Investigation into the robustness of static quantum controls with the introduction of uncertainty

Author: Noor Zehra Zaidi

Student Number: 1706191

Degree: MSc Computing

School: School of Computer Science and Informatics

Supervisor: Frank C. Langbein

Date: 05/11/2021

Contents

Introduction 4
Background 4
Modelling of the System and obtaining controllers 6
Addition of Noise 7
Discussion of Results Obtained 8
Comparing Monte Carlo to QUPID 12
Conclusions and Future Work 14
Self Reflection 15
References 18
Bibliography 18

Abstract

The purpose of this research paper was to investigate the robustness of control parameters obtained using the L-BFGS algorithm (Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm) when uncertainty is introduced to the quantum system. This was investigated using even and odd spin rings modelled with matrix exponentials and using the Wasserstein distances to determine the quality of the results produced by the controller. It was found that the L-BFGS algorithm produce controls that were quite robust and performed well with the added uncertainty however this performance was in part due to the algorithm producing control parameters exceeding realistic values.

During the investigation it was also found that the behaviour of odd and even numbered spin rings is different and that this would be an interesting area for further research.

Acknowledgements

I would like to thank my supervisor Frank C. Langbein for the time he dedicated to helping me with this project and for his guidance throughout the process. I would also like to thank the entire teaching team at the School of Computer Science who helped me get through this year despite the challenges of remote learning.

Introduction

In the field of quantum technology, quantum control theory is a very important aspect as in the case of quantum systems there needs to be some method of manipulation of the state to reach the desired results as unlike classic mechanical systems, quantum systems and their states have many unique features such as entanglement, coherence and superposition and do not always behave as expected and measuring them can be a challenging. Measuring can be a challenge as the measuring of a quantum system can influence the outcome of future measurements. In order for quantum computers and quantum computing to become functional it is important to be able control and manipulate the system into a desired state, such as in the case of manipulating qubits in a ring or chain into the desired states to be able to transfer binary data through the system. (Dong, Peterson 2011)

The purpose of this project is to investigate the robustness of these controllers especially under experimental uncertainty(noise). In particular the algorithm of interest is the L-BFGS algorithm (Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm), and the performance of controllers obtained for both even and odd numbered spin rings. The L-BFGS algorithm it an iterative optimisation algorithm used here to produce static control parameters and it appears to be quite popular and used in multiple research papers. (Ref. 3,5,7) It would be of interest to investigate other popular algorithms such as the Newton-GRAPE algorithm which is also widely used.(Ref. 10,13)

The methodology that is planned to be used is to model the system of qubit rings using matrix exponentials and obtaining control parameters through the L-BFGS algorithm. After the control parameters have been obtained a known value of uncertainty would be introduced into the system Hamiltonian and fidelity plotted against this noise to determine whether the control parameters still perform well under this condition. If the algorithm continues to perform well it would determine that the algorithm is robust and would be suitable for use in experimental work where an unknown value of uncertainty would be present.

Background

Some background knowledge is needed for this paper specifically in relation to spin-1/2 particles. In quantum technology the equivalent to the bit is the qubit which is a 2-state quantum system such as a spin-1/2 particle.

Particles such as neutrons, protons and electrons are all spin-1/2 particles which have an angular momentum of ±1/2 also referred to as spin up or spin down, generally this leads to the particle having 2 spin states which would translate to the states of either 1 or 0 similar to the traditional bits. Spin-1/2 particles may also have a state that is a linear combination of these 2 states. (David Griffiths) The state of a qubit or quantum 2 state system can be written as

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$

Where $|0\rangle$ and $|1\rangle$ correspond to the states 0 and 1 for a classical bit. (Dong, Peterson 2011)

For the purpose of this project, we will be looking at a ring of coupled spins, nearby spins may interact with each other through exchange coupling and the energy of spin states may differ by small amounts which leads to the Hamiltonian of this network taking the form of the following equation.

$$H_{full} = \sum_{n=1}^{N} \Delta_n Z_n + \sum_{m \neq n} J_{mn} [X_m X_n + Y_m Y_n + \kappa Z_m Z_n] \quad (1)$$

In the case of XX coupling which we are using in this paper $\kappa = 0$. Xn, Yn and Zn are operators acting on the Hilbert space of the N-spin network using the X, Y and Z operators which are the Pauli spin operators.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \; Y = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2)$$

If we assume that there is uniform coupling, we can set J=1. For the single excitation subspace, the Hamiltonian is reduced to the following equation.

$$H = \Delta_n |n\rangle\langle n| + J_{mn} |m\rangle\langle n| \quad (3)$$

Where $|m\rangle\langle n|$ is a matrix which is zero except for a 1 in the (m,n) position. $J_{mn} = 0$ unless $m = n \pm -1$ and $J_{N,1} = J_{1,N} \neq 0$. (CDC 2015)

For the script written for the purposes of this paper XX coupling and a single excitation subspace are used. For a single excitation subspace and a ring of size N with XX coupling the Hamiltonian matrix becomes the circulant matrix shown in figure 1. (Jonckheere, Langbein, Schirmer 2015) A single excitation here refers to only a singular state moving around the ring or in a more classical terms a single bit changing such as going from 10000 to 00100.

$$\bar{H}_N = \begin{pmatrix} 0 & 1 & 0 & \dots & & 0 & 1 \\ 1 & 0 & 1 & & & 0 & 0 \\ 0 & 1 & 0 & \ddots & & 0 & 0 \\ \vdots & & \ddots & & & & \vdots \\ & & & & \ddots & & \\ 0 & 0 & 0 & & \ddots & 0 & 1 \\ 1 & 0 & 0 & \dots & & 1 & 0 \end{pmatrix} (=: C_N),$$

Figure 1 *Matrix of the Hamiltonian subspace of a spin ring of size N. (Jonckheere, Langbein, Schirmer 2015)*

The biases of the system would be values on the diagonal from the top left to the bottom right. The biases would be used as the control parameters for the system in later parts of this paper. In this investigation we are using the fidelity of the system to determine the robustness of the controllers. The fidelity of the system is essentially the probability of it being the desired outcome, that is how close to the desired outcome is the measured result, 1 being the ideal fidelity of a transfer. The sum of fidelity and infidelity is always 1.

Modelling of the System and obtaining controllers

As a starting point of the investigation a coding language had to be chosen out the following languages, python, C, C++ or MATLAB as these languages were of interest to the project supervisor. Due to prior experience with the language python was chosen, though it would have been quite valuable to have learnt a new programming language such as C++ as this would have been a useful skill to have.

In this investigation the quantum system used was a ring of coupled spins also known as a spin ring. A seven and eight spin ring were used, the system itself was modelled with matrix exponentials and the controllers were obtained using the LBFGS-B algorithm from Scipy using the same method as that used in the 2015 paper from Jonckheere et al.(CDC 2015) as modelling the system was not part of the investigation so it seemed best to follow the method from an existing paper rather than spending too much time trying to create a new version and this code was itself written in python which was the language chosen for this project.

Initially we input the quantum spin network in the form of a matrix and ensured that it remained symmetric. The next step after this was to give the desired input and output in the form of a string which was then checked in the next step to ensure the number of excited states between the two were the same. When obtaining controllers initial values of time and the biases were input, the min and max time input were 0 and 200 while the min and max biases were -10 and 10. The biases were kept at this value to make sure they were realistic values rather than absurdly large numbers.

The next important step was creating the control Hamiltonians. This was done by creating matrices of the same dimension as the Hamiltonian and ensuring that one of the values on the diagonal was 1, essentially for a network of 8 nodes 8 control Hamiltonians would be made covering each of the positions on the diagonal which is shown in the code snippet in figure 2.

```
# control Hamiltonians
CC = []
for k in range(0,Nspin):
    CM = np.zeros((Nspin,Nspin))
    CM[k,k] = 1
    CC.append(CM)
```

Figure 2 *Code snippet showing the control matrices before the addition of the bias values.*

A function is written for which the results returned are the gradient and infidelity(error). The input for this function is an array containing values for the biases as well as a value for the time. This function is also where the biases will be added to the initial system Hamiltonian. This function was provided from a piece of code provided by the supervisor for this project that was from the CDC 2015 paper.

For the optimisation random values of the biases and time are produced using the initial limits that were input earlier in the script and this array of the biases and time is then used along with the gradient function in the L-BFGS algorithm from SciPy. (LBFGS, SciPy Docs)

After each optimisation loop if the infidelity is below 0.01(which means the fidelity is above 0.99) the result is saved into an array where each result contains the infidelity, time, biases, initial time, and initial biases.

For this investigation the transitions within the rings were done for the first to last nodes and for the first to middle(fourth) nodes. When doing so the algorithm was run 1000 times but only controllers producing a fidelity of 0.99 or more were saved to a file for use in further investigations with noise.

System	Transition	Controllers Obtained
7 spin ring	1 to 4	3
7 spin ring	1 to 7	7
8 spin ring	1 to 4	2
8 spin ring	1 to 8	7

Table 1 *The number of controls obtained for each transition and system.*

It can be seen very few controllers with high fidelities were obtained considering 1000 optimisations were run by the script, it can also be seen that transitions to the middle node seem to produce less controllers indicating this may be a more challenging transition to achieve.

Addition of Noise

When adding uncertainty to the system the noise was added to all the non-zero values in the original Hamiltonian, the noise was not added to the bias values. The noise added to the system had a max value of 0.1 as the non-zero values within the Hamiltonian were a value of 1 and 10% noise seemed like a reasonable amount of background noise that would likely be present in real systems. The noise was also uniformly sampled from points on an n-sphere to add a directional component. (Extreme Learning 2020)

Adding the noise to the non-zero values was the final approach used but along the way to deciding on this approach many other methods were trialled such as adding the noise to the Hamiltonian before running the script to obtain the controllers. This approach was not used as after running the script and looking at the infidelities it was realised that this method would not check the robustness of the controllers but check whether the algorithm could deal with noise in the system. For work with real systems, it would be more likely that a researcher would obtain controls for a mathematical model of the system and that during experimental work there would be unknown levels of uncertainty within the system. Another scrapped approach was adding uncertainty to all the non-zero values including the biases however this did not seem to be the best approach as the control parameters should remain static and not be influenced by system noise.

To determine whether the controllers remained robust under noise some measurement or parameter would be needed to determine the performance of the controller. Initially there

was some thought towards deciding on a quality value or using the mean and seeing the percentage of values obtained above that number however upon testing this method it was not ideal as the quality value would be subjective and did not seem to have any scientific value as it could easily be too low and make controllers seem more robust than they are. In the case of using the mean this would be a different value for each controller and thus would not be consistent enough to compare the results. It was decided that the Wasserstein distance would be used as there was a function for this within SciPy and it would work consistently between the controllers and systems and would be comparing the obtained fidelities against the ideal value of 1 and so would not be affected by human error the way a quality value could be. The Wasserstein distance is a measure of the distance between two data sets generally in the form of the work required to get from one set to another. (Wasserstein Distance, SciPy Docs)

Discussion of Results Obtained

	Fidelity	Wasserstein Distance
Controller 1	0.9997398390189348	0.0277487802038016
Controller 2	0.9957399630842981	0.0206150769581797
Controller 3	0.9935541141659456	0.00723685741997269
Controller 4	0.9927119370485148	0.00832322452946485
Controller 5	0.9925895657767727	0.00190406137386582
Controller 6	0.9912740845679344	0.0103764014116135
Controller 7	0.9912437737114035	0.017674921141374

Table 2 *Fidelities and Wasserstein Distance for 7 spin ring transition 1 to 7.*

In this table the Wasserstein distance of the controllers are small with the smallest obtained being approximately 0.0019 and the largest being approximately 0.0277. These values correspond to controller 5 and 1 respectively.

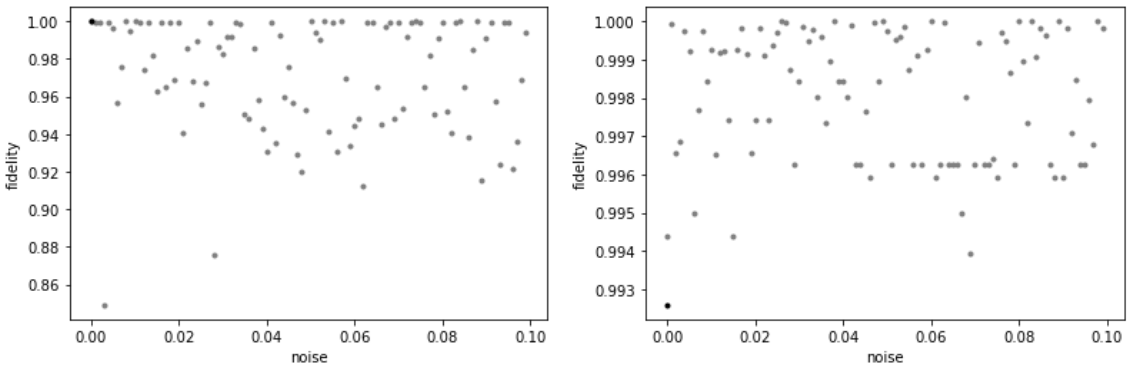

Figure 3 *Fidelity versus noise scatter plots for Controller 1(left) and Controller 5(right).*

In figure 3 a side-by-side comparison of the fidelity versus the noise for controller 1 and 5 respectively is shown. The black dot in each scatter plot represents the fidelity value of the controller when it was produced via the L-BFGS algorithm. As can be seen in the figure the values of the fidelities for controller 1 seem to follow a trend of getting lower with the increase in noise and it can be seen the scatter points seems to form a negative gradient. With controller 5 however the fidelities seem to increase with the addition of noise and are

generally higher than the initial value. The points seem to have a linear spread indicating consistency throughout the addition of noise which would indicate these are robust control parameters.

Both controllers had biases that were within the range of -10 to 10 however one of the bias values for controller 5 was 14.33166459. While this is higher than the limit set there is not enough evidence to indicate that this might have been the cause for the better performance of this controller as no other controllers for this system transition had higher values so there was not enough data with which to interpret whether this hypothesis is true or not.

	Fidelity	Wasserstein Distance
Controller 1	0.9997360713719673	0.0002639292571180118
Controller 2	0.9997359846604862	0.00026402809531753666
Controller 3	0.992404381769754	0.11384390454995771

Table 3 *Fidelities and Wasserstein Distance for 7 spin ring transition 1 to 4.*

This transition only produced 3 controllers with high fidelity values which seems to indicate this transition from node 1 to 4 may have been more difficult than a transition from node 1 to 7. Overall, the Wasserstein distances for controllers 1 and 2 are quite similar and very small at approximately 0.000264 indicating the controllers maintained high fidelities throughout, however there is quite a difference in the value of controller 3 at approximately 0.113844 which is relatively much larger and also the largest distance obtained overall among the controllers for all systems and transitions.

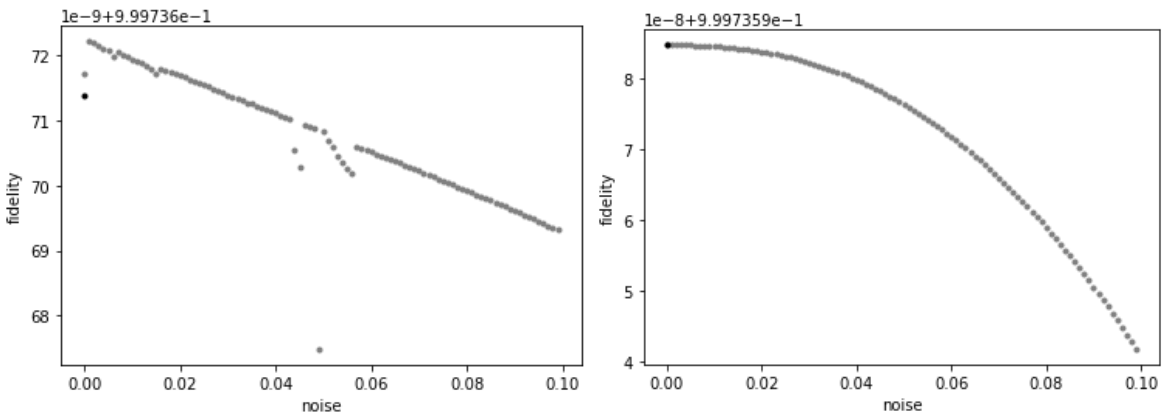

Figure 4 *Fidelity versus noise scatter plots for Controller 1(left) and Controller 2(right).*

Figure 4 shows controllers 1 and 2 side by side and it can be seen both seem to have a downward trend and seem to be more of a continuous line rather than a scatter. Controller 2 seems to have a smooth transition with the fidelity getting exponentially worse with noise. Controller 1 shows a linear trend but with sudden downward spikes occurring at intervals. It would be of interest to investigate this further, both controllers did have one interesting fact in common which was that both have biases with extremely high values in the hundreds and thousands. This bias value is unrealistic and would likely not be used for a real system however it may be of interest to investigate the structure of these matrices and see if there is some difference between them which is causing the spikes within the plot of controller 1

however there was not time to further investigate this for the project and the mathematical analysis was out of scope for my skill set. These high bias values producing good results may show some indication that a similar phenomenon is occurring for controller 5 in the node 1 to 7 transition.

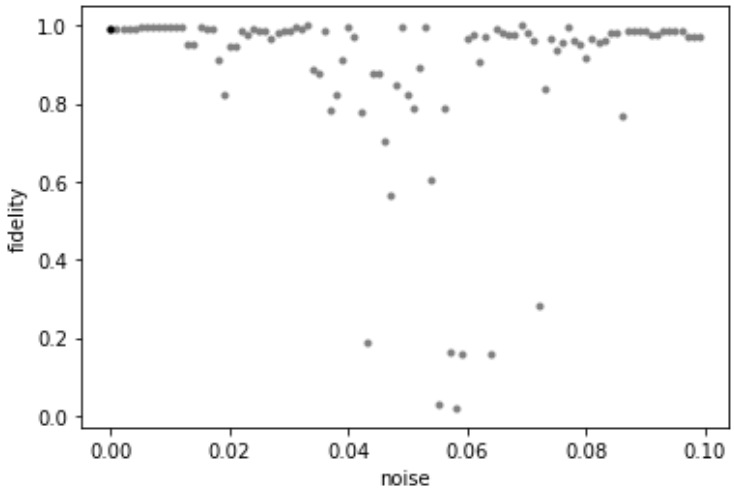

Figure 5 *Fidelity versus noise scatter plots for Controller 3.*

Controller 3 did have a larger Wasserstein distance but the scatter plot for this controller shown in figure 5 is quite interesting as it too seemed to be continuous rather than a scatter but shows more of a scatter pattern between a noise value of 0.03-0.07. Currently it is unknown why it seems to break down at these values, but this may be further investigated and may also be related to the transition of the system itself or the generally low bias values. A lack of data prevents this from being investigated further, more controllers would need to be obtained in order to see if this trend occurs in multiple controllers or if this was a one off, perhaps it might have been prudent to have run 5000 optimisations runs rather than 1000 as it may have produced more controllers to work with.

	Fidelity	Wasserstein Distance
Controller 1	0.9996138130160255	0.00043349185453563365
Controller 2	0.998943537101543	0.00108423951625186
Controller 3	0.997631334739837	0.00191967071179269
Controller 4	0.994802737032047	0.00520662731212359
Controller 5	0.994503332301303	0.00556823553319401
Controller 6	0.99040003410457	0.0104743465100929
Controller 7	0.990212380369828	0.0107489029607523

Table 4 *Fidelities and Wasserstein Distance for 8 spin ring transition 1 to 8.*

For the transition in this system the best and worst Wasserstein distances seemed to be those of controllers 1 and 7 respectively. With controller 1 again demonstrating biases outside the initial range set of -10 to 10 so again this controller would likely not be useful for any future work as it has unrealistic values, however these biases did not reach into the hundreds or thousands so it may still be useable. In the scatter for controller 1 in figure 6 it can be seem that the controller has a smooth continuous pattern which so far seems to be a

common feature for controllers with biases that do not fall within the minimum and maximum values initially set when obtaining controllers.

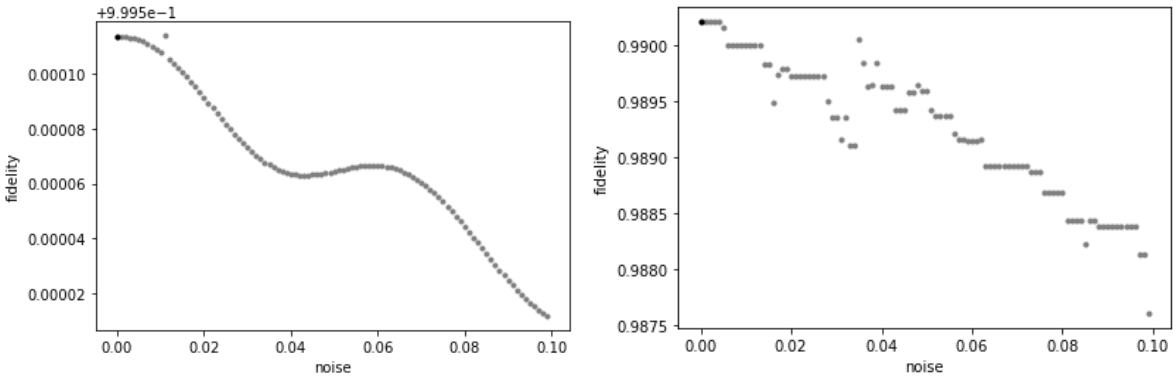

Figure 6 *Fidelity versus noise scatter plots for Controller 1(left) and Controller 7(right).*

The scatter for controller 7 does seem very interesting. The pattern for it looks to be level based with gaps between the continuous spread of the steps. Looking at the figure at a noise value of around 0.03 there appears to be some sort of spike at the transition between these levels, however the spike does not appear elsewhere in the figure. It does look as if there might be 2 different lines intersecting, where one is a step function and the other is more sinusoidal, however no other controllers had the potentially sinusoidal line.

Overall, this system's transition is quite interesting as controllers 2, 5 and 6 also seem to show steps where the fidelity is constant. It is possible that controller 1 is exhibiting similar behaviour but due to the high biases is showing smooth transitions. Perhaps this sort of level system is a property of even numbered spin rings as they may have different behaviour compared to odd numbered rings.

	Fidelity	Wasserstein Distance
Controller 1	0.990212380369828	0.00779440225783707
Controller 2	0.992236248878016	0.00883106057942251

Table 5 *Fidelities and Wasserstein Distance for 8 spin ring transition 1 to 4.*

For this transition only 2 controllers were obtained both showing low Wasserstein distances. In terms of the biases, controller 1 showed very high bias values within the thousands while controller 2 had biases within the -10 to 10 limits, however despite this disparity both controllers show similar values with an approximate 0.001 difference. This shows that the controllers obtained may still be fairly robust even when using bias values within the set range.

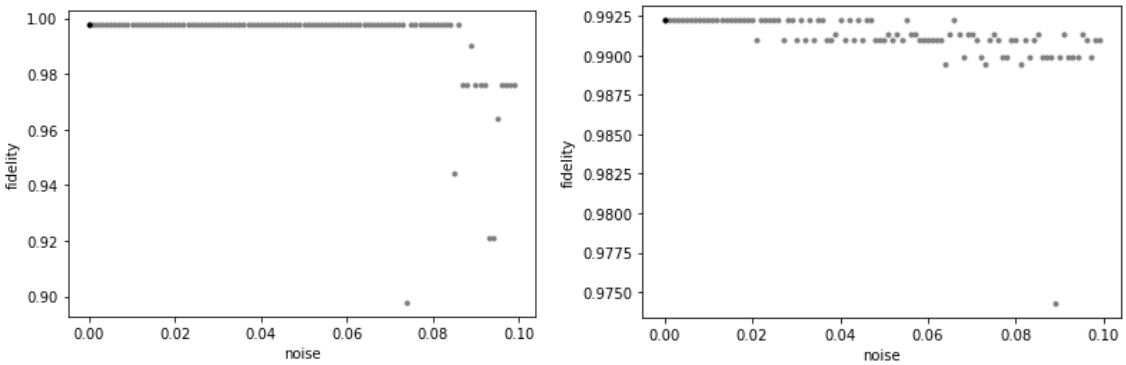

Figure 7 *Fidelity versus noise scatter plots for Controller 1(left) and Controller 2(right).*

In the figure both have a scatter pattern that shows levels which may support the earlier hypothesis that this is a property of even numbered spin rings, however it seems that controller 1 from the node 1 to 8 transition does not follow this pattern. Previously within this discussion it had been suggested that the smooth shape of the plot of controller 1 in the 1 to 8 transition was due to the high bias values, this suggestion seems to have been proven false as in this transition controller 1 does not show a smooth continuous shape despite the high bias values but this controller does have bias values in the thousands so they may not be comparable due to the previous one not having as high values.

Overall, the controls produced by the L-BFGS algorithm seem to be robust as overall 12/18 controllers had quite small values that were in the 3rd decimal place. The small Wasserstein distance does indicate that the values of the plot were not far off from the ideal fidelity value of 1. Throughout this discussion it has been made clear that it would have been desirable to have been able to have obtained more controllers however the optimisation algorithms for obtaining them does take quite a long time to run and initial stages of this project involved figuring out how to obtain and process data so not enough time was left to run the script for long periods. The data obtained does also indicate that there is unique behaviour in the even spin ring compared to the odd one, perhaps for future research more ring sizes could be used to investigate this.

Plotting the fidelity versus noise also took a long time for each of the controllers due to it being repeated many times using the Monte Carlo sampling method. In a new paper from Dalgaard, Weidner and Motzoi titled Dynamical Uncertainty Propagation with Noisy Quantum Parameters, they proposed a new method called QUPID to compare to Monte Carlo sampling.

Comparing Monte Carlo to QUPID

In the paper from Dalgaard et al. 2021 they proposed a modelling method to compete with Monte Carlo simulations. The purpose of their paper was to simulate systems with the addition of experimental uncertainty also known as noise. In the paper they mention how Monte Carlo sampling is the industry standard but also quite slow and that their method would show a significant speed up in comparison.

The method requires using an equation involving the evolution of a system, while the other variables in the equation would be replaced with a set of normally distributed parameters $\theta = (\theta_1, \theta_2 \dots \theta_M)$ with mean $\bar{\theta} = (\bar{\theta}_1, \bar{\theta}_2 \dots \bar{\theta}_m)$ and covariances $\sigma_{i,j} = \mathrm{E}[(\theta_i - \bar{\theta}_i)(\theta_j - \bar{\theta}_j)]$ which are assumed to be relatively small. The function is Taylor expanded around the mean and due to the small covariances is it truncated at the second order. In their paper they found that their prediction based on the QUPID method produced the same results as the Monte Carlo sampling and while the Monte Carlo sampling took approximately 30 minutes to run the QUPID method only took approximately 0.2 seconds to run. (Dalgaard, et al. 2021)

Based on the result of this paper it seemed worthwhile to attempt to use this method as it could be time saving and allow for faster research in the future as Monte Carlo sampling would not be needed. However, in the case of the system for this paper the equation for the system Hamiltonian under noise was essentially the Hamiltonian matrix multiplied by 1 plus the noise.

$$H_{noise} = H \times (1 + noise \times delta) \tag{4}$$

Where the delta was the directional component of the noise from spherical sampling. The equation itself is basically just a polynomial and the Taylor expansion of a polynomial is the polynomial itself which seemed a bit too simplistic. This also led to further issues of trying to figure out what exactly the normally distributed parameters should be as the delta was already a normally distributed value.

In the end assuming that delta was the only parameter that was normally distributed and plotting the Monte Carlo sampling as a box plot a graph was obtained however the results did not match the paper very well.

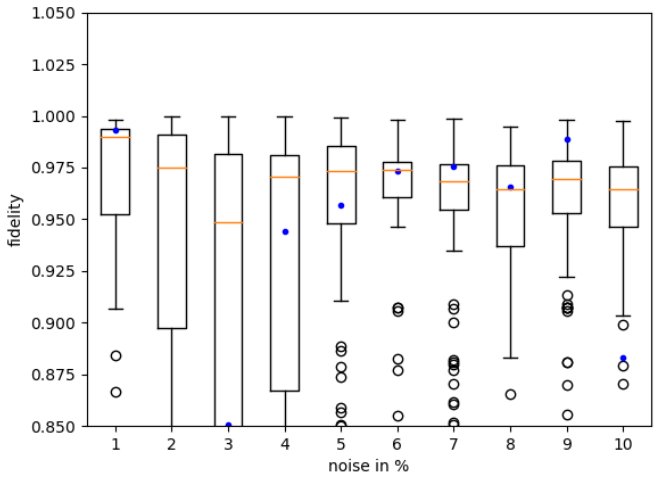

Figure 8 *Box plot of Monte Carlo sampling versus the QUPID method (blue dots).*

When attempting to run this method the points did not all fall within the error bars of the box plot, they were also not noticeably close to the mean values and were not close to the individual points that make up the box plots. Very likely this is due to my own inability to

figure out how to use this method and what parameters I should have been using. It is possible that it might have been the biases that needed to be turned into normally distributed parameters however that also does not seem to make sense as the bias values are the controls we are testing and should remain static. There is potential that due to this being a relatively new method that it simply does not work however based on my current attempts to use this method it is inconclusive.

Conclusions and Future Work

After this investigation I believe that the L-BFGS algorithm is a robust algorithm that performs well when introduced to uncertainty based on the small Wasserstein distances of the results obtained compared to the ideal values. However, it does seem that some of the reason for this algorithm performing very well is the fact that in its optimisation it will produce bias values that greatly exceed what is realistic. This may not have been an issue if these large values were close to the initial limits set, though majority of the best performing controllers had bias values in the hundreds or thousands. It would be of interest to look into whether a different package or a different programming language would have had an implementation of the algorithm where limits could be set on the optimization values the algorithm would output. Though the minimum and maximum bias were set in the script, they were used to produce the initial values and the algorithm itself did not seem to have a fixed range, it would just iterate until it either reached a good result or got trapped and ran out of attempts. If this was possible it probably would have led to an overall better investigation as there would have been more experimentally useful controllers rather than ones that were robust but not realistic. If it were possible to limit the bias range in the algorithm there may have been even fewer controllers obtained in 1000 optimisations however with this knowledge a larger number of optimisations could be run early on for the purpose of a new investigation.

Due to time constraints other algorithms were not touched upon but it would have been interesting to perform a similar investigation for another popular algorithm such as the Newton-GRAPE algorithm. It would have also been interesting to attempt this with a quantum modelling library such as QuTiP which contains many premade control algorithms and models. (QuTiP documentation)

It also seems based on the investigation that the behaviour of even and odd spin rings is different from one another based on the graphs of the even spin rings displaying what seemed to be step function rather than scatter plots of continuous plots. It very likely has something to do with the structure of the system, possibly due to its symmetries and the interactions of the spins in this system. Perhaps in future this could be investigated as a separate paper using more spin ring sizes.

With more time I would have also liked to have further investigated the QUPID method and to have figured out how to implement it for this system with noise, specifically for the fidelity versus noise. The method seems to have been intended for time dependant evolutions so it is likely that it would not work for what I attempted to use it for and may work well for a different research topic. The method itself sparked my interest due to the

idea of it being a time saver. Many aspects of this project required leaving scripts running for long periods of time and they would also often jam and need to be stopped and restarted so it would have been very valuable to have gotten more done in less time. However, I could not obtain any conclusive results as to whether this method would have made things faster or not for my research.

Self Reflection

In the initial stages of the project, I spent a lot of time reading trying to figure out what direction I wished for my project to go as the initial project brief was very open and could be taken in many directions. In my research reading papers I did notice that many papers in this field are very recent as it is a constantly evolving and newer field of research and this itself came with many challenges. One of the challenges of this being an evolving field is that some older papers that were from the 2010s may possibly be outdated by newer research coming in, also as this is a newer field the research itself was harder to understand for me and it was hard to find any textbooks to help understand the concepts. In terms of my background, I studied physics but did not have a large focus on theoretical physics or quantum physics beyond very basic level understanding of spin-1/2 particles, then having done the taught portion of my MSc Computing I did not have a great base in computer science theory either as the course did focus more on industry and practical work over theory. A significant mistake that I had made was underestimating my own lack of knowledge, so a lot of my initial project time was spent reading and trying to understand the research area and see what topics were of current interest, it would have been helpful if I had known or understood that I had so little knowledge and needed to start researching earlier than the official start date.

One of the topics that seemed to be of interest was quantum control theory and robust quantum controls so that ended up being the direction I took my project into. In doing so I was met with new challenges such as trying to figure out how to model the system but as it was not the main focus of my research my supervisor gave me a lot of guidance and helped provide some initial code which was very helpful and prevented me from spending more time with my initial set up of plans.

During my literature review I also found that many author names were repeated across papers I read especially in more recent research and many of those authors seem to work together so there may be some initial bias in my work as it was based on papers that were written by the same people or from a group of researchers who work together often and likely share similar views. This may have been due to my not reading enough papers or not having a varied enough search which would be a weakness in my approach. It may also have been due to the field being considerably new and niche meaning there is less variety of views to be found.

Another thing I noticed during the literature review is that authors do not often include the code for their research paper which overall makes it a lot harder to understand how to implement the mathematics in the form of code. Some researchers do have their code available elsewhere but not always linked to the paper. It does seem as if the reason for this

may be to avoid having other people point out their mistakes but in general it was hurdle and made understanding and researching the topic much harder and less approachable as someone new to this research area.

With respect to my methodology and project management I found that I did not manage my time as well as I should have. Due to the fact that this research area was completely alien to me I spent a lot of time reading, including papers that did not end up being relevant to my final work and perhaps some of this was due to bad decision making on my own part by not choosing a topic I was more familiar with and for not starting my literature review early on before the start of the official dissertation period. I feel I also made some bad time management decisions as I started a full-time job and moved across the country during the dissertation period and did not take these events into consideration during my planning and ended up feeling that I had spread myself too thin. A different time related issue was related to a personal weakness I have which is a habit of getting tunnel vision. I would start trying out an approach or method but would get too invested in it which would prevent me from looking at it objectively that would then cause me to spend a lot of time on it before coming to realise that the method was completely wrong. An example of which was the period where I was attempting to find controls after adding noise which did not add any value to the project and was a very slow process itself.

My final methodology of adding in noise to the system after obtaining controls so far seems to be a good choice as realistically one would obtain them from a mathematical model before using them for a real system and the controls themselves would be static, only the system itself would be experiencing the noise. The Monte Carlo sampling that I used was an acceptable method that is widely used in many areas of research and that I had previously used many times for projects during my undergraduate degree. However, this method is quite slow so I do think it was a valid decision to look into the QUPID method to speed things up, however in the end I could not figure out how to get it to work for the approach I was using and did manage to spend more time on this rather than actually saving time.

I think overall I have come out of this research project realising that an exceptional weakness I have is my tunnel vision and my time management. I have realised that what I need to do is take more frequent smaller breaks from my work to be able to look at it from an objective lens rather than investing a lot of time into a method that is not working and refusing to be flexible. I also need to consider that there is only so much time in a day and I cannot do as much as I want to within one day and need to take into account other commitments I may have. I think this project has helped me grow to realise that I need to consider my own capabilities and mental health when undertaking and planning my projects and that I need to invest more time in prep work for unfamiliar areas as otherwise I run into many roadblocks and cannot easily see past them. Despite my weaknesses which I need to work on and overcome I would say I do have a few strengths; I did find that my ability to code in python had vastly improved throughout the taught portion of my course and I was able to understand and interpret other people's code much better throughout the project period. My mathematical skills while not at their strongest were still useful to me throughout the project when reading papers as there was a heavy focus on mathematical

equations and proofs. I also found that I do really enjoy learning about new topics that I have little experience with however I do struggle to apply what I learned or focus my learning onto relevant research. If I had to repeat the project, I would have been more flexible and might have also tried to use some quantum modelling libraries instead as these had more options for optimisation algorithms than standard python libraries.

I really enjoyed the whole process of this project work, the literature review and discussions with my supervisor were a highlight of the process. Overall, as this was a heavily research-based project, I did not end up with a tangible product. I did enjoy the whole process all the same and felt that it helped me discover some of my significant weaknesses.

References

1. David Griffiths, Introduction to Elementary Particles.
2. D. Dong, I.R. Petersen, 2011, Quantum control theory and application: A survey
3. CDC 2015, E. Jonckheere, F.C. Langbein, S.G. Schirmer, 2015 Time optimal information transfer in spintronics networks.
4. E. Jonckheere, F.C. Langbein, S.G. Schirmer, 2015 Information transfer fidelity in spin networks and ring-based quantum routers.
5. E. Jonckheere, F.C. Langbein, S.G. Schirmer, 2016, Design of Feedback Control Laws for Information Transfer in Spintronics Networks.
6. Extreme Learning 2020, http://extremelearning.com.au/how-to-generate-uniformly-random-points-on-n-spheres-and-n-balls/
7. I. Khalid, C.A Weidner et al. 2021 Reinforcement Learning vs. Gradient-Based Optimisations for Robust Energy Landscape Control of Spin-1/2 Quantum Networks.
8. LBFGS, Scipy Docs, https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fmin_l_bfgs_b.html#scipy.optimize.fmin_l_bfgs_b
9. M. Dalgaard, C.A. Weidner, F. Motzoi, 2021, Dynamical Uncertainty Propogration with Noisy Quantum Parameters.
10. M. Dalgaard, F. Motzoi et al. 2019, Global optimization of quantum dynamics with AlphaZero deep exploration.
11. QuTiP documentation and tutorials, https://qutip.org/docs/4.0.2/guide/guide-overview.html , https://qutip.org/tutorials.html
12. Wasserstein Distance, Scipy Docs, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wasserstein_distance.html
13. X. Yang, C. Arenz et al. 2021, Assessing three closed-loop learning algorithms by searching for high-quality quantum control pulses

Bibliography

1. C. Moler, C.V. Loan, 2003, Ninteen Dubious Ways to Computer the Exponential of a Matrix, Twenty-Five Years Later*. *This was an interesting paper talking about the issues that arise with computing matrix exponentials, this was a follow up to an older paper which I was unable to find. This was an interesting read recommended by my supervisor.*
2. D. Dong, I.R. Petersen, 2011, Quantum control theory and application: A survey. *A paper describing many different types of quantum control methods such as open loop and closed loop control as well as strategies such as optimal control, variable structure control and quantum incoherent control. One of the papers I initially read that made me consider focusing on control as my research topic.*
3. QuTiP documentation and tutorials, https://qutip.org/docs/4.0.2/guide/guide-overview.html , https://qutip.org/tutorials.html . *A useful link for learning more about quantum modelling and also a very useful python library for anyone with interest in this research area.*