a partir de las acciones y el estado en el tiempo actual. Esto se puede realizar si se calcula la evolución de la matriz densidad $\rho(t_i)$ usando los valores de $\Omega(t_i)$ dentro del hamiltoniano y se obtiene $\rho(t_{i+1})$. Para esto, una vez se tienen las acciones y el estado inicial del sitema se utiliza la ecuación de Von-Neumann:

$$\dot{\rho} = -\frac{1}{i}[H, \rho] \quad (3)$$

que describe la dinámica de un sistema cuántico a partir de su matriz densidad. En esta representación de estados, la ecuación resulta ser un sistema de ecuaciones de primer orden que puede ser resuelto numéricamente puesto que consideramos que el hamiltoniano es constante en el intervalo Δt en el que evaluamos la dinámica.
De esta forma, teniendo un estado $s(t_i)$ dado a partir de $\rho(t_i)$, y unas acciones tomadas por un agente $a(t_i)$ que definen el hamiltoniano H del sistema, en este instante de tiempo discreto, podemos resolver numéricamente la ecuación (3) desde el instante inicial hasta un instante final $t_{i+1} = t_i + \Delta t$, lo que nos permite tomar el valor final para obtener $\rho(t_{i+1})$ en un nuevo instante de tiempo discreto. Todo esto, considerando que durante el intervalo Δt, el hamiltoniano es constante.
Así, constituimos un entorno que mapea las acciones y los estados en estados nuevos. En este trabajo se utilizó la función *Odeintw* Weckesser (2021), que es una extrapolación de la función odeint del paquete Scipy a los números complejos, para resolver el sistema de ecuaciones de (3).

3.3 Función refuerzo

La función refuerzo del sistema está definida principalmente como

$$r_{t_i} = -1 + k\rho_{33}(t_i) - \rho_{22}(t_i), \quad (4)$$

donde se observa que se premian valores mas altos de población para el tercer estado y se penalizan aumentos en la población del segundo. El valor de k se usó inicialmente como $k = 1$, aunque se realizaron variaciones sobre el mismo, presentadas en la sección siguiente. Al refuerzo otorgado anteriormente se le suman dos valores bajo condiciones distintas. En caso de que la población del segundo estado exceda un valor límite de 0.05, si $\rho_{22} > 0.05$, entonces al refuerzo se le agrega una penalización de -100, adicionalmente, si el agente logra superar un valor de población en el tercer estado de $\rho_{33} > 0.985$, consideramos que consiguió el objetivo y se asigna un refuerzo de 1000.

3.4 Agente y entrenamiento

El agente para este problema se corresponde con un algoritmo DDPG Lillicrap et al. (2015), que es de la clase llamada algoritmos actor-crítico, en este se utilizan dos redes neuronales para aprender a realizar una predicción correcta de la secuencia de pulsos deseada. La red actor toma la decisión de las acciones y por lo tanto escoge ambos valores Ω_{12} y Ω_{23}, la red crítica por su parte se encarga de juzgar la acción a partir de la predicción de el *Valor acción- estado* que es el valor esperado del refuerzo total para una secuencia de pulsos, partiendo de un estado s y un estado a, para la red actor actual $Q^\pi(s, a) = E[R \mid s, a, \pi]$. La red actor toma la recomendación de la crítica, que viene en forma de Q, y a partir de esto actualiza el valor de sus parámetros, es decir, los pesos de la red. El algoritmo DDPG

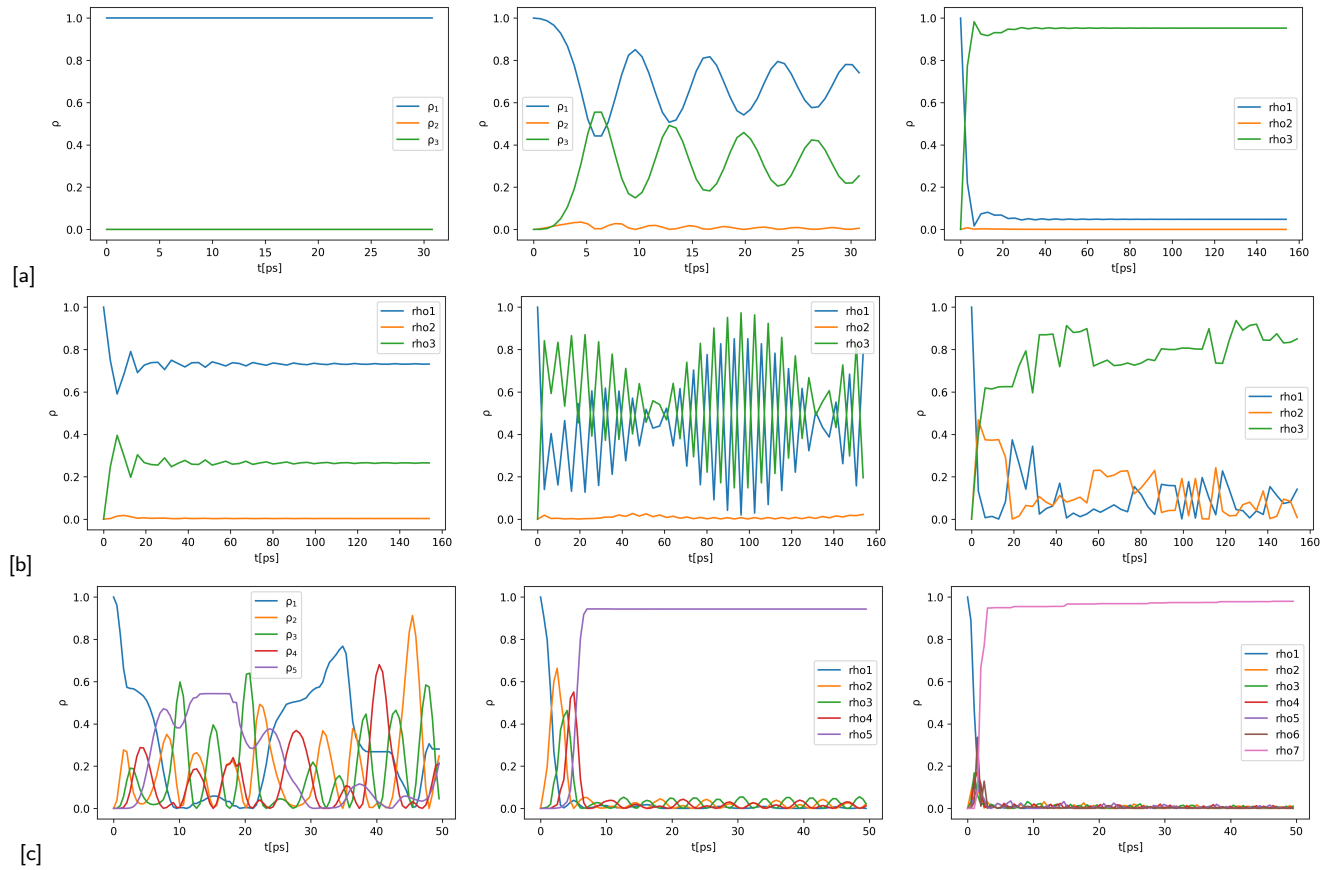

Figure 1. *Gráficas de las densidades de probabilidad para cada caso estudiado. a) En la primera fila se muestran los resultados obtenidos para el sistema de tres puntos cuánticos: 1) 6000 pasos y redes neuronales básicas. 2) 5000 pasos y la función refuerzo descrita en 3.3. 3) 7000 pasos, sin cambios en el refuerzo. b) La segunda fila corresponde a los resultados encontrados para un sistema de tres puntos cuánticos cuando se incluye una perturbación tipo detuning. 1) 7000 pasos y función refuerzo igual a la usada para los 3 puntos cuánticos sin perturbación. 2) 7000 pasos y refuerzo con una constante $k = 20$ y una penalización exponencial para ρ_2. 3) 100000 pasos con refuerzo modificado y descrito en 4.2. c) La ultima fila corresponde a los resultados obtenidos para un sistema de 5 y 7 puntos cuánticos.1) 10000 pasos para 5 puntos cuánticos con penalización exponencial.2) 10000 pasos para 5 puntos cuánticos con recompensa exponencial. 3) 50000 pasos para 7 puntos cuánticos con refuerzo con penalizaciones y refuerzos exponenciales.*

está bastante desarrollado y contiene detalles adicionales como el uso de buffers de entrenamiento y redes objetivo, elementos que se escapan del interés explicativo de este artículo. Para una explicación detallada remitirse a Lillicrap et al. (2015).
A continuación se muestran algunos de los hiperparametros relevantes usados durante la construcción de la red.

Capacidad del Batch	2000
Tamaño del batch	20
Optimizador	Descenso del gradiente
Taza de aprendizaje	10^{-3}
Factor Gamma	9.9×10^{-2}
Factor Tau	5×10^{-3}
Layers	2
Neuronas por capa	$\sim 10^2$

Para el desarrollo del algoritmo se utilizó la librería Tensorflow.

4 Resultados y análisis

A continuación se muestran los resultados detallados obtenidos por el agente de DDPG construido para las distintas variantes del problema. Es importante conocer de antemano las soluciones anzats encontradas para el problema particular de puntos cuánticos sin perturbaciones, esto porque en general esperamos que las acciones tomadas por el agente tengan la misma forma funcional. En la figura 2[a] observamos las gráficas de los anzats para 5 o mas pulsos. Solo son necesarios 3 pulsos ya que si se quiere aumentar el numero de puntos cuánticos la interacción entre los puntos intermedios esta dado por Ω_3, Para 3 puntos cuánticos basta con eliminar este pulso. Se dice que la solución es contra-intuitiva porque se aplica primero el pulso correspondiente a la interacción entre el punto 2 y 3 (ó el ultimo punto con su vecino cercano en caso de tener mas de tres puntos cuánticos), seguido por el pulso relacionado con la interacción entre los puntos 1 y 2.

4.1 Sistema de tres puntos cuánticos sin perdidas

Inicialmente se estudió el trasporte de estados cuánticos coherentes en un sistema de tres puntos separados espacialmente, Como se mencionó anteriormente, para este problema existe

solución de Anzats. Se usó el agente anteriormente descrito y se compararon los pulsos obtenidos del mismo con dichas soluciones. Para el estudio, se vario el número de pasos para entrenar el agente y la función refuerzo se usó de acuerdo a la recomendación de los autores (Star (2013)), se encuentra descrita en la sección 3.3. Algunos resultados obtenidos se muestran en la fig 1[a]. En la primera imagen, observamos un resultado muy común en el que el agente determina que el estado óptimo es no evolucionar el sistema, esto se debe a que no explora lo suficiente ya que encuentra un mayor refuerzo negativo al incrementar la densidad de estado para el punto intermedio que la ganancia obtenida al poblar el estado final. Otra razón por la que se obtiene este resultado es que el numero de neuronas en las capas de redes son muy bajas, esto implica que la red no alcanza el nivel de complejidad requerido para realizar el transporte. Seguido, si se aumenta el número de repeticiones, se observa que la densidad de estado en $|\downarrow\uparrow\downarrow\rangle$ no aumenta y hay una disminución de la densidad de probabilidad ρ_1 y proporcionalmente un aumento en la densidad de probabilidad ρ_3, finalmente se encuentra un comportamiento sinusoidal entre estas densidades. Dado que el resultado se acerca al esperado ya que no se esta poblando el punto intermedio, se optó por aumentar el numero de repeticiones para entrenar el agente a 7000 pasos. Finalmente, se encontró que el transporte ocurre en un tiempo considerablemente menor que al hacerlo mediante las soluciones anzats. Si bien, el transporte es inesperadamente rápido, no se tienen razones para sospechar que el sistema esté mal planteado dado que se realizaron todo tipo de pulsos encontrando que solo se observa transporte cuando se tienen los pulsos adecuados. Podemos observar de la figura 2[b], que las acciones del agente son consistentes con las soluciones anzats. Si bien, en este caso los pulsos se aplican desde el tiempo inicial para ambos coeficientes, el pulso correspondiente a la interacción entre el punto 2 y 3 se prolonga por mas tiempo y con mayor intensidad. Una vez comienza a disminuir el pulso Ω_2, el pulso Ω_1 aumenta y finalmente se hacen constantes los dos. Sabemos por la solucion anzats y por el articulo que usamos de referencia que el cambio de estado se produce en la intersección de los pulsos, dado que nuestro agente realiza los pulsos simultáneamente en el tiempo inicial, creemos que es por esto que el transporte se realiza en un tiempo tan corto. La densidad en la población del punto 2 no sube y en el tiempo final se encuentran densidades de probabilidades cercanas a 1 y 0 para los puntos 3 y 1 respectivamente, por esto concluimos que se realizo transporte de forma exitosa.

4.2 Sistema de tres puntos cuánticos con Detuning

Una vez que se obtuvieron resultados favorables para el caso mas simple, se introducen efectos de perturbaciones, en particular se estudia el efecto del detuning que es introducido en el sistema cuando las diferencias eigenenergias de los puntos no es cero. Es decir, cuando los puntos poseen diferentes auto energías. Como se mencionó anteriormente, para este tipo de sistemas no se tiene una solución de ansatz. Se puso a prueba el agente (fig 1[b]) usando la misma función de refuerzo usada para el caso en el que no existe la perturbación. Se encontró, como se esperaba, que el agente no consigue aumentar la población en el punto 3 pero se resalta que mantienen, en aproximadamente 0, la densidad en el estado ρ_2. Seguido se reescribió la función refuerzo siguiendo la recomendación de los autores en la información suplementaria Porotti et al. (2019). En el que se penaliza el aumento de la densidad de estado en

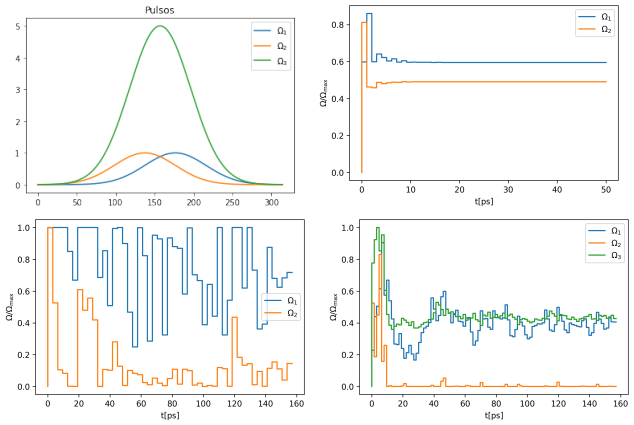

Figure 2. *Gráficas de los pulsos aplicados para conseguir transporte adiabático. a) En la esquina superior izquierda se encuentran los pulsos dados por las soluciones anzats. b) En la esquina superior derecha se encuentran los pulsos dados por el agente para 3 puntos cuánticos. c) La imagen inferior izquierda muestra los pulsos aplicados cuando se tiene detuning. d) La imagen inferir derecha muestra los pulsos para transporte cuando se tienen 7 puntos cuánticos*

ρ_2 de forma: $B(t_i) = e^{\rho_{22}}$. Se entrenó el agente hasta con, 70000 pasos y se encontró que el agente no consigue poblar el punto cuántico 3 de forma consistente y se observa un comportamiento "aleatorio". Mediante diferentes pruebas con distintos pasos se concluyó que el agente encontraba óptimo mantener ρ_1 y ρ_3 constantes o oscilando entre ellos. Esto puede deberse a que una vez aumenta el refuerzo en los tiempos iniciales no consideraba óptimo explorar más. Esto puede deberse a que en caso de poblar el punto 2 sería fuertemente castigado de forma exponencial. Se propuso partir un refuerzo más bajo (-100) así de esta forma no es suficiente con el refuerzo ganado en los tiempos iniciales, aumentamos la recompensa linealmente con un factor k = 10 (descrito en la sección 3.3) y finalmente, bajamos la penalización al poblar el punto 2 con el fin de darle más libertad a explorar y que no encuentre que las acciones óptimas son no transportar los estados. Finalmente se consiguió que, al igual que con 3 puntos cuánticos sin perturbación, se realiza transporte en un tiempo corto con una densidad de probabilidad más alta de lo conveniente para el segundo punto. Aunque se ejecutaron muchas más pruebas cambiando la función de refuerzo, no se encontró un resultado más cercano al esperado que el mostrado en la figura 1[b]. Claramente las acciones encontradas por el agente (fig 2[c]) distan de las soluciones anzats. Esto es totalmente esperado dado que se introdujo perturbaciones naturales al sistema. Sin embargo, podemos observar que se conserva la duración de los pulsos, es decir, las acciones se realizan simultáneamente en la misma magnitud, sin embargo, el pulso Ω_2 comienza a decaer mas rápidamente mientras que el pulso correspondiente a la interacción entre los puntos 1 y 2 perdura por mas tiempo en un nivel mas alto.

Creemos que si se refina la penalización por poblar el estado intermedio se pueden encontrar mejores resultados. Pero aún más importante, creemos que con un número mayor de repeticiones el agente podría se capaz encontrar una mejor política y mejore el resultado encontrado. La potencia computacional es un factor que afecta significativamente, la falta de los mismos influye en no realizar pruebas con mayor numero de repeticiones que implica un mejor entrenamiento.

4.3 Sistema de 5 y 7 puntos cuánticos.

Como objetivo final se planteó probar el agente con 5 y 7 puntos cuánticos. Para esto se probó el agente con las mismas funciones de refuerzo usadas anteriormente. Se encontró que en general para más de 3 puntos cuánticos el agente tiende a escoger una configuración en la que mantiene las densidades de probabilidad constantes. Esto puede deberse a que es castigado aún más fuerte que en el caso de 3 puntos cuánticos. Esto se debe a que eventualmente encontrará un castigo correspondiente a incrementar la densidad de estado en los 3 y 5 estados intermedios respectivamente. Se probaron diversas funciones de refuerzo en el que se incentiva la exploración al no penalizar tan fuerte el hecho de que los puntos intermedios sean poblados, pero asegurando que se castigue exponencialmente el aumento en la densidad de portabilidad de dichos estados para asegurar que no aumenten mas de lo estrictamente necesario. De igual manera se recompensa tanto exponencialmente como linealmente con factores de k $= 20$ el aumento en la población del último punto cuántico. En la figura 1[c] se puede observar que para 5 puntos cuánticos se alcanza a tener transporte pero, la densidad en los estados intermedios aumenta incluso hasta 0,6. Esto no es favorable y en general debemos tratar de reducir el aumento de estas probabilidades. Se realizaron diferentes pruebas entrenado la red con diversos números de pasos y cambiando la función refuerzo, sin embargo no se llego a un mejor resultado que al mostrado en la gráfica. Concluimos que es cuestión de ensayar otras funciones, darle mas tiempo de ejecución y ampliar el numero de neuronas en las redes para encontrar el transporte tal como se espera.

Por otro lado, La ultima gráfica en la figura 1[c] muestra el transporte adiabático para un sistema de 7 puntos cuánticos. Se observa que el estado ρ_{77} se pobla con una densidad de probabilidad cercana a 1. De igual forma, los estados intermedios no superan un 0.2 en el momento del transporte. Esto nos indica que se realizo transporte con una eficiencia bastante alta. La figura 2[d] muestras las acciones del agente que usa para alcanzar el transporte. Los pulsos siguen una forma funcional parecida a la obtenida para los otros casos. Finalmente, cabe resaltar que para este caso teníamos un resultado similar al mostrado para los 5 puntos cuánticos, sin embargo, calibrando la función refuerzo fue posible obtener el resultado de transporte adiabático.

Se resaltan los resultados obtenidos, se cubrió el espectro de los objetivos específicos planteados. Pero también se tiene presente que es posible mejorar los resultados principalmente aumentando las neuronas en las capas usadas y refinado las funciones refuerzo para cada caso, de igual manera si es posible aumentar el tiempo de entrenamiento del agente los resultados se verían de igual forma mas consistentes.

5 Conclusiones y discusión

En este trabajo se construyó un entorno flexible de aprendizaje reforzado que recrea el transporte de estados en un sistema de 3, 5 o 7 puntos cuánticos. En vista de que el entorno cuenta con los atributos genéricos del framework de OpenAI, cualquier interesado puede poner a prueba nuevos agentes que traten acciones continuas para evaluar sus resultados, contando con la posibilidad de variar fácilmente factores como el refuerzo.
Así mismo, se mostró que el uso de agentes con redes neuronales, en particular de un algoritmo DDPG, permite encontrar soluciones parciales cuyos comportamientos se asemejan a los de las soluciones Ansatz ya conocidas en la literatura para los casos mas simples. El uso de este agente a su vez permite explorar soluciones para problemas con mas puntos y con diferencias en los niveles de energía de los estados.
Durante el proceso, sin embargo, se encontraron algunas limitantes e incongruencias al problema. Los tiempos de entrenamiento para las redes son largos y fácilmente superan tiempos de 3 horas, para redes de 2 capas con ≈ 500 neuronas por capa, usando la configuración con GPU que provee Google Colab. Esto impidió la exploración de redes de mayor tamaño o el entrenamiento de estas para mas de 10.000 episodios. Respecto a las redes acá usadas, aunque su comportamiento se asemeja a lo esperado y algunos casos muestran resultados satisfactorios, es sospechoso el corto tiempo en el que parecen resolver el problema, respecto a las soluciones tradicionales. De lo anterior se sugiere revisar que el entorno esté permitiendo comportamientos que no se corresponden con lo real.
Así mismo, se encontró una falta de robustez en el entrenamiento de los agentes, variar el número de episodios de entrenamiento cambia significativamente el comportamiento de las acciones tomadas y, aunque la aleatoriedad en la inicialización de los pesos de la red implica que pueden haber variaciones en el entrenamiento para condiciones iguales, muchas veces estas variaciones llevaban a resultados notablemente distintos. Se atribuye este hecho a la carencia en tamaño y en número de episodios comentada anteriormente.
Los autores sugieren realizar nuevamente el entrenamiento de los agentes en un hardware con mayores capacidades y hacer una revisión detallada de los parámetros usados en el entorno.

References

Ferraro E., Michielis M. D., Fanciulli M., Prati E., 2015, Physical Review B, 91, 075435

Fösel T., Niu M. Y., Marquardt F., Li L., 2021, arXiv

Greentree A. D., Cole J. H., Hamilton A. R., Hollenberg L. C. L., 2004, Physical Review B, 70, 235317

Gullans M. J., Petta J. R., 2020, Physical Review B, 102, 155404

Guo K., Yang Z., Yu C.-H., Buehler M. J., 2020, Materials Horizons, 8, 1153

Lillicrap T. P., Hunt J. J., Pritzel A., Heess N., Erez T., Tassa Y., Silver D., Wierstra D., 2015, arXiv

Moro L., Paris M. G. A., Restelli M., Prati E., 2021, Communications Physics, 4, 178

Porotti R., Tamascelli D., Restelli M., Prati E., 2019, Communications Physics, 2

Star B., 2013, Supplementary infomation, 22, 0

Sutton R. S., Barto A. G., 2018, Reinforcement Learning: An Introduction, second edn. The MIT Press, http://incompleteideas.net/book/the-book-2nd.html

Weckesser W., 2021, odeintw, https://github.com/WarrenWeckesser/odeintw

Williamson D. J., Burn G. L., Simoncelli S., Griffié J., Peters R., Davis D. M., Owen D. M., 2020, Nature Communications, 11, 1493

Wong D., Yip S., 2018, Nature, 555, 446