ELEC4605 Assignment 1

Ronson Chan

Z5165499

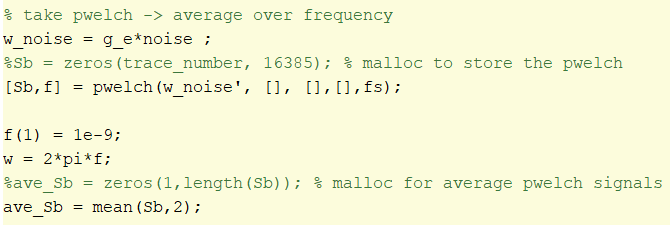
**Question a)**

Use filter-function theory to plot the decay of coherence of the spin ensemble due to this noise for the following experimental pulse sequences:

1. Ramsey free induction decay
2. Hahn echo
3. Dynamical decoupling

Before we get into individual pulse sequence, let us look at the setup, specifically obtaining the power spectral density.

We are given a 100x80,000, with 100 spins with each spin having a total time of 0.2seconds, with a sampling time of 2.5 microseconds.



We first convert the noise signals into units of Hertz by multiplying the noise signal with the gyromagnetic ratio ), and input it into Matlab function “pwelch”, which outputs the power spectral density of the input noise, in frequency domain.

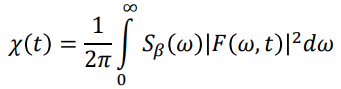
Figure 1 shows the power spectral density of the noise in logarithmic scales.

Chart, line chart

Description automatically generatedFigure(1)

We decide to take the average in the frequency domain to obtain an average power noise spectrum since averaging in the time domain will average out the noise signal, which is vital to our analysis. We then have defined to be our frequency ranging from 0 to the sampling frequency (40kHz).

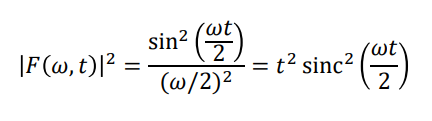
The Decoherence function is given by the following formula:



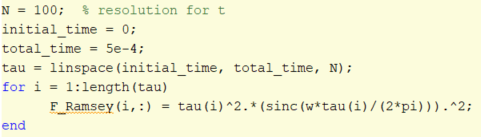
In this assignment we have decided to replace the infinity to our sampling frequency since this is sufficient to act as our limit of integration. For each individual pulse sequence the filter function is different. In the following section we shall investigate individual filter functions for each pulse sequence, and their coherence decay function:

**Question ai)**

Ramsey Free induction decay gives the following filter function:



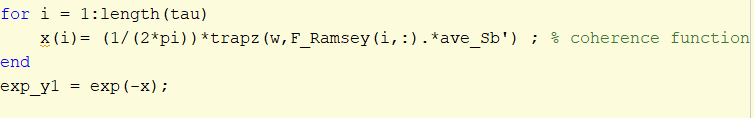
Figure(2) and Figure(3) shows the code for generating the Ramsey filter function and its corresponding plot:

Figure(2)

Chart, histogram

Description automatically generatedFigure(3)

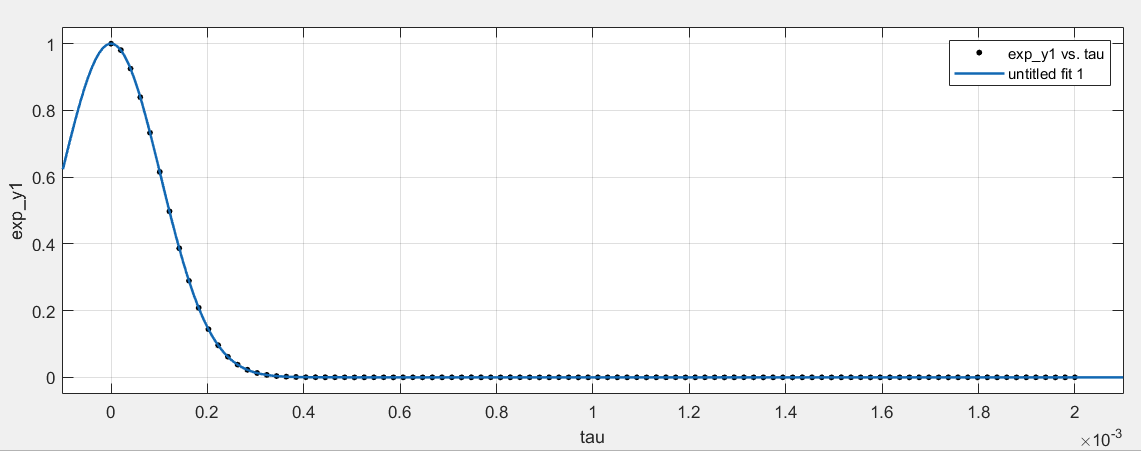
Lastly we will integrate the product of our power spectral density with the filter function over the range from 0 to the sampling frequency, which is performed with the code in figure(4), and figure(5) shows the decoherence function against total pulse length of Ramsey.

figure(4)

Chart

Description automatically generatedfigure(5)

In order to extract the decoherence time , which corresponds to the decoherence decaying to 1/e of its initial peak, we will fit the above data points with an appropriate function. The following is the optimized curve that explains the behavior of Ramsey decoherence.

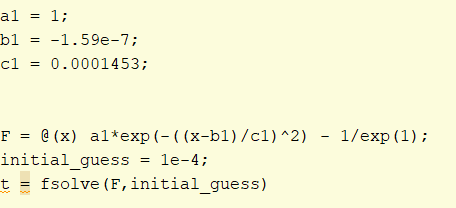
Figure(6)

With figure(6) describing the gaussian equation:

|  |  |
| --- | --- |
| a | 1 |
| b | -1.59e-7 |
| c | 0.0001453 |

With R-squared value = 1.00

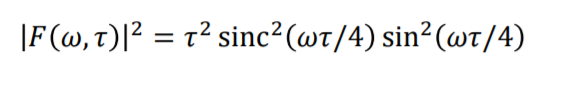
We can extract the by substituting .



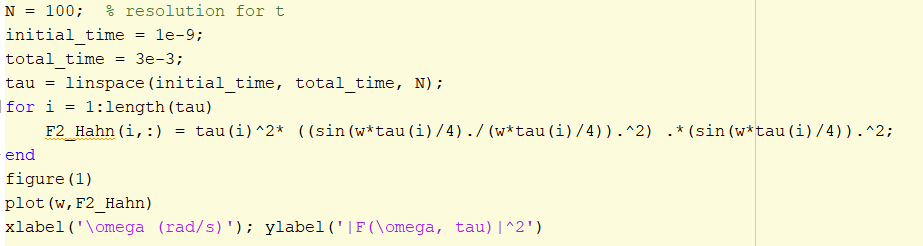
With the code above we can find that .

**Question aii)**

For our second pulse sequence of interest, the Hahn Echo, gives the following filter function:



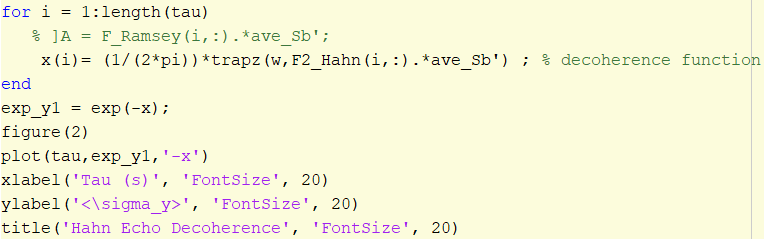
In figure(7) describes the code that generates the Hahn echo filter function and the following filter diagram:

figure(7)

Chart, line chart, histogram

Description automatically generatedfigure(8)

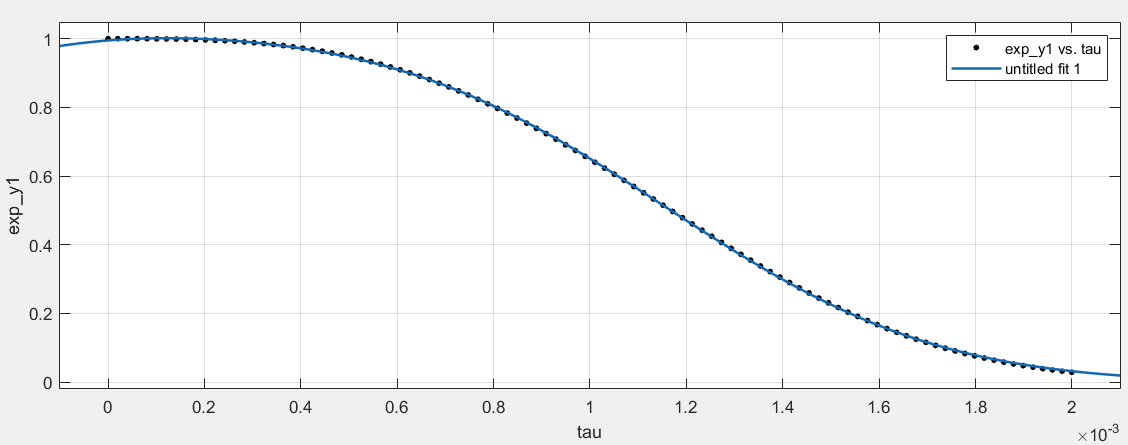
Similar to the Ramsey experiment, we integrate from 0 up to the sampling frequency of the product between the Noise spectral density and the filter function. Figure(9) shows the code corresponding to such numerical integration, and figure(10) shows the Hahn Echo decoherence.

figure(9)

Chart, line chart, histogram

Description automatically generatedfigure(10)

The same procedure from Ramsey, we input the above data in a curve fitting software in Matlab in order to extract the time.



With the Gaussian function

where

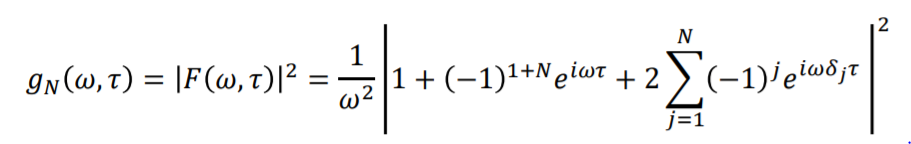
|  |  |
| --- | --- |
|  | 0.8986 |
|  | -0.0001478 |
|  | 0.0009442 |
|  | 0.4762 |
|  | 0.0008206 |
| \_ | 0.0006947 |

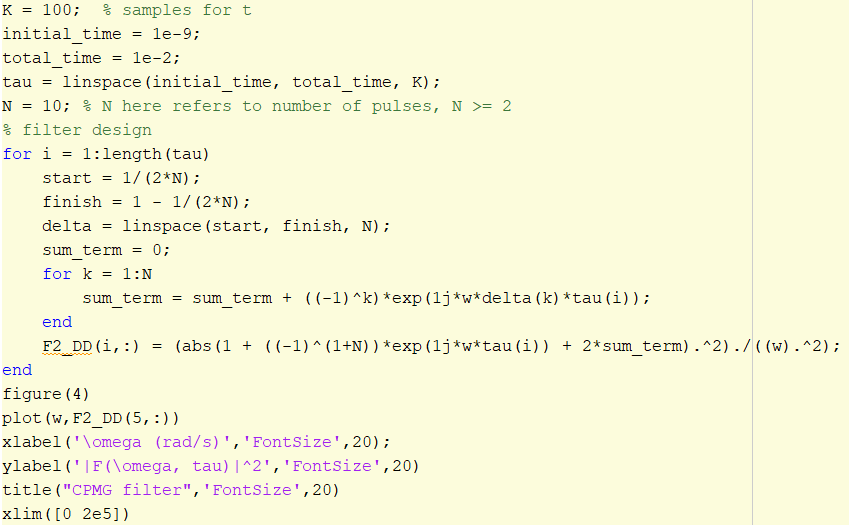
With an R-Square value = 1.00

Using fsolve we can find the , which is approximately 10 times greater than .

**Question aiii)**

Lastly for our Dynamical decoupling gives the following filter function that varies with N, the number of π pulses about Y. Figure(11) shows the code for generating the filter function for dynamical decoupling and the filter function itself:



figure(11)

With the following filter plot of 1 pulse, 5 pulses, 10 pulses and 20 pulses.

Chart, histogram

Description automatically generatedfigure(12)

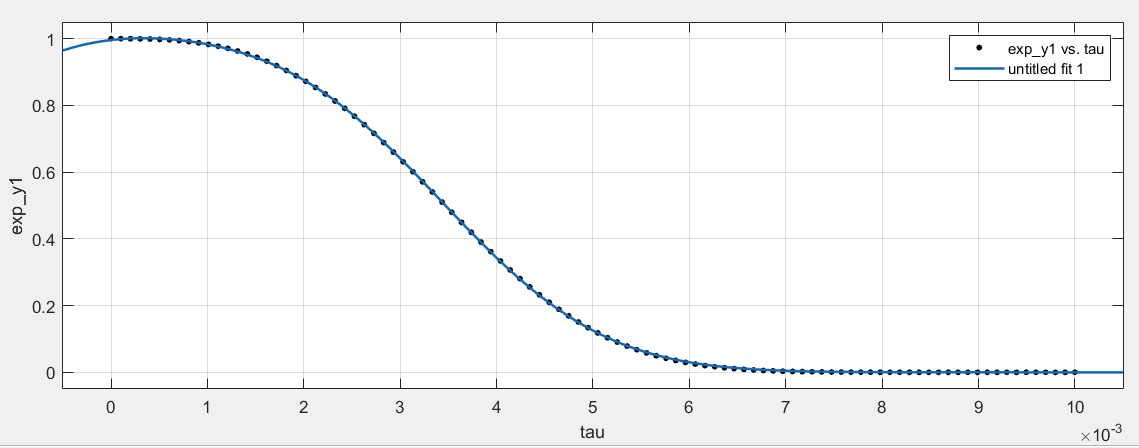
We then integrate up to the sampling frequency for the product between the noiwer power spectrum and the filter function and obtain our decoherence for N = 1, N = 5, N = 10 and N = 20 as illustrated in figure(13):

Chart

Description automatically generatedfigure(13)

We can see from figure(13) how the sampling frequency of the filter is proportional to the number of pulses. The higher number of pulses, the filter will be more inclined to sampling at higher frequencies.

At 1 pulse, CPMG dynamic decoupling behaves like a Hahn Echo sequence (the filter function has different amplitude between Hahn and dynamical decoupling, which explains the offset), and as we increase N, the increases with diminishing effects. Take N = 5 pulses as an example for curve fitting example in figure(14):

figure(14)

We get the following equation:

Where

|  |  |
| --- | --- |
|  | 0.8923 |
|  | -0.0005062 |
|  | 0.002805 |
|  | 0.4922 |
|  | 0.002404 |
|  | 0.002907 |

With R-Square value = 1.00

From the above equation and using fsolve we can extract the

Briefly comparing the values for :

|  |  |
| --- | --- |
| (ms) | 1.33 |
| (ms) | 4.30 |
| (ms) | 6.16 |

**Question b)**

Use the “brute force” simulation of the decoherence by calculating the density matrices and applying rotation and time-evolution operators.

**Question bi)**

We can break individual noise traces into smaller segments to increase the effective number of spins in your simulation because we are treating the noise signals from the spin ensemble as a random variable with a Gaussian distribution. A random variable with Gaussian distribution has the following properties:

1. the mean of the noise is zero
2. the noise is stationary (linearly time invariant)

Furthermore, since the noise signals are treated as random variables, and we have 100 spins in our ensemble with each spin having very long range of data points (80,000), we can assume the noise is “ergodic”. This is the same as saying we can extract the statistics of a long sample of noise with a shorter duration, but with more samples to compensate.

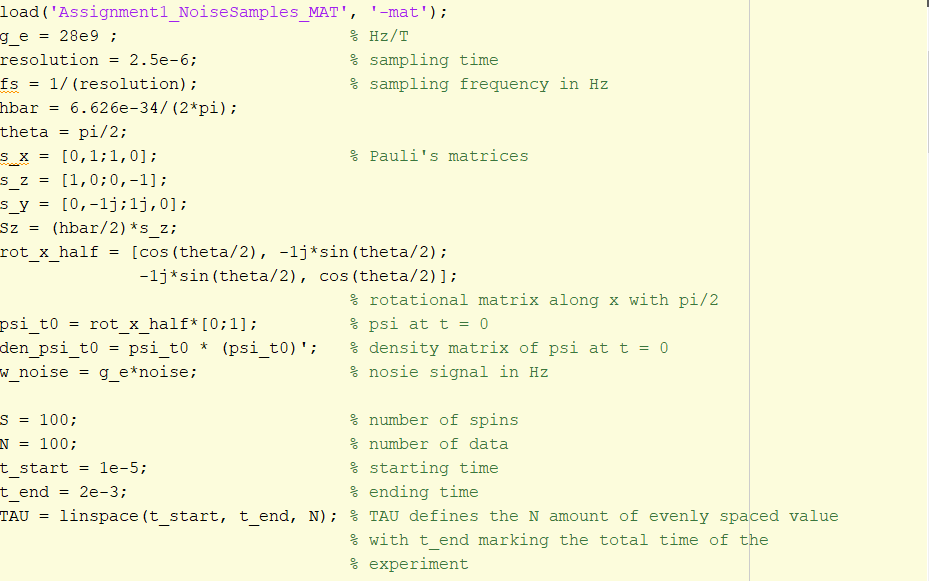
Henceforth using the above 2 properties and assumption of noise being ergodic, we can divide individual noise traces into shorter traces but analyzed over a larger sample of spins (100).

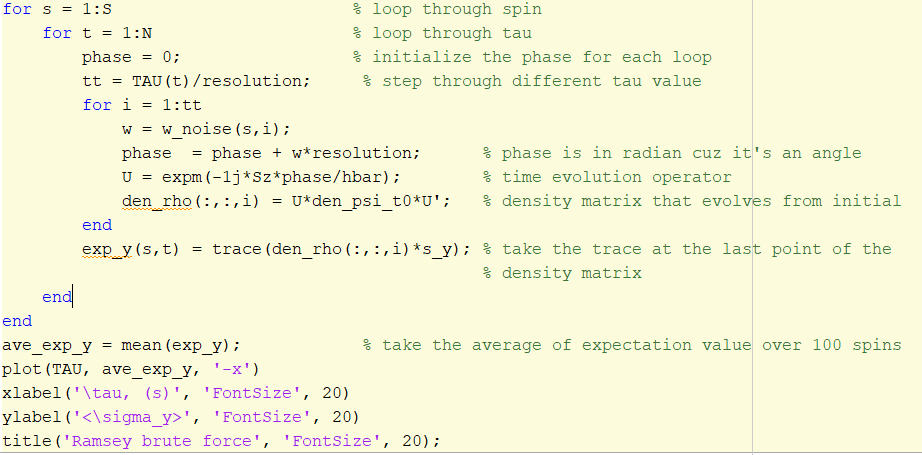
**Question bii)**

For this part of the question we will be repeating question 1 but with a different approach. Rather than using filter theory formalism to get an idealized behavior of the decoherence function, we will be utilizing density matrices, rotational operators and time evolution matrices to simulate the decoherence of the spin ensemble.

For our first pulse sequence, Ramsey free induction decay.

Figure(15) and figure(16) illustrates the code to simulate the free induction decay:

figure(15)

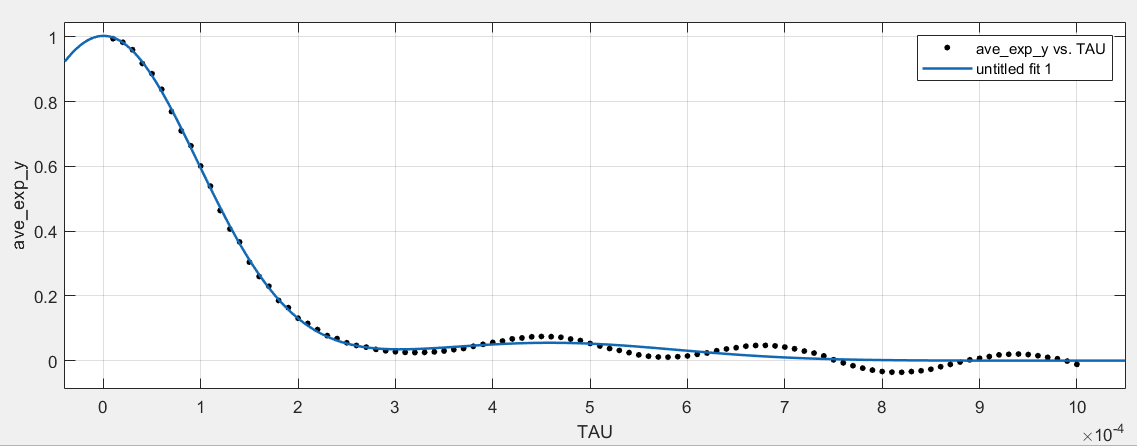
figure(16)

Figure(17) shows the Ramsey decoherence function:

Chart, histogram

Description automatically generatedfigure(17)

We invoke the curve fitting toolbox from Matlab to extract the , figure(18) shows the fitted curve below:

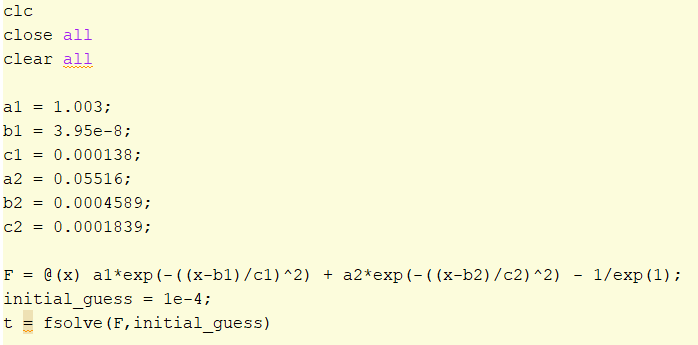
figure(18)

Whose equation is described as the following:

With the following parameters:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

With R-square value = 0.9958



From the above equation and code, we can extract the

The following is a comparison between the filter-function theory and brute-force method of the same Ramsey decoherence:

|  |  |
| --- | --- |
| Chart  Description automatically generated | Chart, histogram  Description automatically generated |
|  |  |

We can see in the Brute-force method, there are a lot more noise oscillating about 0 after the signal decoheres, which behaves as we expect what noise should behave, comparing to the filter-function theory where it is a smooth decaying curve throughout.

We shall now investigate simulating the Hahn Echo pulse sequence using the brute force method. The following is the code in figure(19) and figure(20) generates such sequence. Notice in contrast to Ramsey, I created a range of pulse length interval between π/2 pulse and the pulse, and stepped through each pulse length.

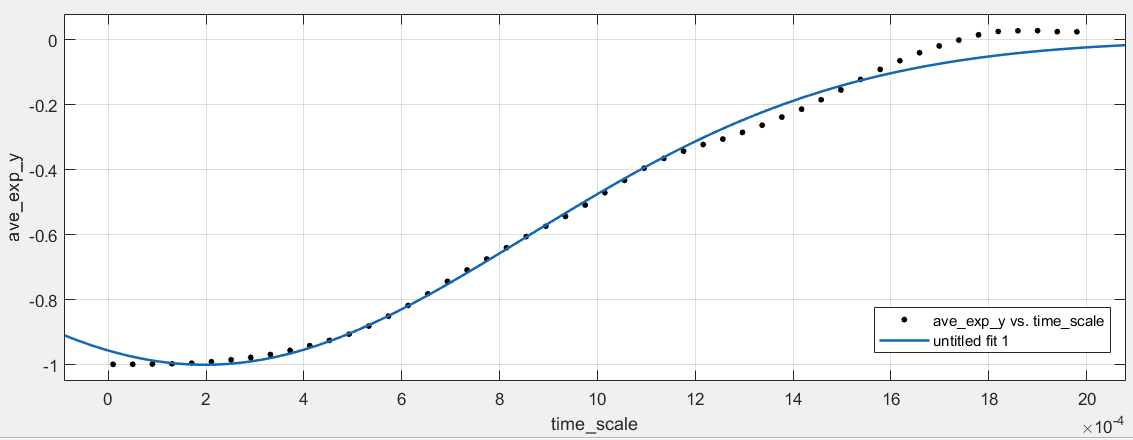
|  |
| --- |
| figure(19) |
| figure(20) |

Which produce the following Hahn Echo decoherence shown in figure(21)

Chart, line chart

Description automatically generatedfigure(21)

Following a similar procedure from previous, we input these data points into curve fitting tool and obtain the following equation in figure(22):

figure(22)

Described by the following Gaussian equation:

Where

|  |  |
| --- | --- |
| a | -1 |
| b | 0.0001971 |
| c | -0.0009303 |

R-square value = 0.9933

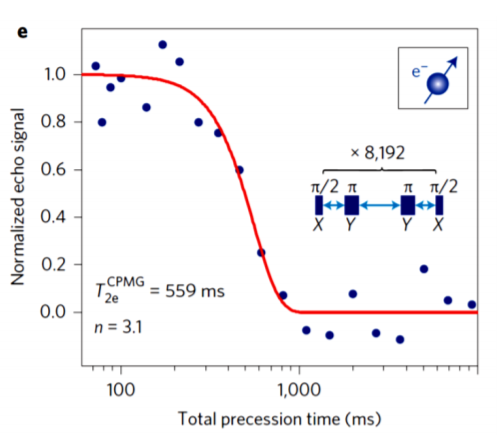
Which gives us the using the “fsolve” function in Matlab

If we compare the decoherence from filter function theory and brute-force method:

|  |  |
| --- | --- |
| Chart, line chart, histogram  Description automatically generated | Chart, line chart  Description automatically generated |
|  |  |

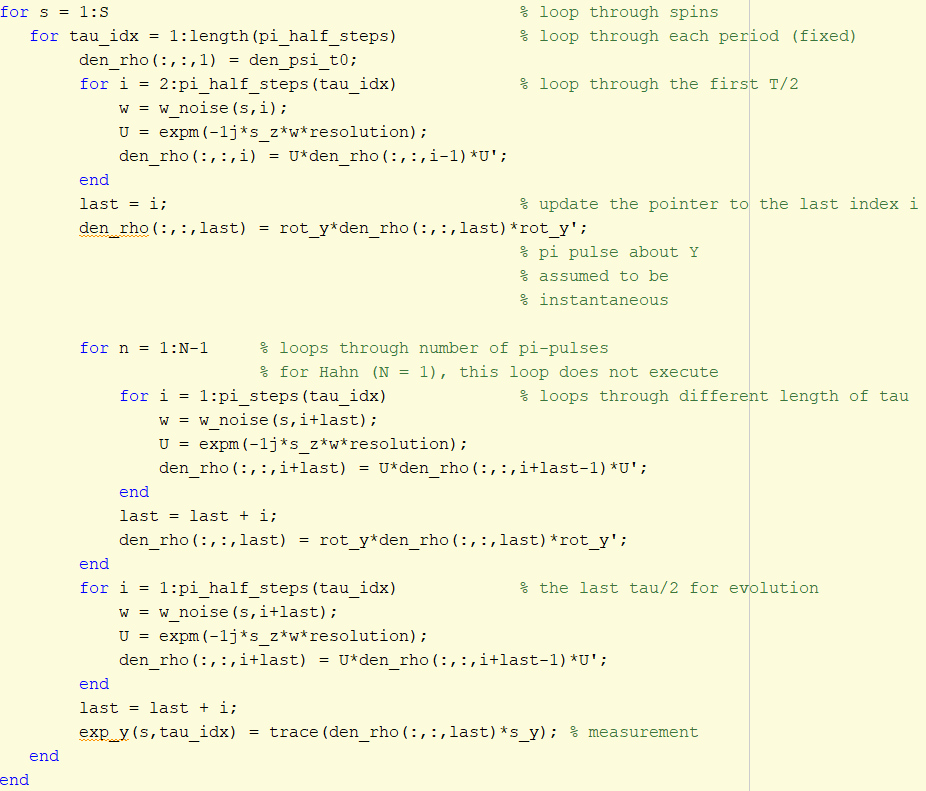
Much like the comparison we did for Ramsey decay, we can see some minor oscillations in the brute-force method, which reflect the realistic behavior of the spin ensemble. The time between them are very close in the same order of magnitude, which is what one should expect. The main difference between is that in the -brute force’ method, we specifically apply a π pulse rotation about X axis and take the expectation value about Y axis. This essentially take the projection of the spin on the Y axis, and since we applied a π pulse, the spin ensemble is rotated about π phase, which lands on the negative side of Y. Hence, taking the trace of the product between the density matrix and Pauli Y matrix will give you a negative value as shown above.

Lastly we will perform a ‘brute-force’ simulation on Dynamical Decoupling. As suggested from the journal article in Nature, we applied the π pulses about Y axis instead of X, as we have done for Hahn Echo. Figure(23) proves a brief explanation of dynamical decoupling.



[Juha T. Muhonen et al., Nature Nanotechnology] figure(23)

The brute-force simulation of dynamical decoupling shares the same code and structure as the one from Hahn echo, but the π pulse rotation about Y instead of X as shown below:

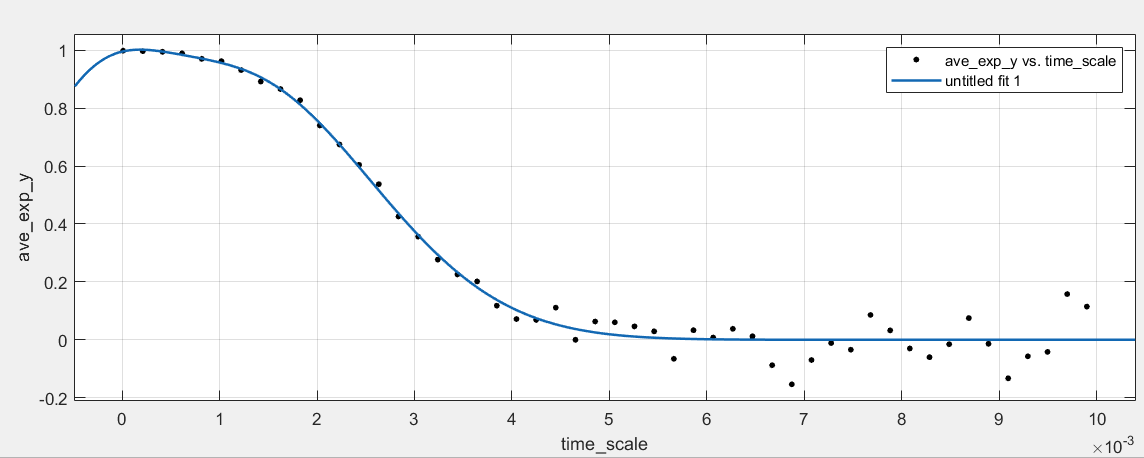


Figure(24) shows the dynamical decoupling when N = 5 pulses and N = 10 pulses.

Chart

Description automatically generatedfigure(24)

We then input the data from dynamical decoupling for 5 pulses into curve fitting and obtain the following equation from figure(25)

figure(25)

Where

|  |  |
| --- | --- |
|  | 0.434 |
|  | -0.0004302 |
|  | 0.001003 |
|  | 0.9078 |
|  | 0.001168 |
|  | 0.00195 |

With R-square value = 0.9814

From the above equation and using fsolve, we can extract .

The following is a comparison between the filter-function theory and the brute force simulation for 5 pulses dynamical decoupling.

|  |  |
| --- | --- |
| Chart  Description automatically generated | Chart  Description automatically generated |
|  |  |

We can see a slight offset in between filter function theory and brute force mechanism, but they are both in the order of milliseconds, hence they are verify each other. If the size of the spin ensemble would be increased by another factor of 2, it is believed that the of brute force method will come closer to filter theory.

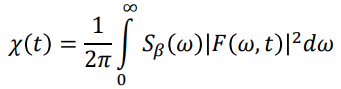
**Question biii)**

One of the advantages of filter function over brute-force simulation is that the filter function theory takes less time to simulate the system which models a close proximity of the actual system. It is an idealized estimation that for the case of brute-force, given a long enough trace, samples and computational power, its behavior will be the same as the filter function theory.

However, the advantage of brute-force model is that this method gives you the real behavior of the system, it provides a more realistic approach towards the analyzing the system and its behavior with a cost of higher computational cost.

**Question ci)**

Note that the decoherence function is given by the following equation:



If we are only interest in a small range of frequency, , we can safely assume that the bandwidth of the filter function of the dynamical decoupling is sufficiently narrow, this further justifies that we can treat the power spectral density of the noise at to be constant, which allow us to do the following :

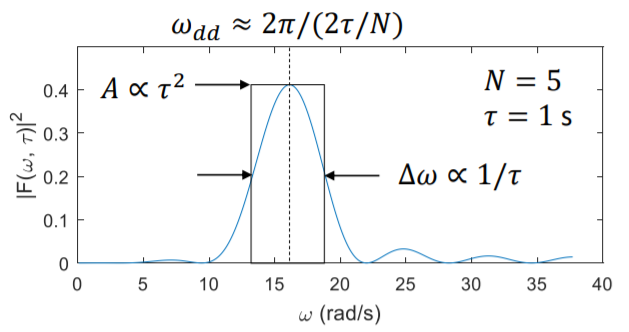
With

Hence, we can conclude with the following:

**Question cii)**

The brute force method allows us to obtain the decoherence function for a particular number of pulses**.** If we fix the time interval between π pulses, which indirectly defines the center frequency. By adding more pulses in the end, we increase the total duration of the experiment. This inherently changes the width of the filter and cause narrowing. We can use this to our advantage, as the width of the filter increases as we decrease the number of pulses, we are able to sample more frequency around the center frequency, of interest. For a small enough range of broadening, we are able to sample more signal due to a larger window. Hence using the formula

We are able to get a power spectral density of the noise at . We can then change the time interval between pulses which changes the center frequency, henceforth we can create a range of



One important assumption to note is that we assume the bandwidth of our filter function is narrow enough(approximately a Dirac delta function), such that can be removed from the integral. This is inherently an approximation of the formula, and we are assuming that is narrow enough.

Since we are broadening our filter for low number of pulses, a lot of noise will also be sampled around the center frequency as a by-product. Hence it is worth mentioning that the Signal-to-Noise ratio of this method is low, one can say that we are trading off resolution of the SNR for a faster way of obtaining the power spectral density across frequencies.