

# Novel strategies for Quantum Control:

Applications in Quantum Information Processing & Spectroscopy

A Thesis

submitted to

Indian Institute of Science Education and Research Pune

in partial fulfillment of the requirements for the

BS-MS Dual Degree Programme

by

**Gaurav Bhole**

20121078



Indian Institute of Science Education and Research Pune

Dr. Homi Bhabha Road,  
Pashan, Pune 411008, INDIA.

May, 2017

Supervisor: Dr. T. S. Mahesh

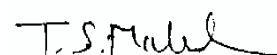
© Gaurav Bhole 2017

All rights reserved



# Certificate

This is to certify that this dissertation entitled ‘**Novel strategies for Quantum Control: Applications in Quantum Information Processing & Spectroscopy**’ towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Gaurav Bhole at Indian Institute of Science Education and Research under the supervision of Dr. T. S. Mahesh, Associate Professor, Department of Physics, during the academic year 2016-2017.



Dr. T. S. Mahesh

Committee:

Dr. T. S. Mahesh

Dr. M. S. Santhanam



*This thesis is dedicated to my parents*



# Declaration

I hereby declare that the matter embodied in the report entitled '**Novel strategies for Quantum Control: Applications in Quantum Information Processing & Spectroscopy**', are the results of the work carried out by me at the Department of Physics, Indian Institute of Science Education and Research Pune, under the supervision of Dr. T. S. Mahesh and the same has not been submitted elsewhere for any other degree.



Gaurav Bhole

30-03-2017





# Acknowledgments

- First and foremost I would like to express my sincere gratitude to my supervisor Dr. T. S. Mahesh for giving me an opportunity to work with him at the NMR-Quantum Information Processing group right from my 2<sup>nd</sup> year, and his exceptional guidance, support, inspiration and insights.
- I would like to thank my thesis co-supervisor Dr. M. S. Santhanam for the useful discussions, inputs and the wonderful course in Quantum Information.
- Collaborators – Dr. Abhishek Shukla, Anjusha V. S., Deepak Khurana and Varad Pande.
- Lab members Soham Pal and Sudheer Kumar for the useful discussions.
- NMR-Quantum Information Processing group at IISER-Pune.
- My parents and family for their continuous support and encouragement.
- Friends at IISER-Pune for a wonderful time who made this journey enjoyable.



# Abstract

Control over the quantum world is expected to open new horizons in this century. “Given a quantum system, how best we can control its dynamics?” is the question at the heart of quantum control technology. Several advancements in diverse fields such as spectroscopy, quantum information science, communication science, chemical kinetics, imaging, and sensors hinge upon establishing precise and robust control over the quantum dynamics. In this thesis, we examine certain novel strategies for efficiently generating robust quantum controls. First by using a suitable interaction frame, we separate the free evolution of the quantum system from the external controls. Secondly, by coarse graining the external control parameters we evaluate any general control propagator in terms of a small set of precalculated unitaries. These strategies alleviate the complexity issues posed by standard propagator evaluation methods, which are based on iteratively exponentiating the full Hamiltonian. We benchmark the efficiency of our algorithm with standard algorithms in terms of computational time. Further, we incorporate these strategies on a hybrid quantum control algorithm which combines a global initial-optimization with a local final-optimization. We also describe the importance of our quantum control algorithms in quantum information as well as spectroscopy. Finally, some applications in quantum information processing and spectroscopy are demonstrated using nuclear spin-systems controlled by NMR techniques. Super-adiabatic quantum state transfer in spin chains was demonstrated in a 3- qubit NMR system. Single and multi-band selective inversion pulses were designed and used in the heat-bath algorithmic cooling protocol to increase the polarisation of low natural abundance nuclear isotopes.



# Contents

<b>Abstract</b>	<b>xi</b>
<b>1 Introduction</b>	<b>3</b>
1.1 What is control theory all about? . . . . .	3
1.2 What is this thesis about? . . . . .	5
1.3 Why do we need quantum control? . . . . .	5
1.4 What is optimal quantum control? . . . . .	6
1.5 Outline of the thesis . . . . .	9
<b>2 State of the art</b>	<b>11</b>
2.1 Framework for optimal quantum control algorithms . . . . .	12
2.2 GRAPE algorithm . . . . .	16
2.3 Krotov algorithm . . . . .	19
2.4 Discussions . . . . .	25
<b>3 Speeding up optimal control algorithms</b>	<b>29</b>
3.1 Quantum dynamics in the interaction picture . . . . .	31
3.2 Optimal control in the interaction picture . . . . .	32
3.3 Coarse graining and matrix recycling . . . . .	34

3.4	Fast matrix exponentiation . . . . .	36
3.5	Comparison and discussion . . . . .	40
<b>4</b>	<b>Experiments in NMR using quantum control</b>	<b>43</b>
4.1	Applications in quantum information processing . . . . .	43
4.2	Applications in spectroscopy . . . . .	52
<b>5</b>	<b>Summary and Conclusions</b>	<b>55</b>

# List of publications

1. Pande, Varad R., **Gaurav Bhole**, Deepak Khurana, and T. S. Mahesh. "Strong Algorithmic Cooling in Large Star-Topology Quantum Registers." arXiv:1702.04992 (2017).
2. **Gaurav Bhole**, V. S. Anjusha, and T. S. Mahesh. "Steering quantum dynamics via bang-bang control: Implementing optimal fixed-point quantum search algorithm." Physical Review A 93.4 (2016): 042339.
3. **Gaurav Bhole**, Abhishek Shukla, and T. S. Mahesh. "Benford analysis: A useful paradigm for spectroscopic analysis." Chemical Physics Letters 639 (2015): 36-40.
4. **Gaurav Bhole**, Abhishek Shukla, and T. S. Mahesh. "Benford distributions in NMR." arXiv:1406.7077 (2014).





# Chapter 1

## Introduction

### 1.1 What is control theory all about?

Weather, cyclones, earthquakes – such are things no one can control. Everything else can be controlled and falls within the realms of *control theory*. Control theory deals with the manipulation of dynamical systems which are influenced by numerous external parameters and constraints. Thus, control theory finds applications wherever one encounters such a dynamical system – ranging from the macroscopic aeroplanes and railway engines to the microscopic atoms and molecules. Typically, the central problem of control theory is to steer the evolution of a dynamical system to a desired output state. Often, there are many solutions to this problem which are known as *controls*. However, one seeks to attain the ‘best’ solutions, which are referred to as *optimal controls*. One needs to specify what this ‘best’ means, i.e. the optimality criterion. For example, if the dynamical system has to reach the target state as fast as possible, the optimality criterion is time and the controls satisfying this criterion are said to be time optimal. In summary, optimal control theory is all about finding the control variables which optimize a certain objective respecting the imposed constraints.

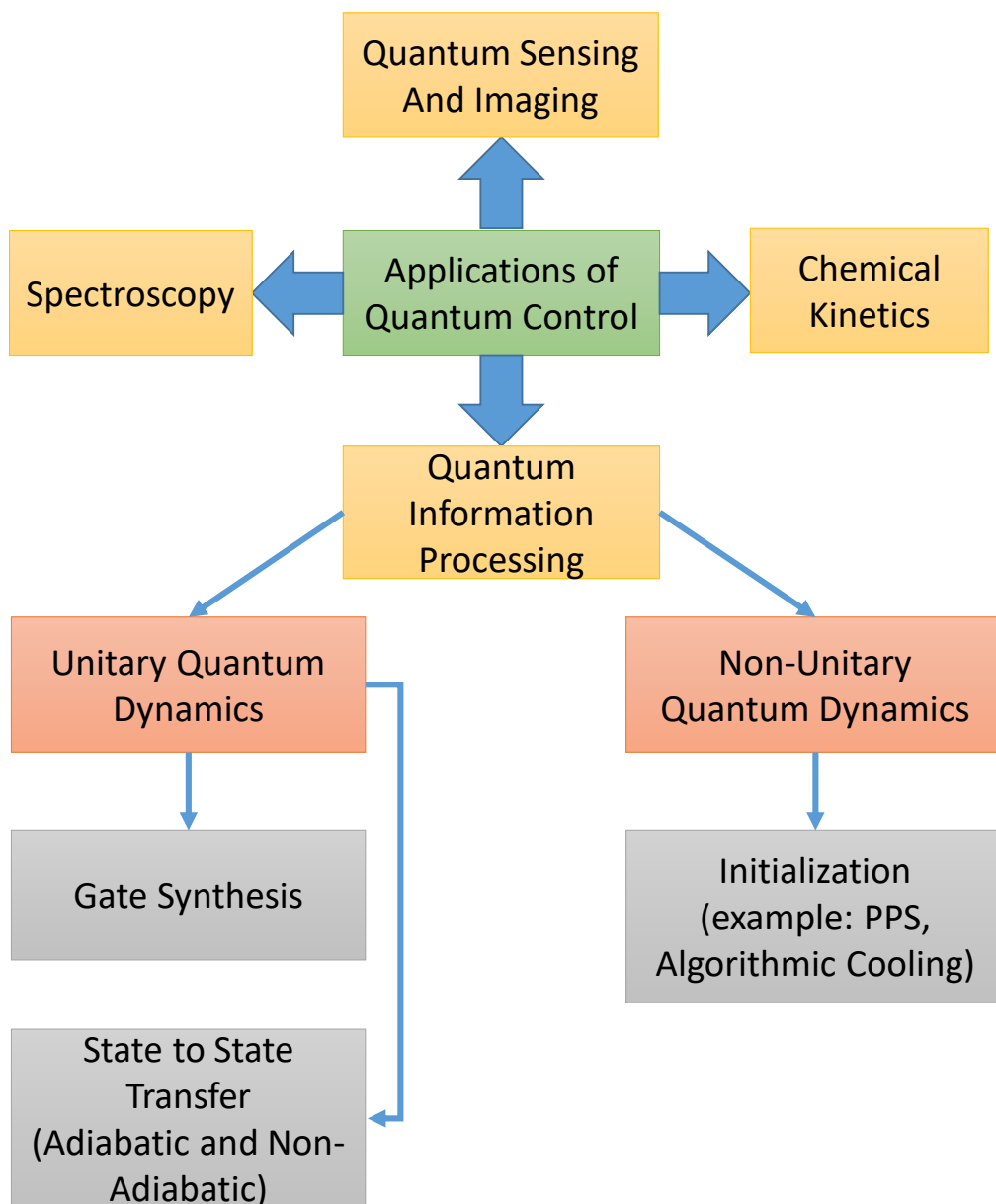


Figure 1.1: Applications of quantum control

## 1.2 What is this thesis about?

In this thesis, we would be dealing with the coherent control of quantum systems. Coherent control is the control of the dynamics of a quantum system usually by light or shaped electromagnetic pulses. In particular, our focus would be on controlling the quantum dynamics of spin-1/2 nuclei via shaped radio-frequency (RF) pulses. The spin-1/2 nucleus placed in a strong magnetic field acts as a qubit or a natural two level system – an ideal setting for quantum information processing[1]. This thesis deals with the numerical algorithms used to design and optimize the shaped RF pulses which will steer our spin-qubits to a desired target state or implement a target operation as required by various quantum information processing tasks.

## 1.3 Why do we need quantum control?

Information processing devices based on quantum mechanical systems, known as quantum devices, have the potential to outperform their classical counterparts on the grounds of computational complexity. Classical computers encode information in bits – 0 and 1 which correspond to certain physical states of the transistor. This classical information possess the inherent robustness of digital logic; that is, the classical bits remain in a particular state until one wishes to change them. On the other hand, the information which is encoded by quantum devices in quantum systems or qubits is fragile. Meaning, whatever we can do to control the evolution of the quantum system, Nature can also do it in an unknown and uncontrollable fashion thereby ruining the computation. Thus, a robust and efficient control over the dynamics of the quantum system is indispensable for any ‘quantum device’ to prove its potential computational usefulness.

Moreover, DiVincenzo has laid down the following criterion[2] for any physical implementation of quantum computation or a quantum information processing device:

1. A scalable physical system with well characterized qubits.
2. The ability to initialize the qubits to a simple known state.
3. Long decoherence time, much longer than gate operation time.

4. A universal set of quantum gates.
5. A qubit specific measurement capability.
6. The ability to inter-convert stationary and flying qubits.
7. The ability to faithfully transmit flying qubits between specified locations.

This thesis is mainly concerned with criterion 4, which is the ability to implement any logic gates on our quantum system. Any computation or algorithm, be it classical or quantum, consists of a network of logic gates acting on the input bits/qubits, which change the state of input bits/qubits and lead to the desired output. All the classical logic gates can be easily implemented via transistor-transistor logic (TTL). However, generating arbitrary quantum gates is a non-trivial task, and thus, developing and applying methods for doing this accurately and efficiently would be the central focus of the thesis.

## 1.4 What is optimal quantum control?

Any quantum system interacting with time dependent electromagnetic pulses can be described by the following Hamiltonian:

$$\mathcal{H} = \mathcal{H}_0 + \sum_{k=1}^m u_k(t)\mathcal{H}_k$$

Here,  $\mathcal{H}_0$  refers to the internal Hamiltonian which is an intrinsic property of the quantum system and is thus time independent. The set  $\{\mathcal{H}_k\}$  consists of the control Hamiltonians which are externally applied. The set  $\{u_k(t)\}$  refers to the shaped electromagnetic pulses which decide the strength of the external Hamiltonians  $\{\mathcal{H}_k\}$  to be applied at each instant of time. The time evolution of the quantum system under the Hamiltonian  $\mathcal{H}$  is governed by the Schrödinger equation:

$$\frac{d}{dt} |\psi\rangle = -i\mathcal{H} |\psi\rangle$$

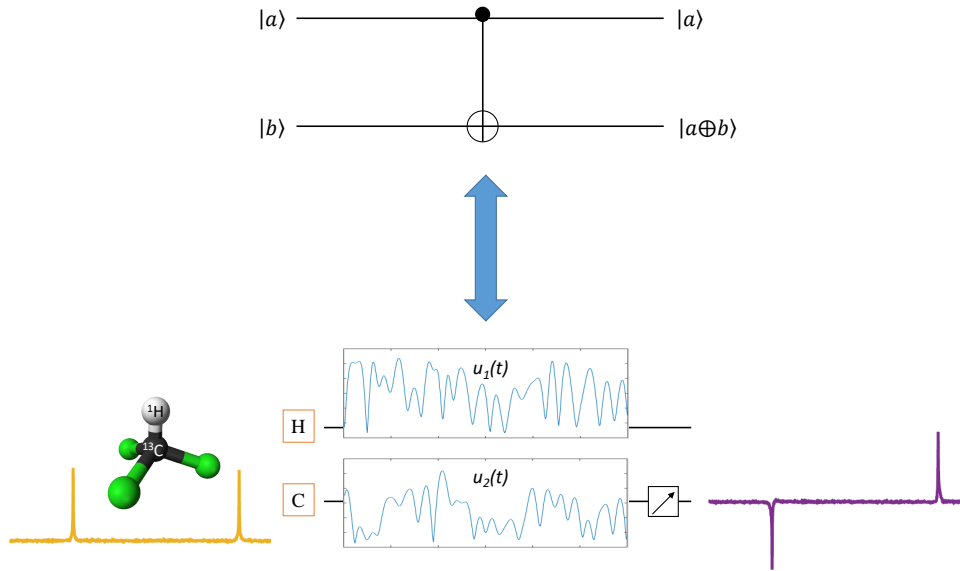


Figure 1.2: Shown in the top trace is the two qubit C-NOT gate, the quantum equivalent of an XOR gate. The lower trace is an indicative of how a C-NOT gate is experimentally implemented using spin qubits in an NMR-spectrometer. The  $^1\text{H}$  and  $^{13}\text{C}$  nuclei of chloroform form our 2 qubit system.  $\{u_1(t)\}$  and  $\{u_2(t)\}$  are the shaped RF-pulses which act on the 2 qubits and experimentally implement the C-NOT gate. Note that the shaped pulses  $\{u_1(t)\}$  and  $\{u_2(t)\}$  are not unique.

Note that we are working with  $N$ -dimensional Hilbert spaces where  $N = 2^n$  and  $n$  is the number of interacting qubits or spin-1/2 particles. Thus,  $\mathcal{H}$  is an  $N \times N$  matrix and  $|\psi\rangle$  is a  $N \times 1$  column vector. This amounts to solving  $N$  first-order linear differential equations, or equivalently, computing the matrix exponential of an  $N \times N$  matrix.

The control problem is to arrive at a set of controls  $\{u_k(t)\}$  so that the evolution of the quantum system is guided to implement a desired quantum logic gate or reach a desired target quantum state. We also need to define a fidelity function  $\Phi(u_k)$  which quantifies how good the controls  $\{u_k(t)\}$  are and how well have they managed to implement a desired gate or reach a desired state. Thus, the optimal control algorithms aim to maximize this fidelity function  $\Phi(u_k)$  with respect to certain constraints depending upon the physical system and hardware involved.

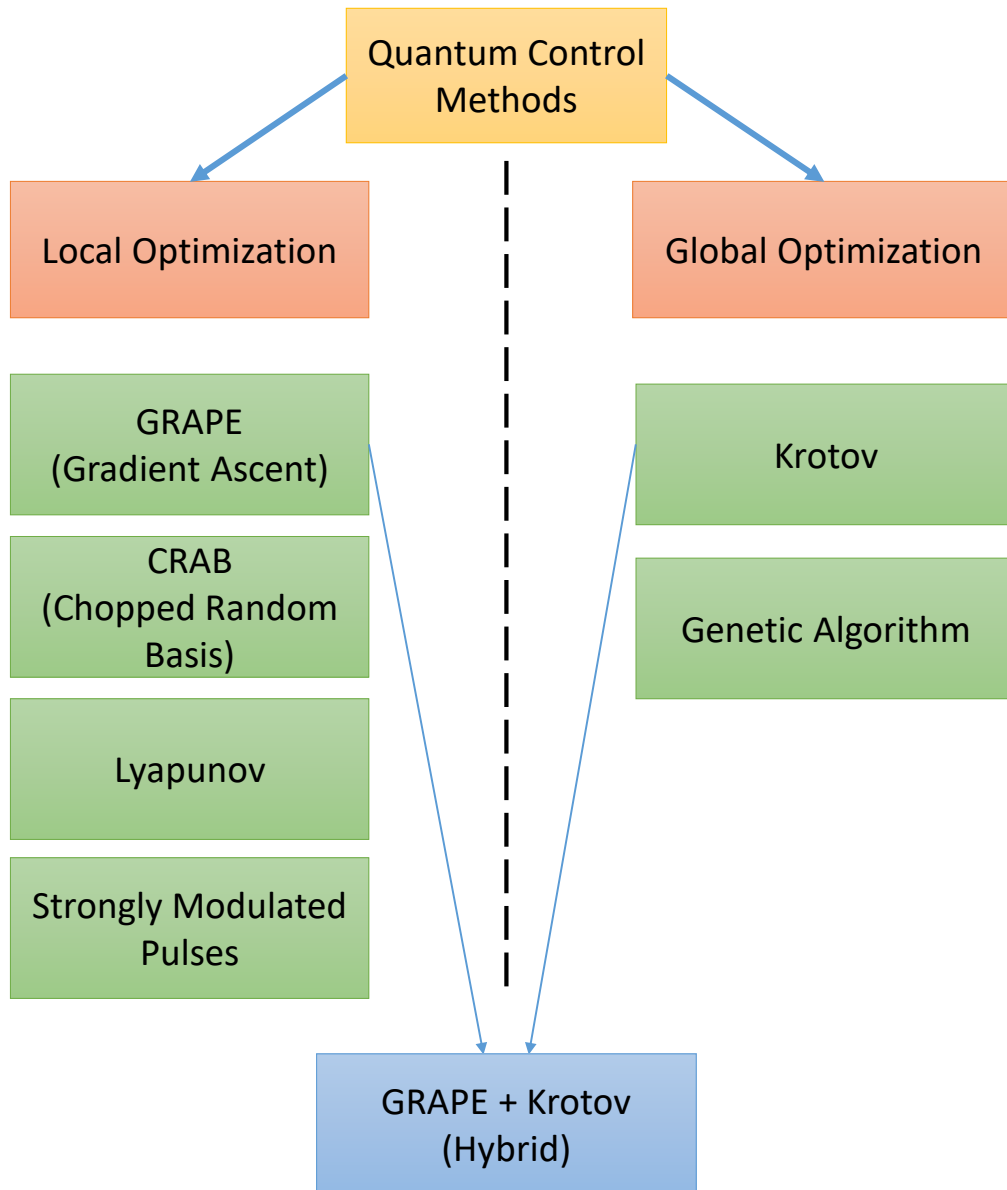


Figure 1.3: Numerical methods for quantum control such as GRAPE[3], CRAB[4], Lyapunov[5], strongly modulated pulses[6], Krotov[7], Bang-Bang using genetic algorithm[8].

## 1.5 Outline of the thesis

The rest of the thesis is organized as follows:

**Chapter 2:** We describe a common framework for optimal control algorithms based on the smooth modulation of electromagnetic pulses. Within this framework, we further describe two famous optimal control algorithms – GRAPE (**G**radient **A**scent **P**ulse **E**ngineering) and Krotov for synthesizing quantum gates and preparing desired quantum states for NMR quantum information processing. Although these algorithms are general and can be adopted for any architecture for quantum information processing, we shall focus only on their adapted versions for spin qubits in NMR.

**Chapter 3:** Here, we propose the evaluation of quantum dynamics in the interaction frame of quantum mechanics (Dirac picture) and incorporate it into optimal control algorithms. We also combine coarse graining and matrix recycling to significantly speed up matrix exponentiation, which is the bottle-neck of all quantum control algorithms. We name this method of matrix exponentiation as REDO (**R**apid **E**xponentiation by **D**iscrete **O**perators). Using these tools, we modify the existing optimal control algorithms and achieve a speed-up of over 2-3 times over the standard methods.

**Chapter 4:** In this chapter, we apply the developed optimal control algorithms for experimental applications in NMR quantum information processing and spectroscopy. Firstly, we perform selective inversions in a 3-qubit quantum register. We also prepare a pseudo-pure state which mimics the state  $|000\rangle$  in the same 3-qubit quantum register in order to initialize it. Next, we implement super-adiabatic quantum state transfer in a 3-spin chain [9]. We also perform multi-bandwidth selective inversions as an application in NMR spectroscopy and algorithmic cooling[10].





# Chapter 2

## State of the art

In this chapter, we describe two widely used optimal control algorithms namely GRAPE (**G**radient **A**scent **P**ulse **E**ngineering)[3] and Krotov[7]. Although these algorithms are general and can be adopted to any architectures for quantum computing, we describe only the NMR adaptations of the algorithms. The objective of these algorithms is to arrive at a set of control variables which implement the desired quantum gate or reach a target quantum state. For smaller spin systems consisting of one or two qubits, it is trivial to construct the quantum gates analytically. Moreover, in a heteronuclear NMR system, each qubit can be addressed individually and hence single qubit rotations and two qubit gates can be implemented using resonant RF pulses and natural couplings between each pair of qubits. The main difficulty lies in implementing selective single qubit gates in homonuclear NMR systems where a single resonant RF control field affects all qubits of the same nuclear specie. This makes it very difficult to analytically construct selective gates and bring about selective excitations. Hence, it becomes necessary to resort to numerical methods for optimization which can perform selective gates, especially in homonuclear NMR systems. Additionally, while dealing with large number of qubits, i.e. large Hilbert spaces, it is advantageous to use numerical optimization techniques rather than analytical methods.

## 2.1 Framework for optimal quantum control algorithms

A quantum system interacting with time-dependent electromagnetic fields can be described by the following Hamiltonian:

$$\mathcal{H} = \mathcal{H}_0 + \sum_{k=1}^m u_k(t) \mathcal{H}_k \quad (2.1)$$

where  $\mathcal{H}_0$  is the time-independent internal Hamiltonian,  $\{\mathcal{H}_k\}$ 's are the external control Hamiltonians and  $\{u_k(t)\}$  are the control fields which denote the strength of the external Hamiltonians to be applied at each instant of time. In the NMR setting, the qubits are formed by spin-1/2 nuclei of a molecular ensemble placed in a strong magnetic field  $B_0 \hat{z}$ . Thus, in the rotating frame picture of an NMR spin system in an isotropic medium, the internal Hamiltonian is as follows:

$$\mathcal{H}_0 = - \sum_i \omega_i I_{iz} + 2\pi \sum_{i < j} J_{ij} I_{iz} I_{jz} \quad (2.2)$$

where,  $\omega_i$ ,  $J_{ij}$ , and  $I_i$  denote the resonance offset, indirect(scalar) coupling and collective Pauli spin operator respectively. The time dependent Hamiltonian which we can control via radio-frequency (RF) pulses is as follows:

$$\mathcal{H}_c(t) = \sum_{k=1}^m u_k(t) \mathcal{H}_k \quad (2.3)$$

where the index  $k$  runs over all nuclear species (e.g.  $^1\text{H}$ ,  $^{13}\text{C}$ ,  $^{15}\text{N}$ ). In NMR, we can modulate the amplitude and frequency of the RF field, or equivalently, the strength of RF  $u_{kx}(t)$  and  $u_{ky}(t)$  in  $X$  and  $Y$  quadratures respectively. The control Hamiltonian can thus be expressed as:

$$\mathcal{H}_c(t) = \sum_{k=1}^m (u_{kx}(t) I_x + u_{ky}(t) I_y) \quad (2.4)$$

where  $I_x$  and  $I_y$  are the collective spin operators for all the nuclear spins which are sensitive to a particular control field. The evolution of the quantum system under the Hamiltonian  $\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_c(t)$  is governed by the Schrödinger equation:

$$\frac{d}{dt} |\psi\rangle = -i\mathcal{H} |\psi\rangle \quad (2.5)$$

The evolution of the state vector in time is governed by

$$|\psi(t)\rangle = U(t) |\psi(0)\rangle \quad (2.6)$$

where the propagator  $U(t) = \exp(-i\mathcal{H}t/\hbar)$ . In order to evaluate the propagator, one needs to make use of the Dyson series which is computationally expensive. Hence, we discretize the total evolution time  $T$  into  $N$  steps, each of duration  $\Delta t = T/N$ . As a result, we can consider the strength of the control pulse as a constant during every time step.

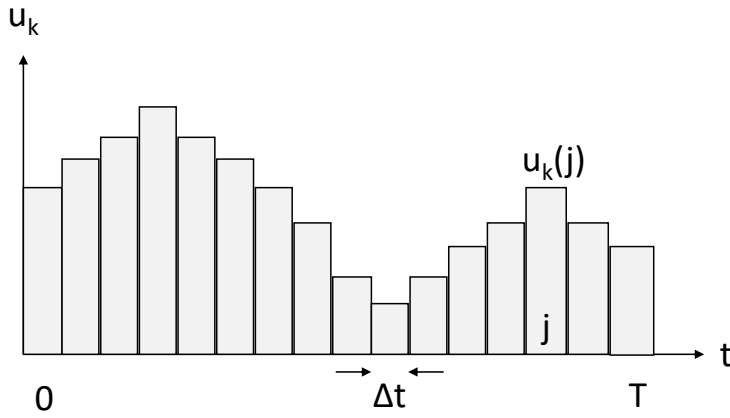


Figure 2.1: Schematic representation of the discretization of control amplitude  $u_k(t)$  into  $N$  segments, each of duration  $\Delta t = T/N$ .

Due to the discretization of total control duration, we now have piece-wise constant Hamiltonian. Thus, the propagator during the  $j^{th}$  step is:

$$U_j = \exp \left[ -i\Delta t \left( \mathcal{H}_0 + \sum_{k=1}^m u_k(j)\mathcal{H}_k \right) \right] \quad (2.7)$$

The time evolution during the entire control duration  $T$  can now be evaluated without the use of Dyson series. The total propagator or unitary is:

$$U(T) = U_N U_{N-1} U_{N-2} \dots U_2 U_1 \quad (2.8)$$

Our aim is to synthesize a target unitary  $U_F$ . For synthesizing such a desired unitary, we need to maximize the overlap between  $U(T)$  and  $U_F$  up to a global phase. That is, we need to minimize  $\|U_F - e^{i\phi}U(T)\|^2$ . Equivalently, we need to maximize the fidelity function:

$$\Phi = |\langle U_F | U(T) \rangle|^2 \quad (2.9)$$

Here,  $\Phi$  can take values between 0 and 1. If our synthesized unitary  $U(T)$  matches perfectly with the desired unitary  $U_F$ , then  $\Phi = 1$ . Note that  $\Phi = \Phi(u_k)$ . Thus one needs to optimize the controls  $\{u_k(t)\}$  in order to maximize  $\Phi$ . Following is a general scheme for the numerical optimization algorithms for arriving at a set of  $\{u_k(t)\}$  which maximize the fidelity  $\Phi$ .

This general scheme which would be followed for GRAPE and Krotov optimal control algorithm reads as:

1. Start with a initial random guess for the shape of the pulse.
2. Calculate the fidelity functional for the pulse shape, which is the overlap between target unitary and simulated unitary.
3. Make small changes to the shape of the pulse so that the fidelity functional improves.
4. Repeat step 2 and 3 until no more improvements to the fidelity functional are possible or threshold fidelity has reached.

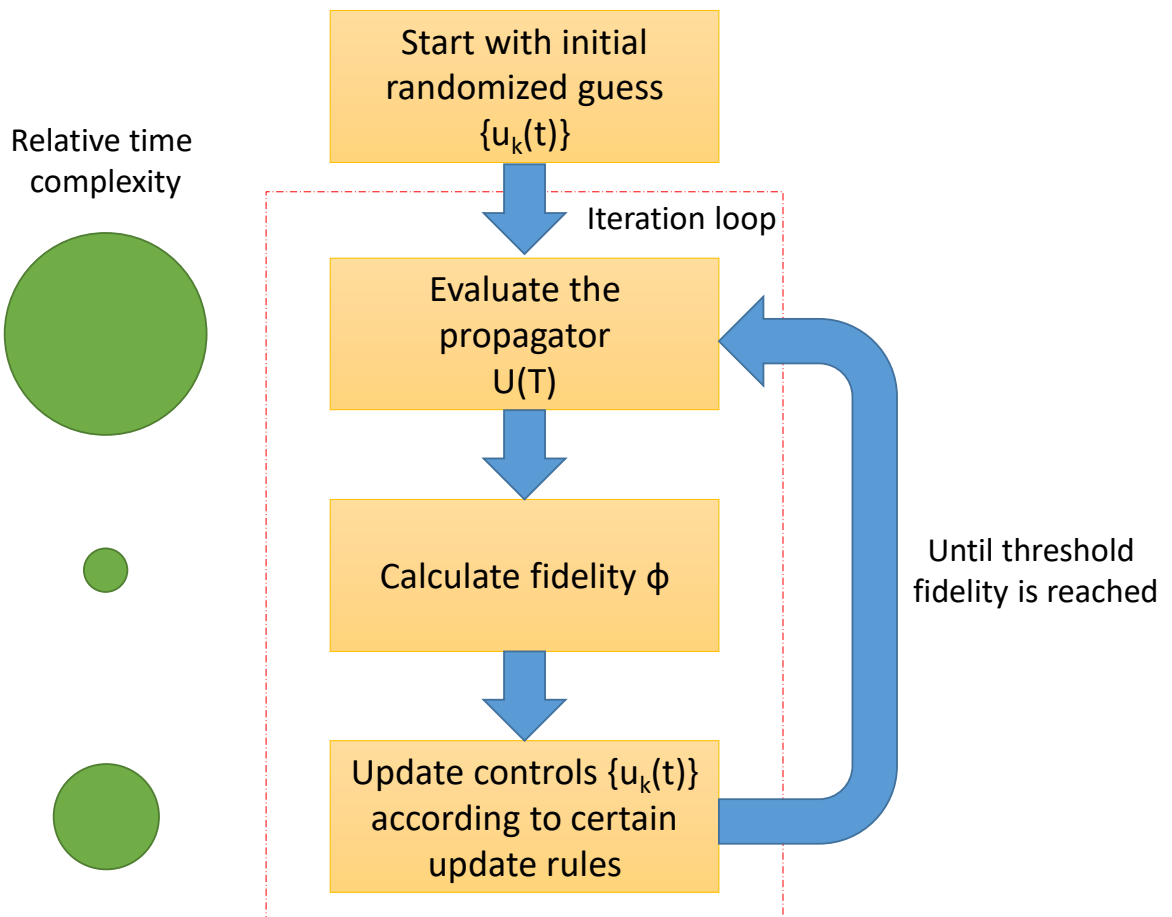


Figure 2.2: Schematic representation of numerical optimization algorithms for optimal quantum control.

## 2.2 GRAPE algorithm

The GRAPE algorithm falls within the category of gradient ascent algorithms where in each iteration, the control variables are modified in the direction of the steepest ascent to maximize a fidelity functional. The direction of steepest ascent is determined by the gradient of the fidelity functional w.r.t to the control variables. In the following, we provide a step by step description of the GRAPE algorithm adapted for NMR.

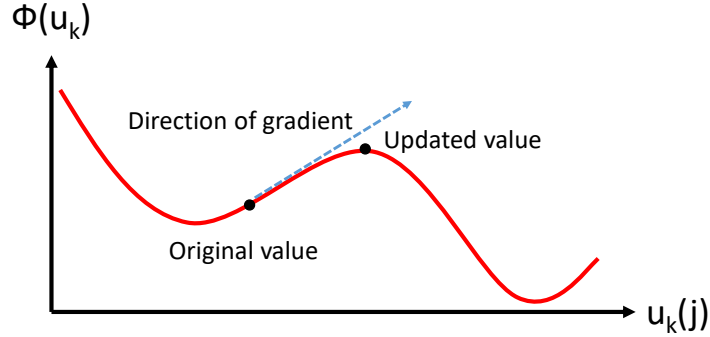


Figure 2.3: At every time step  $j$ , the controls are updated in the direction of gradient.

1. **Initial randomized guess:** Start with a randomized guess of RF controls  $\{u_k(t)\}$ . If there are  $m$  nuclear species, there would be  $m$  sets of RF controls  $\{\{u_1(t)\}, \{u_2(t)\}, \dots \{u_m(t)\}\}$ . Each control  $\{u_i(t)\}$  is a vector  $(u_i(1), u_i(2), \dots u_i(N))$  denoting the amplitude of RF field of  $i^{th}$  control during each segment of duration  $\Delta t = T/N$ .
2. **Calculating propagators:** Calculate the propagator for each segment  $j = 1$  to  $N$  using:
 
$$U_j = \exp \left[ -i\Delta t \left( \mathcal{H}_0 + \sum_{k=1}^m u_k(j)\mathcal{H}_k \right) \right] \quad (2.10)$$
3. **Forward and backward propagators:** Calculate and store each forward propagator  $X_j = U_j U_{j-1} \dots U_2 U_1$  for  $j = 1$  to  $N$ . Similarly calculate and store each backward propagator  $P_j = U_{j+1}^\dagger U_j^\dagger \dots U_N^\dagger U_F$  for  $j \leq N$  with  $P_N = U_F$ .
4. **Calculating the fidelity:** Using  $X_N = U(T)$ , the fidelity for the simulated unitary is:

$$\Phi = |\langle U_F | U(T) \rangle|^2 \quad (2.11)$$

5. **Evaluating the gradient:** Calculate the gradient of fidelity functional  $\Phi(u_k)$  with respect to each  $u_k(j)$ . The fidelity functional  $\Phi = |\langle U_F|U(T)\rangle|^2$  can be expressed as:

$$\Phi = \langle U_F|U(T)\rangle \langle U(T)|U_F\rangle \quad (2.12)$$

$$= \langle U_F|U_N U_{N-1} \dots U_2 U_1\rangle \langle U_N U_{N-1} \dots U_2 U_1|U_F\rangle \quad (2.13)$$

$$= \langle P_j|X_j\rangle \langle X_j|P_j\rangle \quad (2.14)$$

Now we need to evaluate the gradient  $\delta\Phi/\delta u_k(j)$  to first order in  $\Delta t$ . Note that  $u_k(j)$  appears only in the propagator  $U_j$ . Thus,

$$\frac{\delta\Phi}{\delta u_k(j)} = \frac{\delta}{\delta u_k(j)} \langle P_j|X_j\rangle \langle X_j|P_j\rangle \quad (2.15)$$

$$= \frac{\delta}{\delta u_k(j)} \langle U_{j+1}^\dagger \dots U_N^\dagger U_F|U_j \dots U_1\rangle \langle U_j \dots U_1|U_{j+1}^\dagger \dots U_N^\dagger U_F\rangle \quad (2.16)$$

$$= \langle U_{j+1}^\dagger \dots U_N^\dagger U_F|\frac{\delta U_j}{\delta u_k(j)} \dots U_1\rangle \langle \frac{\delta U_j}{\delta u_k(j)} \dots U_1|U_{j+1}^\dagger \dots U_N^\dagger U_F\rangle \quad (2.17)$$

$$(2.18)$$

Substituting  $\delta U_j/\delta u_k(j) \approx -i\Delta t \mathcal{H}_k U_j$  and using the chain rule of differentiation, the expression for gradient is:

$$\frac{\delta\Phi}{\delta u_k(j)} = -2\Re\{\langle P_j|i\Delta t \mathcal{H}_k X_j\rangle \langle X_j|P_j\rangle\} \quad (2.19)$$

6. **Updating the RF controls:** The RF control amplitude at every segment  $j$  can be update using the following update rule:

$$u_k(j) \longrightarrow u_k(j) + \epsilon \frac{\delta\Phi}{\delta u_k(j)} \quad (2.20)$$

Here,  $\epsilon$  is the user defined step size which tells us ‘how much to go in the direction of the gradient’. However, always moving the pulse in the direction of the gradient in constant step size  $\epsilon$  may lead to loss of convergence, as the algorithm can get stuck in local minima. Hence, one needs to use the method of conjugate gradients. Before using conjugate gradients, the step size  $\epsilon$  must be optimized at each step such that the pulse

is moved to the highest fidelity point in the direction of gradient. This can be done by moving the pulse in steps of  $\epsilon$  and  $2\epsilon$  in the direction of gradient and evaluating the fidelity at these two points. Further, a quadratic function can be fit through these fidelity points. The maxima of the quadratic, which is the maximum fidelity gives us the optimum value of step size  $\epsilon$ .

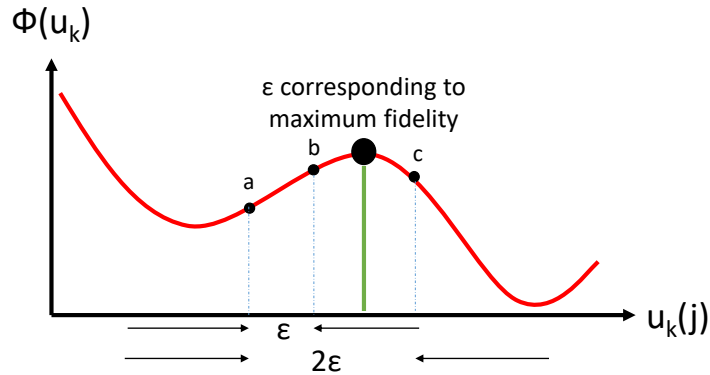


Figure 2.4: Determining the optimum value of step size  $\epsilon$  in the direction of gradient by fitting a quadratic function through three points a, b and c corresponding to an update step size of 0,  $\epsilon$  and  $2\epsilon$  respectively.

7. **Iterations:** With the updated values of  $u_k(j)$  go to step 2 and repeat until threshold fidelity is reached.



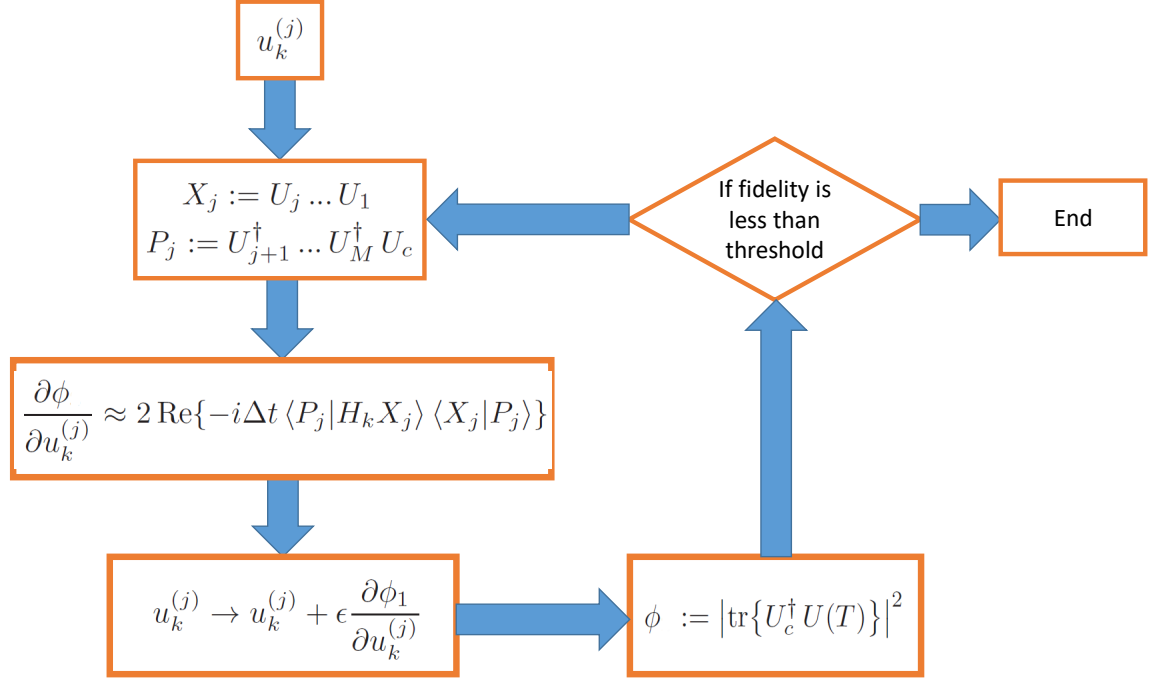


Figure 2.5: Summary of the GRAPE algorithm.

## 2.3 Krotov algorithm

### 2.3.1 Basis of Krotov optimization

The Krotov optimal control algorithm is based on the method of Lagrangian multipliers which are used to find the maxima or minima of a function w.r.t equality constraints. For example, consider a function  $f(x, y)$  which is to be maximized w.r.t the constraint  $g(x, y) = 0$ . Define a variable  $\lambda$  known as the Lagrangian multiplier. With this, the Lagrangian is defined as:

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda \cdot g(x, y) \quad (2.21)$$

The method says that if  $f(x_0, y_0)$  is a maximum of  $f(x, y)$  w.r.t the constraint  $g(x, y) = 0$ , then there exist a  $\lambda_0$  such that  $(x_0, y_0, \lambda_0)$  is a stationary point for  $\mathcal{L}(x, y, \lambda)$ . Stationary

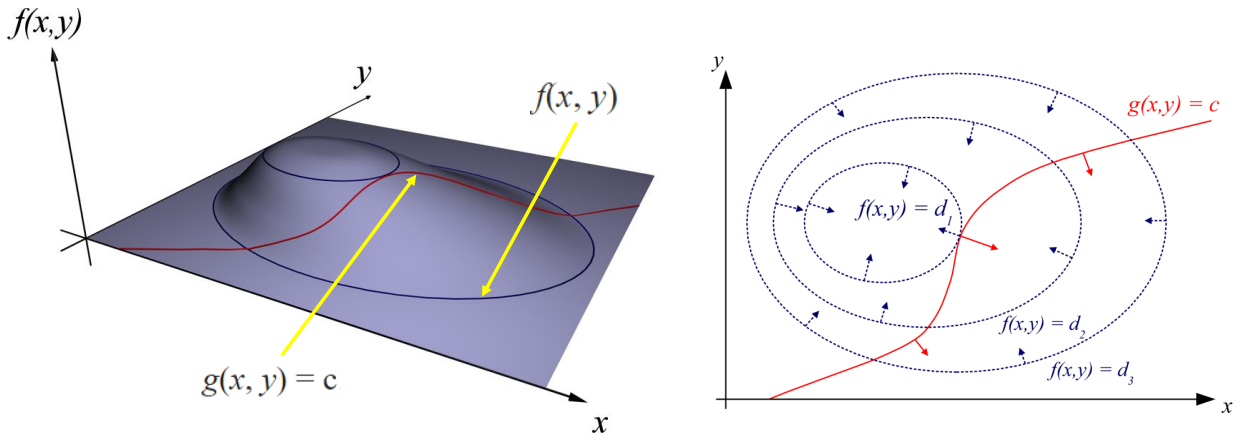


Figure 2.6: Shown on the left is the plot of function  $f(x, y)$  which is to be maximized w.r.t the constraint  $g(x, y) = c$ . Shown on the right is the contour plot of the same. The point  $(x, y)$  where  $g(x, y) = c$  intersects the function  $f(x, y)$  is the maxima of the function w.r.t the constraint. (Image taken from Wikipedia)

points are those points where the partial derivatives of  $\mathcal{L}$  w.r.t  $(x, y, \lambda)$  are all zero. Thus the necessary condition (but not sufficient) for stationary points is:

$$\nabla_{x,y,\lambda}\mathcal{L} = \left( \frac{\partial\mathcal{L}}{\partial x}, \frac{\partial\mathcal{L}}{\partial y}, \frac{\partial\mathcal{L}}{\partial\lambda} \right) = (0, 0, 0) \quad (2.22)$$

Solving the above equation gives us the values of  $x$  and  $y$  which maximize  $f(x, y)$  w.r.t  $g(x, y) = c$ .

### 2.3.2 Krotov optimal control algorithm

Here, we adopt the method of Lagrange multipliers as explained in the previous section to derive the Krotov optimal control algorithm. Our aim is to design a target unitary  $U_F$ . We define a fidelity functional  $\Phi = |\langle U_F | U(T) \rangle|^2$  which is to be maximized; where  $U(T)$  is the synthesized unitary evolution from time  $t = 0$  to  $t = T$ . Along with maximizing the fidelity, we also wish to minimize the consumption of RF power. Hence we incorporate an additional term in the fidelity which acts as a penalty for total RF power consumed. Thus, our modified

fidelity functional is:

$$J = \Phi - \lambda \int_0^T \sum_k u_k^2(t) dt \quad (2.23)$$

Here,  $\lambda$  is a user defined parameter for scaling the penalty and is not to be confused with the Lagrange multiplier in the previous section. The functional  $J$  is analogous to the function  $f(x, y)$  in the previous section. We now need to find the maximum of this functional  $J$  using the method of Lagrange multipliers. Similar to the constraint  $g(x, y)$  in the previous section, our constraint here is that the evolution of our quantum system follows the Schrödinger equation. Thus, the constraint is:

$$\langle B(t) | \frac{d}{dt} + i\mathcal{H} | U(t) \rangle \quad (2.24)$$

where  $B(t)$  is analogous to the Lagrange multiplier  $\lambda$  we had in the previous section. When the evolution  $U(t)$  follows Schrödinger equation, we have,  $\frac{dU(t)}{dt} = -i\mathcal{H}U(t)$  and the constraint value becomes 0. The functional analogous to  $\mathcal{L}(x, y, \lambda)$  has the form:

$$\tilde{J} = \Phi = |\langle U_F | U(T) \rangle|^2 - \lambda \int_0^T \sum_k u_k^2(t) dt - \int_0^T \langle B(t) | \frac{d}{dt} + i\mathcal{H} | U(t) \rangle \quad (2.25)$$

Now we can use calculus of variations to find out the gradient of  $\tilde{J}$  w.r.t  $B(t)$ ,  $U(t)$ ,  $u_k(t)$  and set each of them to 0.

$$\frac{\delta \tilde{J}}{\delta B(t)} = 0; \quad \frac{\delta \tilde{J}}{\delta U(t)} = 0; \quad \frac{\delta \tilde{J}}{\delta u_k(t)} = 0; \quad (2.26)$$

The first relation  $\frac{\delta \tilde{J}}{\delta B(t)} = 0$  above gives back the equation of motion or Schrödinger equation for  $U(t)$ , implying that the evolution must follow the Schrödinger equation:

$$\frac{dU(t)}{dt} = -i\mathcal{H}U(t) \quad (2.27)$$

Solving the second relation  $\frac{\delta \tilde{J}}{\delta U(t)} = 0$  gives us two conditions. The first condition implies that  $B(t)$  which can also be interpreted as the backward propagator, must follow the Schrödinger

equation:

$$\frac{dB(t)}{dt} = -i\mathcal{H}B(t) \quad (2.28)$$

The second condition gives us a initial state condition for the above equation.

$$B(T) = \frac{\partial\Phi}{\partial U(T)} \quad (2.29)$$

It is evident that as  $\Phi$  is a function of  $U(T)$ , the initial condition  $B(T)$  indeed depends on the final state  $U(T)$ . Finally, the third relation  $\frac{\delta\tilde{J}}{\delta u_k(t)} = 0$  gives us the rule for update of RF controls as follows:

$$u_k(t) = \Im \langle B(t) | \mathcal{H}_k | U(t) \rangle \quad (2.30)$$

Thus on solving  $\frac{\delta\tilde{J}}{\delta B(t)} = 0$ ,  $\frac{\delta\tilde{J}}{\delta U(t)} = 0$  and  $\frac{\delta\tilde{J}}{\delta u_k(t)} = 0$  we have the three conditions for a stationary point or an extremum for the functional  $\tilde{J}$ :

$$\frac{dB(t)}{dt} = -i\mathcal{H}B(t) \quad (2.31)$$

$$B(T) = \frac{\partial\Phi}{\partial U(T)} \quad (2.32)$$

$$u_k(t) = \Im \langle B(t) | \mathcal{H}_k | U(t) \rangle \quad (2.33)$$

Using these 3 conditions, we now describe the Krotov optimal control algorithm. Similar to the GRAPE algorithm, we also discretize the control duration of time  $T$  into  $N$  segments of duration  $\Delta t = T/N$  each.

Preliminary definitions: (Note that  $i$  stands for iteration number)

**Forward propagator:**  $U_i(t) = \exp[-i\Delta t (\mathcal{H}_0 + \sum_{k=1}^m u_{k,i}(j)\mathcal{H}_k)]$

**Initial condition for forward propagator:**  $U_i(0) = \mathbb{I}$

**Forward update rule:**  $u_{k,i}(t) = (1 - \delta)\tilde{u}_{k,i-1}(t) + \frac{\delta}{\lambda}\Im \langle B_{i-1}^\dagger(t) | \mathcal{H}_k | U_i(t) \rangle$

**Backward propagator:**  $B_i(t) = \exp[-i\Delta t (\mathcal{H}_0 + \sum_{k=1}^m \tilde{u}_{k,i}(j)\mathcal{H}_k)]$

**Initial condition for backward propagator:**  $B_i(T) = \frac{\partial \Phi}{\partial U(T)} = U_F \langle U_F | U_i(T) \rangle$

**Backward update rule:**  $\tilde{u}_{k,i}(t) = (1 - \eta)u_{k,i}(t) + \frac{\eta}{\lambda}\Im \langle B_i^\dagger(t) | \mathcal{H}_k | U_i(t) \rangle$

Using the above definitions, the algorithm is as follows:

1. **Initial random guess:** Start with a randomized guess of RF controls  $\{u_{k,0}(t)\}$ . Set  $u_{k,0}(t) = \tilde{u}_{k,0}(t)$ , where  $\{\tilde{u}_{k,0}(t)\}$  is the set of backward propagating controls. The ‘0’ in the subscript stands for the iteration number.
2. **Forward and backward propagators:** Evaluate the forward propagator  $U_0(t)$  using  $u_{k,0}(t)$  to get  $U_0(T)$ . Using  $U_0(T)$ , evaluate the initial condition  $B_0(T)$  for backward propagation. Now propagate  $B_0(T)$  backwards using the controls  $\tilde{u}_{k,0}(t)$ .
3. **Updating the forward control:** Update the RF amplitude  $u_{k,1}(0)$  using the forward update rule.
4. **Propagating forward:** Propagate  $U_1(0)$  to  $U_1(\delta t)$  using the RF control  $u_{k,1}(0)$ .
5. **Forward evolution:** Repeat steps 3 and 4 to complete the forward evolution and obtain  $U_1(T)$ .
6. **Calculating fidelity:** Using the total propagator obtained above, calculate the fidelity  $\Phi$ .

7. **Initial condition for backward propagation:** Calculate  $B_1(T)$  using  $U_1(T)$  obtained above.
8. **Updating the backward control:** Update the RF amplitude  $\tilde{u}_{k,1}(T)$  using the backward update rule.
9. **Propagating backward:** Propagate  $B_1(T)$  to  $B_1(T-\delta t)$  using the RF control  $\tilde{u}_{k,1}(T)$ .
10. **Backward evolution:** Repeat steps 8 and 9 to complete the backward evolution.
11. **Iterations:** Go back to step 3 and repeat the above procedure until threshold fidelity is reached.

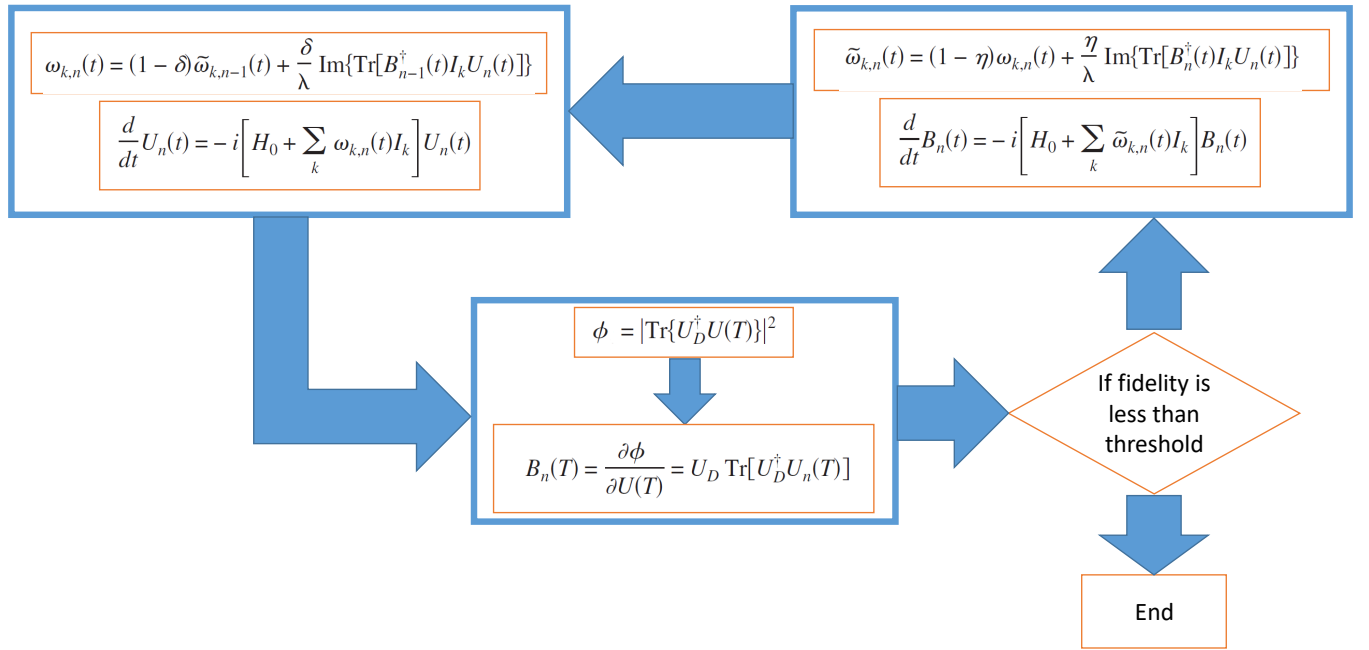


Figure 2.7: Summary of the Krotov optimal control algorithm.

## 2.4 Discussions

1. **GRAPE versus Krotov:** GRAPE is a local optimization algorithms whereas Krotov is a global optimization algorithm. The fundamental difference in GRAPE and Krotov is in the way RF controls are updated.

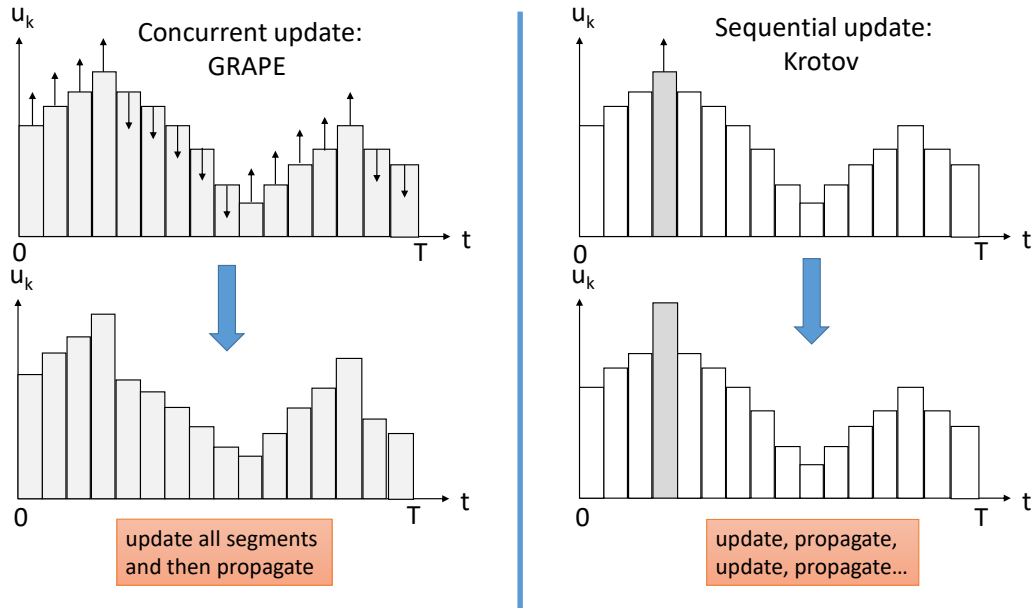


Figure 2.8: Concurrent update in GRAPE and sequential update in Krotov

**Update:** In GRAPE, all the segments are updated at once and then the entire propagator for control duration  $T$  is evaluated. Thus, the update of a particular segment does not depend upon any other segment. This type of update is known as concurrent update. On the contrary, in Krotov, a particular segment is updated and then propagated. This is done until the propagator for entire control duration  $T$  is found. Thus, the update of a particular segment depends upon the previously updated segments.

**Convergence:** In GRAPE, before updating we always need to adjust the step size and move the pulse to highest fidelity point to maintain monotonic convergence. On the other hand, the Krotov has the inbuilt property of monotonic convergence because of the way the algorithm is constructed. In fact during update of every segment, Krotov tries to maximize the fidelity functional. Since Krotov is a global optimization algorithm, it makes coarse adjustments rapidly and reaches high fidelities within few iterations. However, being a global optimization algorithm, the drawback is that it

cannot make finer adjustments at higher fidelities and thus the convergence is very slow at high fidelities. On the other hand, GRAPE is a local optimization algorithm. It needs more iterations than Krotov to reach high fidelities, however, once a high fidelity has reached, GRAPE combined with the method of conjugate gradient provides a faster convergence than Krotov. This suggest a hybrid algorithm, in which one uses Krotov for initial iterations for coarse adjustments until a high fidelity is reached and then switches to GRAPE for finer adjustments and higher fidelities.

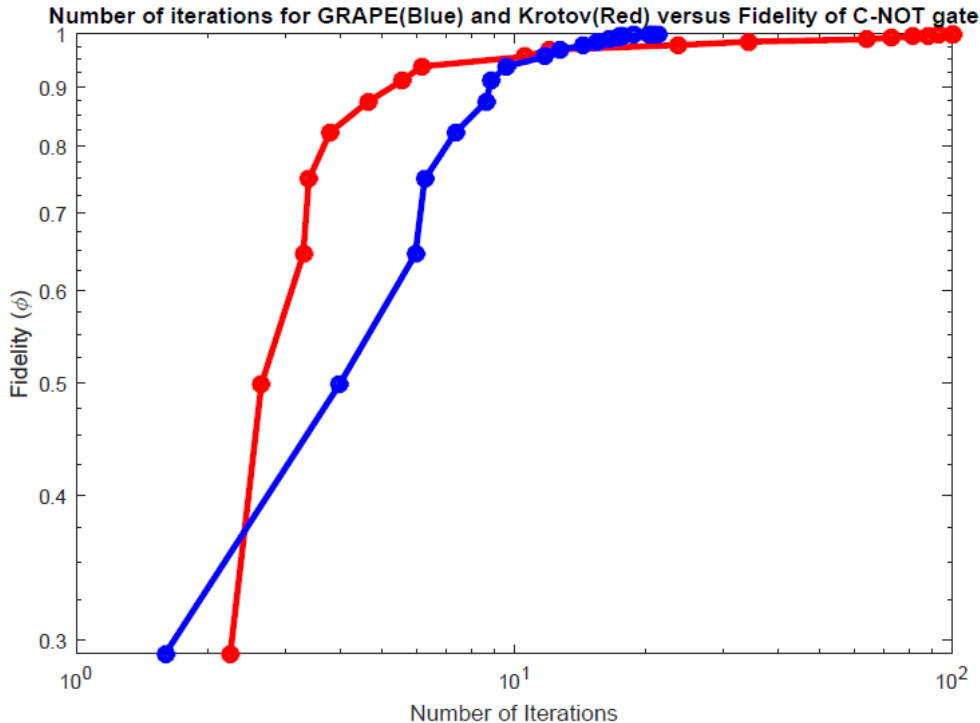


Figure 2.9: Convergence in GRAPE and Krotov

2. **RF inhomogeneities:** During experiment, the NMR sample tube receives different amount of RF irradiation in different parts. Say, the top 1/3<sup>rd</sup> part of the tube receives only 90% of RF, the bottom 1/3<sup>rd</sup> part receives 110% of RF and only the middle part receives the actual RF as synthesized by our optimal control algorithms. Thus, it is necessary to make the RF pulse robust against these inhomogeneities. For this, we design the pulse in such a way that the fidelity at 90% or RF irradiation as well as 110% of RF irradiation has the same high fidelity as 100% of RF irradiation.



3. **Use of functional spaces:** The GRAPE algorithm studied here involved updating each segment according to the update rule. However, one can also use a weighted set of basis functions which define the control profiles. In this case, instead of updating each segment we update the coefficient or weight corresponding to every basis function which results a change in entire control profile. This method, known as GRAFS (Gradient Ascent in Functional Space)[11] is used for generating bandwidth limited controls.



# Chapter 3

## Speeding up optimal control algorithms

In the previous chapter, we studied two famous optimal control algorithms – GRAPE and Krotov. Although the two algorithms have different schemes for updating the RF controls, the common process in each algorithm is evaluating the propagator  $U_j$  for every segment  $j$ . Evaluation of the propagator involves matrix exponentiation and is a bottle-neck of the algorithm; meaning it is the most time consuming routine of the algorithm. Moreover, the propagator needs to be calculated for every segment at least once during each iteration. Matrix exponentiation is responsible for severely slowing down the algorithms as the number of segments  $N$  increases or the dimension of Hilbert space (size of quantum system/number of qubits) increases. Thus any improvement in the matrix exponentiation routine can significantly speed up the algorithms[12, 13].

Here, we present a method we name as REDO (**R**apid **E**xponentiation by **D**iscrete **O**perators), to speed up the matrix exponentiation specifically in optimal control algorithms using coarse graining and matrix recycling. For this, we first need to evaluate the quantum dynamics in the interaction picture. We incorporate the REDO method of matrix exponentiation in GRAPE and Krotov optimal control algorithms and name them iGRAPE and iKrotov respectively.

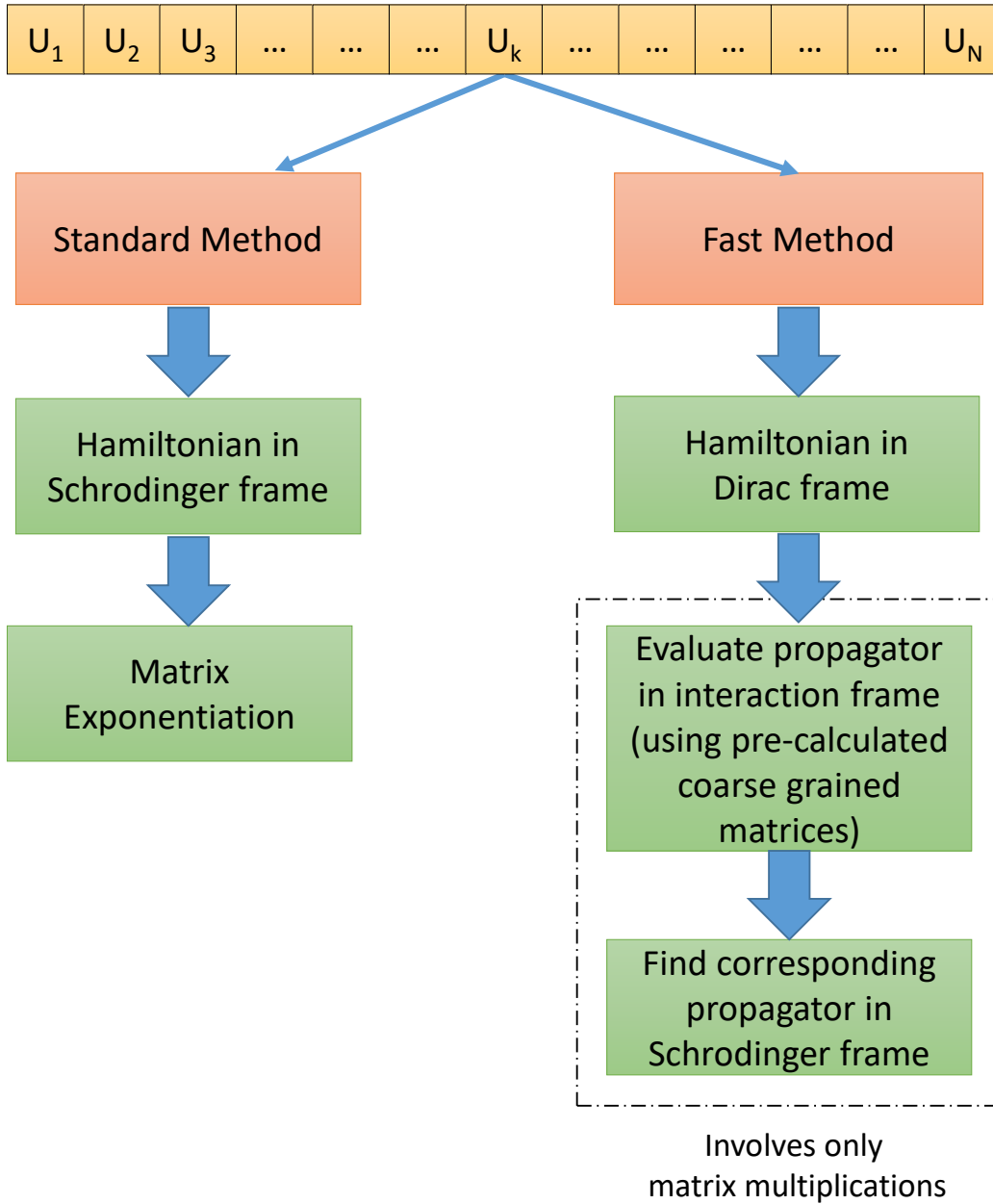


Figure 3.1: General scheme of evaluating propagator using standard method and REDO (fast) method.

### 3.1 Quantum dynamics in the interaction picture

Quantum mechanics has three different representations - Schrödinger picture, Heisenberg picture and the Dirac picture. These representations provide three different approaches to calculate some physical quantity or the expectation value of an observable. Of course, the answer to the physical quantity measured is independent of the representation used. In the Schrödinger picture, the state vectors evolve with time while the operators are fixed. On the other hand, in the Heisenberg picture, the operators evolve in time while the state vector is fixed. The Dirac representation or the interaction picture is an intermediate between the Schrödinger and Heisenberg picture. In the interaction picture, the state vector as well as the operator have time dependence. The interaction picture is particularly useful when the Hamiltonian can be separated into time independent and time dependent parts. The aim of the interaction picture is to go in the frame of the time independent Hamiltonian so that the dynamics of the quantum system can be purely described in terms of the external time dependent Hamiltonian. Consider the following Hamiltonian:

$$\mathcal{H}(t) = \mathcal{H}_0 + V(t) \quad (3.1)$$

Our aim is to solve the Schrödinger equation:

$$\frac{d}{dt} |\psi(t)\rangle = -i (\mathcal{H}_0 + V(t)) |\psi(t)\rangle \quad (3.2)$$

where  $|\psi(t)\rangle$  is the state vector in the Schrödinger picture. Let  $|\psi_I(t)\rangle$  be the state vector in interaction picture. The state vectors in the two pictures are related by:

$$|\psi_I(t)\rangle = e^{i\mathcal{H}_0 t} |\psi(t)\rangle \quad (3.3)$$

The evolution of state vector in the Schrödinger picture is given by  $|\psi(t)\rangle = U(t, t_0) |\psi(t_0)\rangle$  where  $U(t, t_0)$  is the propagator in the Schrödinger picture. Using this in the above equation, we have,

$$|\psi_I(t)\rangle = e^{i\mathcal{H}_0 t} U(t, t_0) |\psi(t_0)\rangle \quad (3.4)$$

Substituting  $|\psi(t_0)\rangle = e^{-i\mathcal{H}_0 t_0} |\psi_I(t_0)\rangle$  in the above equation, we have,

$$|\psi_I(t)\rangle = e^{i\mathcal{H}_0 t} U(t, t_0) e^{-i\mathcal{H}_0 t_0} |\psi_I(t_0)\rangle \quad (3.5)$$

Comparing the above equation with  $|\psi_I(t)\rangle = U_I(t, t_0) |\psi_I(t_0)\rangle$ , where  $U_I(t, t_0)$  is the propagator in the interaction frame, we have the following relation between the propagators in the two pictures:

$$U_I(t, t_0) = e^{i\mathcal{H}_0 t} U(t, t_0) e^{-i\mathcal{H}_0 t_0} \quad (3.6)$$

Equivalently,

$$U(t, t_0) = e^{-i\mathcal{H}_0 t} U_I(t, t_0) e^{i\mathcal{H}_0 t_0} \quad (3.7)$$

The propagator in the Schrödinger picture follows the differential equation:

$$\frac{d}{dt} U(t, t_0) = -i(\mathcal{H}_0 + V(t)) U(t, t_0) \quad (3.8)$$

Substituting  $U(t, t_0) = e^{-i\mathcal{H}_0 t} U_I(t, t_0) e^{i\mathcal{H}_0 t_0}$  in the above equation, we have,

$$\frac{d}{dt} U_I(t, t_0) = -iV_I(t) U_I(t, t_0) \quad (3.9)$$

where  $V_I(t) = e^{i\mathcal{H}_0 t} V(t) e^{-i\mathcal{H}_0 t}$  is the external time dependent Hamiltonian in the interaction picture. The above equation suggests that the propagator in the interaction picture depends only on the external time dependent Hamiltonian.

## 3.2 Optimal control in the interaction picture

Here, we apply the techniques from the previous section to optimal control algorithms. A quantum system interacting with time-dependent electromagnetic fields can be described by the following Hamiltonian:

$$\mathcal{H}(t) = \mathcal{H}_0 + \sum_{k=1}^m u_k(t) \mathcal{H}_k \quad (3.10)$$

In the optimal control algorithms, we have discretized the total control duration  $T$  into  $N$  segments of duration  $\Delta t = T/N$  each. Using the expression for evolution of operators in the interaction frame, the operator  $\mathcal{H}_k$  in the interaction frame in every segment is:

$$\tilde{\mathcal{H}}_j = e^{i\mathcal{H}_0 \Delta t} \mathcal{H}_k e^{-i\mathcal{H}_0 \Delta t} \quad (3.11)$$

The propagator in the Schrödinger picture during the  $j^{\text{th}}$  segment is:

$$U_j = \exp \left[ -i\Delta t \left( \mathcal{H}_0 + \sum_{k=1}^m u_k(j) \mathcal{H}_k \right) \right] \quad (3.12)$$

The above propagator in the interaction picture would be:

$$\tilde{U}_j = \exp \left[ -i\Delta t \left( \sum_{k=1}^m u_k(j) \tilde{\mathcal{H}}_k \right) \right] \quad (3.13)$$

where,  $\tilde{\mathcal{H}}_c(t) = \sum_{k=1}^m u_k(j) \tilde{\mathcal{H}}_k$ . In NMR, we can either modulate the amplitude and phase of the RF field or the strength of the field in  $X$  and  $Y$  quadratures independently. Thus,

$$\tilde{\mathcal{H}}_c(t) = \sum_{k=1}^m \left( u_{kx}(t) \tilde{I}_x + u_{ky}(t) \tilde{I}_y \right) \quad (3.14)$$

where  $\tilde{I}_x = e^{i\mathcal{H}_0\Delta t} I_x e^{-i\mathcal{H}_0\Delta t}$  and  $\tilde{I}_y = e^{i\mathcal{H}_0\Delta t} I_y e^{-i\mathcal{H}_0\Delta t}$  are the collective spin operators in the interaction picture. Note that the spin operators have now gained a time dependence. Equivalently, we can express the control Hamiltonian  $\tilde{\mathcal{H}}_c(t)$  in the amplitude-phase forms as follows:

$$\tilde{\mathcal{H}}_c(t) = \sum_{k=1}^m \Omega_k(t) \left( \cos(\theta_k(t)) \tilde{I}_x + \sin(\theta_k(t)) \tilde{I}_y \right) \quad (3.15)$$

Thus the propagator  $\tilde{U}_j$  can now be written as:

$$\tilde{U}_j = \exp \left[ -i\Delta t \left( \sum_{k=1}^m \Omega_k(j) \left( \cos(\theta_k(j)) \tilde{I}_x + \sin(\theta_k(j)) \tilde{I}_y \right) \right) \right] \quad (3.16)$$

An expression of the form  $\exp(-i\Omega(I_x \cos(\theta) + I_y \sin(\theta)))$  can be interpreted as rotation through an angle  $\Omega$  about the axis  $\hat{x} \cos(\theta) + \hat{y} \sin(\theta)$  in the  $x - y$  plane subtending an angle  $\theta$  with the  $x$ -axis[14]. This rotation can equivalently be expressed as a composition of three rotations – first through an angle  $\theta$  about the  $z$ -axis, second through an angle  $\Omega$  about the  $x$ -axis and third through an angle  $-\theta$  through the  $z$ -axis. Using this fact the propagator  $\tilde{U}_j$  can now be written as:

$$\tilde{U}_j = e^{(-i\sum_{k=1}^m \theta_k(j) \tilde{I}_z)} \exp \left( -i\Delta t \sum_{k=1}^m \Omega_k(j) \tilde{I}_x \right) e^{(i\sum_{k=1}^m \theta_k(j) \tilde{I}_z)} \quad (3.17)$$

where  $\tilde{I}_z = e^{i\mathcal{H}_0\Delta t}I_ze^{-i\mathcal{H}_0\Delta t}$  is the collective spin operator in the interaction picture. Since  $\mathcal{H}_0 = -\sum_i\omega_iI_{iz} + 2\pi\sum_{i<j}J_{ij}I_{iz}I_{jz}$  contains only  $z$ -terms,  $\tilde{I}_z$  commutes with  $\mathcal{H}_0$  and hence  $\tilde{I}_z = I_z$ . Thus, the propagator  $\tilde{U}_j$  is:

$$\tilde{U}_j = e^{(-i\sum_{k=1}^m\theta_k(j)I_z)} \exp\left(-i\Delta t\sum_{k=1}^m\Omega_k(j)\tilde{I}_x\right) e^{(i\sum_{k=1}^m\theta_k(j)I_z)} \quad (3.18)$$

Here, since  $I_z$  is a diagonal matrix,  $e^{(\pm i\sum_{k=1}^m\theta_k(j)I_z)}$  is just element by element exponential and not a matrix exponential. The only matrix exponentiation in  $\tilde{U}_j$  is the middle term. In the next section, we will study coarse graining which when combined with matrix recycling can help in getting rid of the matrix exponential in the evaluation of the propagator during the runtime of the optimal control algorithms.

### 3.3 Coarse graining and matrix recycling

The calculation of the propagator  $\tilde{U}_j$  involves the calculation of a single matrix exponential. However this propagator needs to be calculated for all the  $N$  segments in one iteration and typically one needs several iterations of the optimal control algorithm to attain high fidelities ( $> 0.99$ ). To overcome the calculation of repeated matrix exponentials, we can store  $\exp\left(-i\Delta t\sum_{k=1}^m\Omega_k(t)\tilde{I}_x\right)$  for all values of  $\Omega_k$  which lie in the range of admissible RF control powers as per the hardware. We can just call the stored matrices during the runtime of the algorithm and thus the evaluation of  $\tilde{U}_j$  will involve only 3 matrix multiplications, which is very fast compared to matrix exponentiation. However, the range of RF amplitude  $\Omega_k$  for a typical NMR experiment lies approximately between 0 to  $2^{18}$  rad/s. It is impossible to store matrices for all values of  $\Omega_k$  between 0 to  $2^{18}$ . Hence, using coarse graining, we take only the integer values between 0 to  $2^{18}$ . Even if we take only the integer values between 0 to  $2^{18}$ , it is still inefficient to store all the  $2^{18}$  matrices. All the more, if there are  $M$  different nuclei, we have to store  $M \times 2^{18}$  matrices. How does we store so many matrices?

In the decimal system, any integer  $'a_1a_2a_3 \dots a_n'$  where  $a_i$ 's are the digits of the integer, can be expressed as:

$$a_1a_2a_3 \dots a_n = a_1 \times 10^{n-1} + a_2 \times 10^{n-2} + \dots + a_{n-1} \times 10^1 + a_n \times 10^0 \quad (3.19)$$



Here,  $0 \leq a_i \leq 9$  since we are using a base of 10 i.e the decimal system. Analogously, suppose we use the base of 64, any integer ' $a_1a_2a_3 \dots a'_n$ ' can be expressed as:

$$a_1a_2a_3 \dots a_n = a_1 \times 64^{n-1} + a_2 \times 64^{n-2} + \dots + a_{n-1} \times 64^1 + a_n \times 64^0 \quad (3.20)$$

where,  $0 \leq a_i \leq 63$ . Using just three digits  $a_1a_2a_3$  in such a base, the maximum value one can obtain is  $63 \times 64^2 + 63 \times 64 + 63 = 2^{18} - 1$ . Thus with the help of 64-base, we can cover the entire RF range with just 3 digits. Now we just need to store the  $63 + 63 + 63 = 189$  matrices corresponding the different digits  $a_1a_2a_3$ . Even if there are  $M$  different nuclei, we have to store only  $M \times 189$  matrices which is very less than storing  $M \times 2^{18}$  matrices; as there can be at maximum 4 to 5 different NMR active nuclear species in a given molecule.

Using such a base of 64, any RF amplitude  $\Omega_k$  between 0 and  $2^{18}$  can be written as:

$$\Omega_k = a_1 \times 64^2 + a_2 \times 64^1 + a_3 \times 64^0 \quad (3.21)$$

The matrix exponential (L.H.S) we wish to evaluate can now be split as:

$$\exp \left( -i\Delta t \sum_{k=1}^m \Omega_k(t) \tilde{I}_x \right) = \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \quad (3.22)$$

where,

$$\mathcal{A}_1 = \exp \left( -i\Delta t \sum_{k=1}^m (a_{1k}(t) \times 64^2) \tilde{I}_x \right) \quad (3.23)$$

$$\mathcal{A}_2 = \exp \left( -i\Delta t \sum_{k=1}^m (a_{2k}(t) \times 64^1) \tilde{I}_x \right) \quad (3.24)$$

$$\mathcal{A}_3 = \exp \left( -i\Delta t \sum_{k=1}^m (a_{3k}(t) \times 64^0) \tilde{I}_x \right) \quad (3.25)$$

Here, all the matrices  $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$  are calculated only once at the beginning of the algorithm and stored. Thus, we do not have to perform any matrix exponentiations in the runtime of the algorithm. We only have to call the stored matrices  $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$  corresponding to the RF amplitude  $\Omega_k = 'a_1 a_2 a_3'$  and perform 2 matrix multiplications  $-\mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3$  for getting the required matrix exponential. Using the interaction picture of quantum mechanics along with coarse graining and matrix recycling, we now present a method for matrix exponentiation which can be incorporated in GRAPE and Krotov optimal control algorithms.

### 3.4 Fast matrix exponentiation

The propagator we wish to evaluate is:

$$U_j = \exp \left[ -i\Delta t \left( \mathcal{H}_0 + \sum_{k=1}^m u_k(j) \mathcal{H}_k \right) \right] \quad (3.26)$$

The above propagator is in the Schrödinger picture. In the interaction picture, the same propagator is:

$$\tilde{U}_j = e^{(-i\sum_{k=1}^m \theta_k(j) I_z)} \exp \left( -i\Delta t \sum_{k=1}^m \Omega_k(j) \tilde{I}_x \right) e^{(i\sum_{k=1}^m \theta_k(j) I_z)} \quad (3.27)$$

As the dynamics of our optimal control algorithms GRAPE and Krotov are in the Schrödinger, the above propagator needs to be taken back in the Schrödinger frame, i.e.  $U_j = e^{-i\mathcal{H}_0\Delta t} \tilde{U}_j$ . In the following, we present a method to calculate the propagator:

$$U_j = e^{-i\mathcal{H}_0\Delta t} \left( e^{(-i\sum_{k=1}^m \theta_k(j) I_z)} \exp \left( -i\Delta t \sum_{k=1}^m \Omega_k(j) \tilde{I}_x \right) e^{(i\sum_{k=1}^m \theta_k(j) I_z)} \right) \quad (3.28)$$

For simplicity, we assume that only one type of NMR active nuclei is present in the molecule (homonuclear). This method can be easily extended to heteronuclear NMR systems.

- **Precalculating and storing**

1. Calculate the internal Hamiltonian  $\mathcal{H}_0$  and store the propagator:

$$U_0 = \exp(-i\mathcal{H}_0\Delta t) \quad (3.29)$$

2. Transform the spin operator  $I_x$  from Schrödinger basis to interaction frame as follows:

$$\tilde{I}_x = U_0^\dagger I_x U_0 \quad (3.30)$$

3. For all  $a_1, a_2, a_3$  such that  $0 \leq a_1 \leq 63$ ,  $0 \leq a_2 \leq 63$  and  $0 \leq a_3 \leq 63$  calculate and store  $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$  such that:

$$\mathcal{A}_1 = \exp\left(-i\Delta t(a_1 \times 64^2)\tilde{I}_x\right) \quad (3.31)$$

$$\mathcal{A}_2 = \exp\left(-i\Delta t(a_2 \times 64^1)\tilde{I}_x\right) \quad (3.32)$$

$$\mathcal{A}_3 = \exp\left(-i\Delta t(a_3 \times 64^0)\tilde{I}_x\right) \quad (3.33)$$

- **Runtime**

For homonuclear systems, the propagator during the  $j^{\text{th}}$  segment is:

$$U_j = e^{-i\mathcal{H}_0\Delta t} \left( e^{(-i\theta(j)I_z)} \exp\left(-i\Delta t\Omega(j)\tilde{I}_x\right) e^{(i\theta(j)I_z)} \right) \quad (3.34)$$

In order to evaluate this –

1. Find the decomposition of amplitude  $\Omega(j)$  into digits  $(a_1, a_2, a_3)$  such that

$$\Omega(j) = a_1 \times 64^2 + a_2 \times 64^1 + a_3 \times 64^0 \quad (3.35)$$

2. Choose the stored matrices  $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$  corresponding to the digits  $(a_1, a_2, a_3)$  found out in previous step and hence evaluate the product

$$\mathcal{A}_j = \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \quad (3.36)$$

3. Calculate the phase  $\theta(j) = \tan^{-1}(u_y(j)/u_x(j))$  and hence compute the rotation:

$$\mathcal{Z}_j = e^{(-i\theta(j)I_z)} \quad (3.37)$$

4. The required propagator is:

$$U_j = U_0(Z_j A_j Z_j^\dagger) \quad (3.38)$$

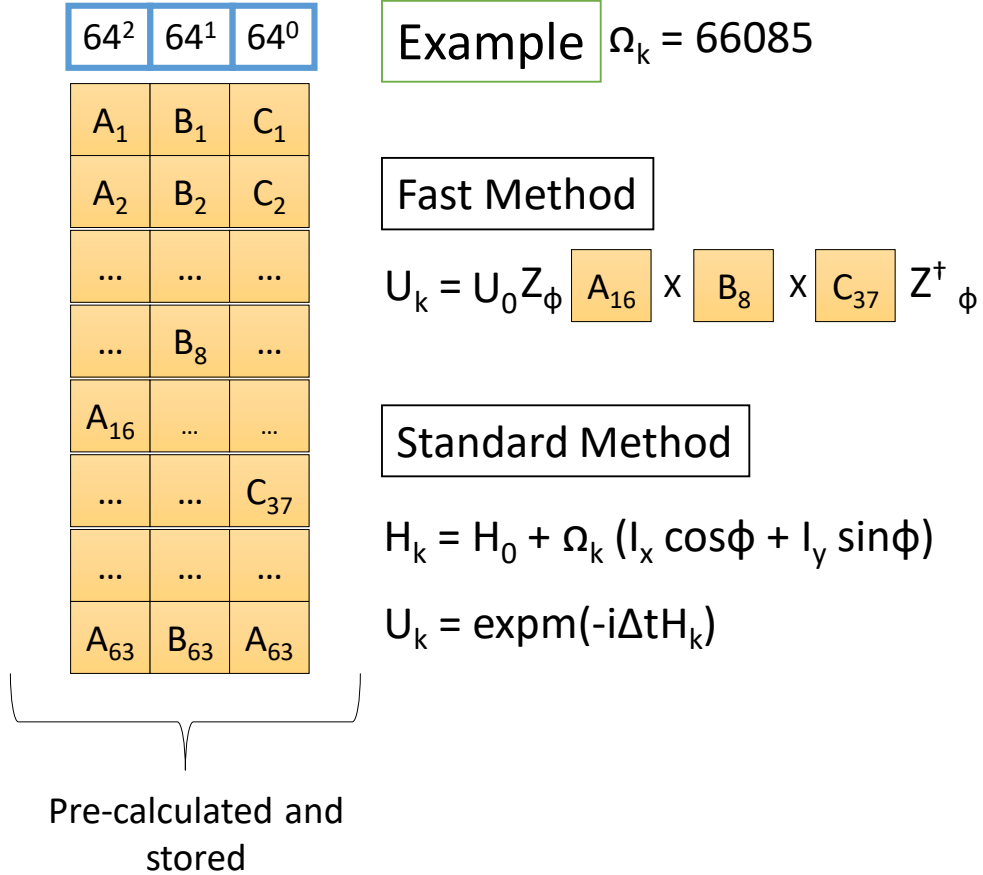


Figure 3.2: Example of evaluating propagator using standard method and REDO (fast)method.

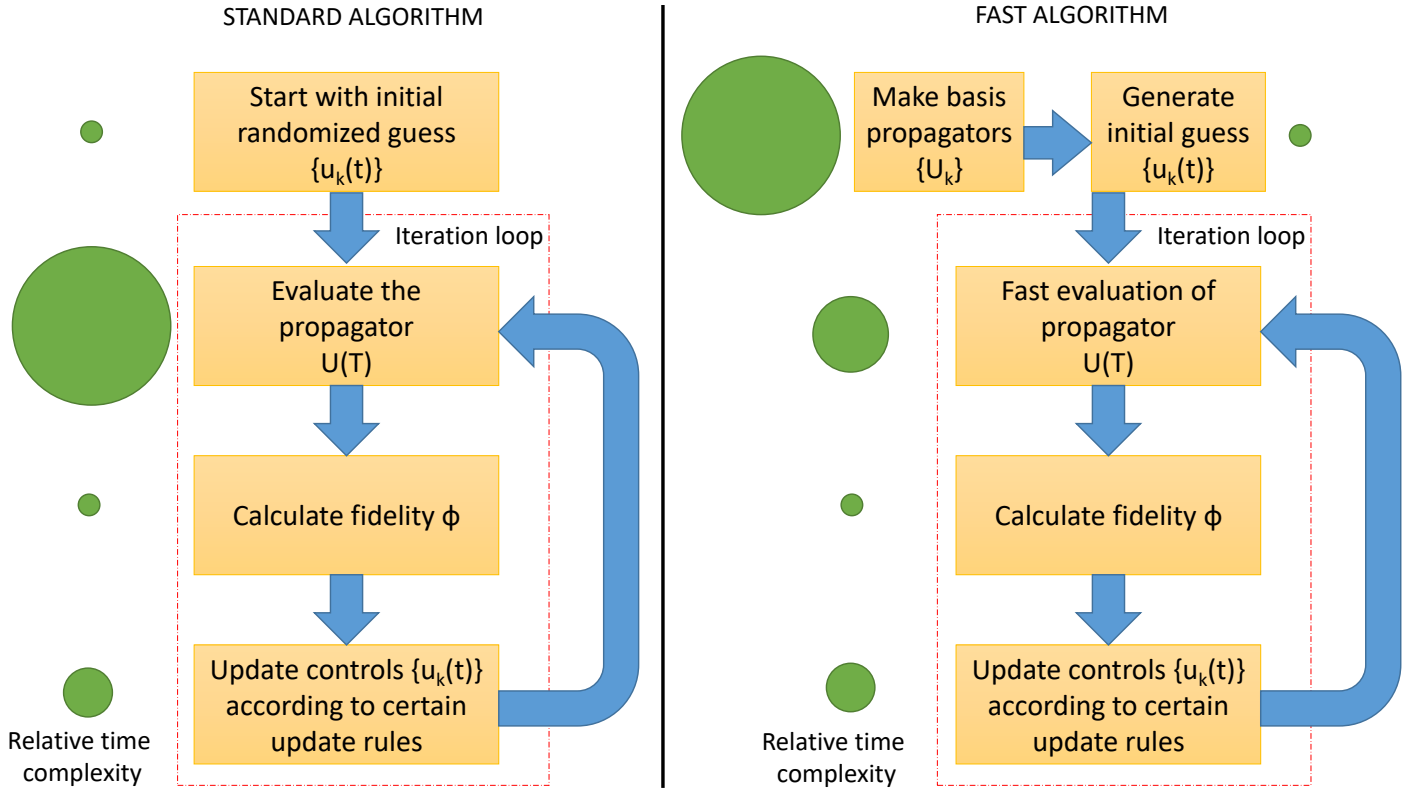
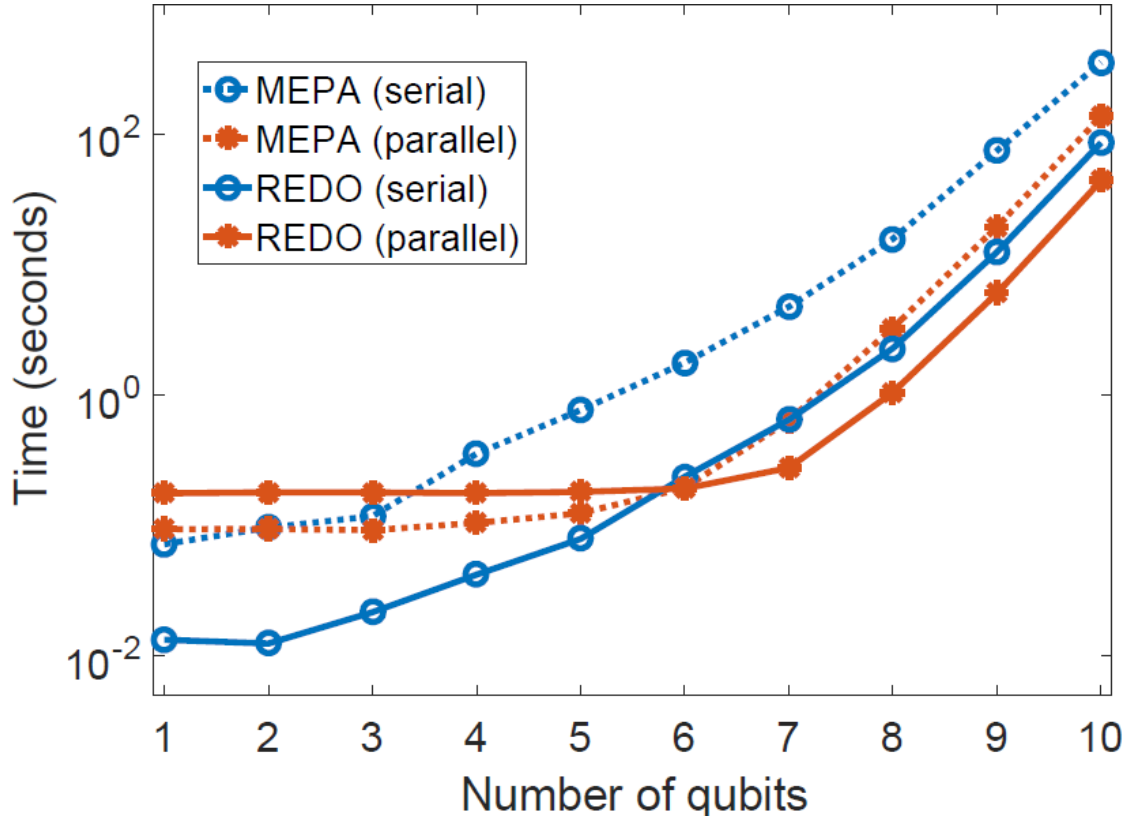


Figure 3.3: Shown on the left is a scheme for optimal control algorithms which rely on the use of standard matrix exponentiation in the iteration loop. Shown on the right is an optimal control scheme which uses the REDO (fast) method for matrix exponentiation. The basis propagators which use standard matrix exponentiation are calculated only once outside the iteration loop. Within the iteration loop, the fast optimal control algorithm utilizes the pre-calculated and stored basis propagators to evaluate any general propagator by matrix multiplication (which is computationally inexpensive compared to matrix exponentiation).

### 3.5 Comparison and discussion

Here, we compare the usual method of matrix exponentiation used in MATLAB versus fast matrix exponentiation (REDO). MATLAB makes the use of Pade approximation for the calculation of matrix exponentiation. For comparison, the matrix exponential using the *expm* function in MATLAB which we abbreviate MEPA(Matrix Exponentiation by Pade Approximation) was used for evaluating  $U_j$ . The time required for 1000 such matrix exponentials with random RF amplitudes was noted. Moreover, these 1000 matrix exponentials were performed in serial as well as parallel (using *parfor* routine) in MATLAB. This is analogous to finding the forward propagator in optimal control algorithms with  $N = 1000$  segments. Also, each simulation was repeated 10 times; analogous to iterations in the optimal control algorithm. This procedure was done for a system of  $\{1, 2, \dots, 10\}$  qubits – a representative of dimension of Hilbert space or matrix dimension. A similar protocol was followed for fast matrix exponentiation(REDO) – in serial as well as parallel. The result of the comparison are shown in Fig3.4

We can see that, for smaller size of quantum systems (1 – 5) qubits, the fast matrix exponentiation (REDO) is several times faster than the usual method of matrix exponentiation (MEPA). Parallel fast matrix exponentiation (parallel REDO) is slower for small sizes of quantum systems because the time required for sending the data to the parallel computing workers is comparable to the actual time required for computing the fast matrix exponential. However, as the size of the quantum system increases, parallel fast matrix exponentiation (parallel REDO) takes over and is significantly faster (2 – 3 times) than the usual method of matrix exponentiation(MEPA). Thus, we can use fast matrix exponentiation without the parallel computing feature for small number of qubits (1 – 5) and then switch to the parallel computing feature for higher number of qubits (6 – 10). With this, our method of fast matrix exponentiation(REDO) is about 3 times faster on an average than the usual method for matrix exponentiation (MEPA).



Qubits	MEPA (parallel) / REDO (serial)	MEPA (parallel) / REDO (parallel)	MEPA (serial) / REDO (serial)
1	7.0467	0.5312	5.4084
2	7.5553	0.5235	7.7509
3	4.2572	0.5141	5.4473
4	2.491	0.5914	8.4507
5	1.5727	0.6854	9.7292
6	0.8236	1.0142	7.4952
7	0.9866	2.3105	7.3513
8	1.4323	3.1205	6.8971
9	1.5678	3.2413	5.997
10	1.5894	3.1179	4.1265

Figure 3.4: Comparison of MEPA and REDO methods in serial and parallel. The speed-up for REDO method over the usual MEPA is shown in the table. The table suggests that we must use the parallel REDO method for larger quantum systems with 6-10 qubits and REDO serial method for smaller quantum systems with 1-5 qubits.





# Chapter 4

## Experiments in NMR using quantum control

### 4.1 Applications in quantum information processing

Here, we use the quantum control algorithms described previously to experimentally implement quantum gates on spin-1/2 qubits in the NMR setting. We will use  $^{13}\text{C}$  carbons of the molecule Alanine in  $\text{D}_2\text{O}$  as a 3-qubit quantum register. The structure and the Hamiltonian of the molecule is shown below. All the experiments were carried out in a 600 MHz BRUKER NMR spectrometer at an ambient temperature of 298K.

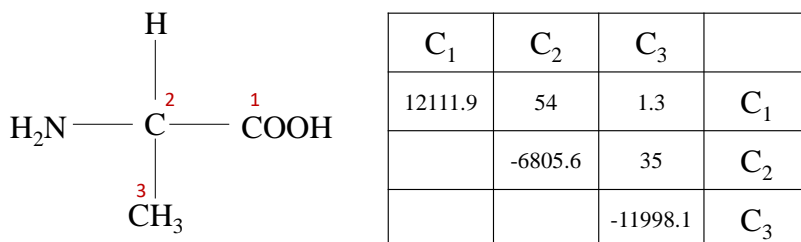


Figure 4.1: Shown on the left is the structure of Alanine. The 3-qubit quantum register is formed by the three carbons labelled in red. Also shown on the left is the Hamiltonian of the molecule in a 600 MHz NMR spectrometer. The diagonal elements denote the chemical shifts whereas the off-diagonal elements are the J-couplings (in Hz.) between the 3-qubits.

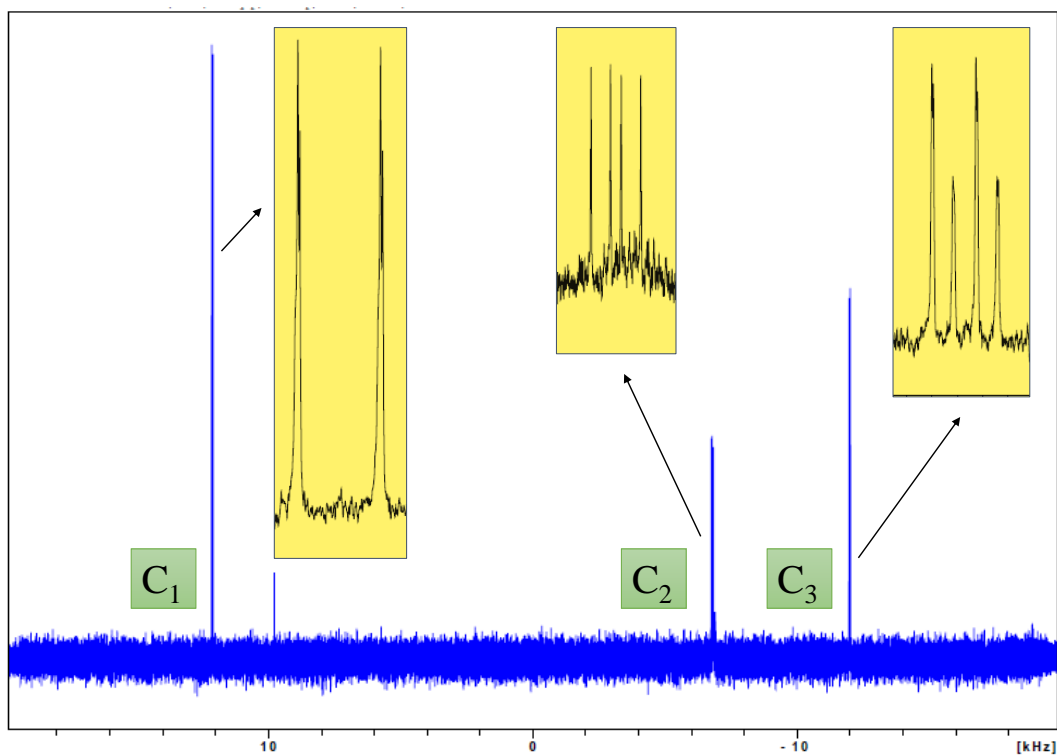


Figure 4.2: The thermal equilibrium spectra of Alanine. Shown in the inset is the zoomed in spectra of each of the  $^{13}\text{C}$  carbons.

### 4.1.1 Performing selective inversions

Here, we want to selectively invert one of the transitions of carbon in Alanine without affecting the other transitions. This is equivalent to applying a  $\Pi$  pulse on one of the qubits with  $I$  on other qubits. For this, we use the iGRAPE algorithm described in the previous chapters for designing the shaped RF controls. A typical input file for running the iGRAPE algorithm in MATLAB is as follows:

```
%% Algorithm Specific Parameters
epsilon = 5e+7;

%% Hamiltonian Parameters
% Spins
spinList = 3;
spinNumbers = 1/2;
% Default Parameters
defaultparams;
% Chemical Shifts and Couplings
v(1) = 12111.9;
v(2) = -6805.6;
v(3) = -11998;
J(1,2) = 54;
J(1,3) = 2;
J(2,3) = 35;

%% Input Parameters for Pulse Sequence
N = 100; % Number of segments
dt = 5e-6; % Time duration of each segment in seconds
% Initial Random Guess for Pulse Sequence
max_initRF = 8e3;
w = gen_initRF(max_initRF, 5);
% RF Inhomogeneties
rfin_wt = [0.25 0.50 0.25]';
rfin_range = [0.9 1.0 1.1]';

%% Target Unitary
Utarget = expm(-1i*pi*Ix(:,:,1));

%% Target Fidelity and Maximum Iterations
target_fidelity = 0.99;
max_iter = 1000;
optim_type = 'unitary';
```

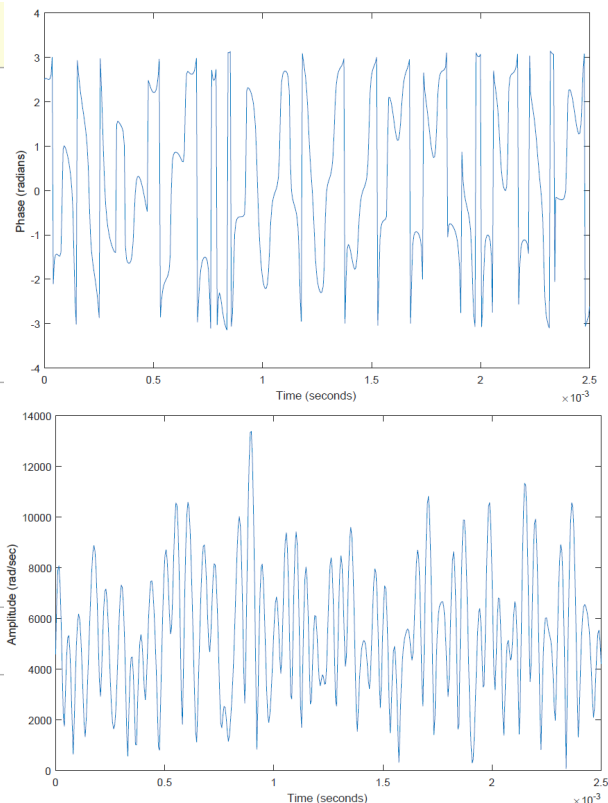


Figure 4.3: Input file for iGRAPE algorithm is shown on the left. The output of the GRAPE algorithm is shown on the right. On the top right, shown is the time-modulated RF amplitude profile and on the bottom right shown is the corresponding phase.

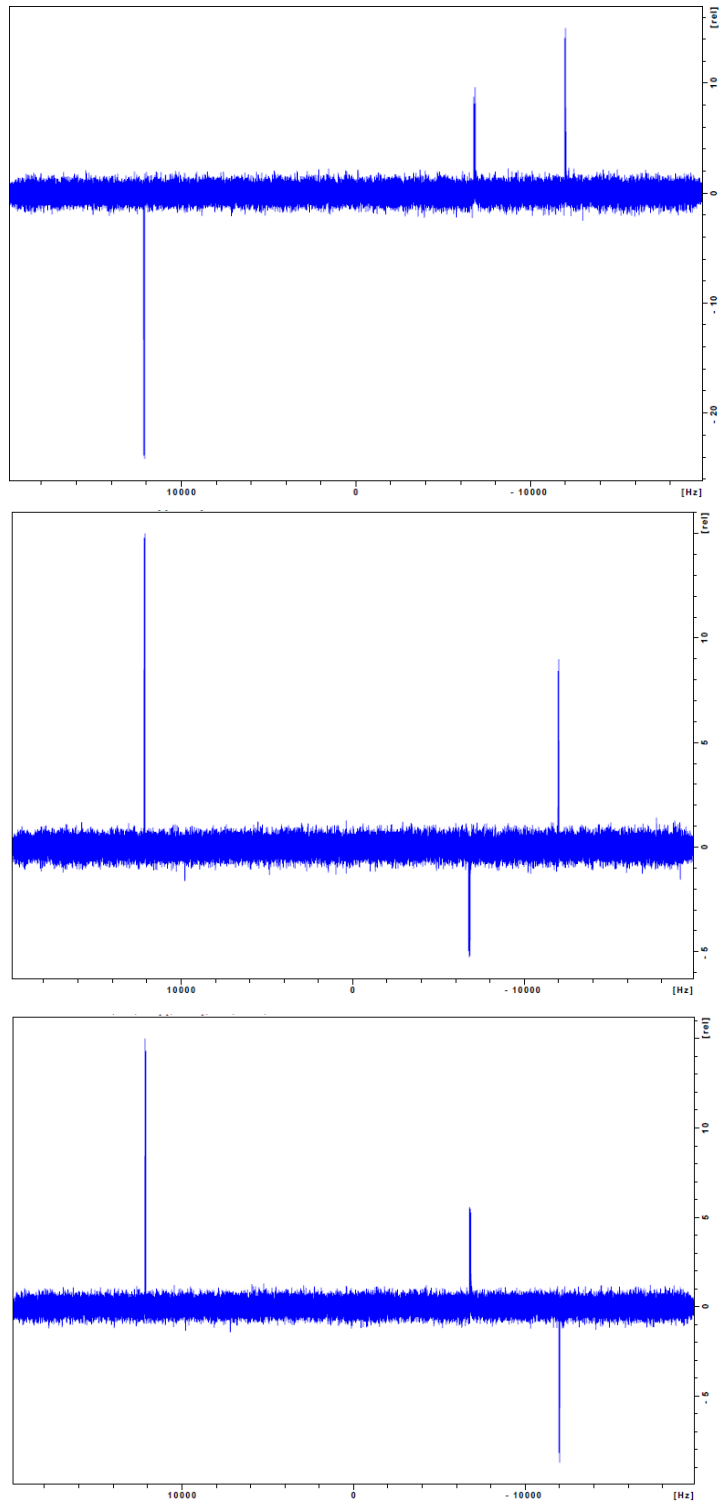


Figure 4.4: Spectra obtained after selectively inverting each qubit in Alanine.

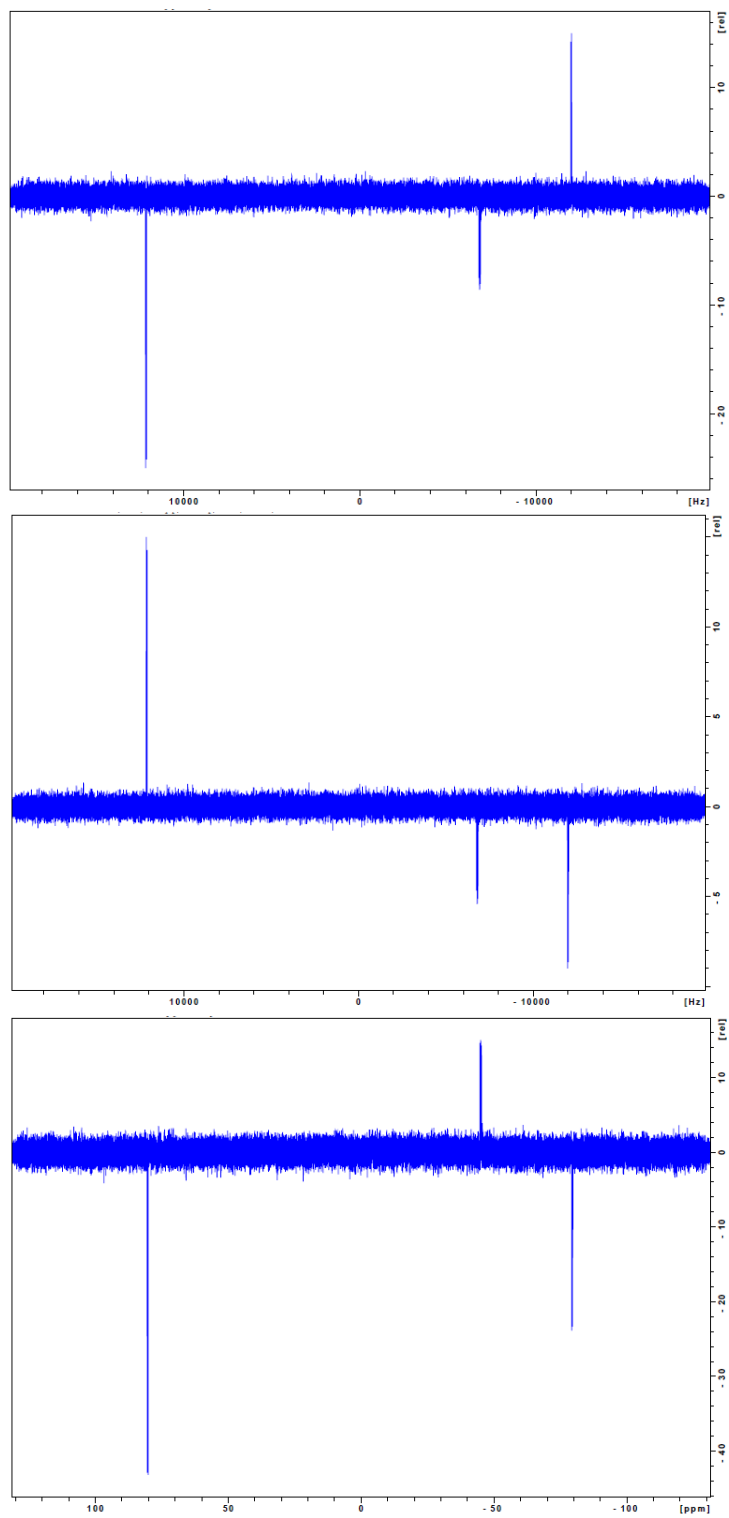


Figure 4.5: Spectra obtained after selectively inverting a pair of qubits in Alanine.

### 4.1.2 Initializing the quantum register

The thermal equilibrium state of the 3-qubit register is represented by the density matrix  $\rho_{eq} = (\mathcal{I} + \epsilon_C I_{zC})/8$ , where  $\epsilon_C \sim 10^{-5}$  is called the purity factor and  $I_z$  is the collective spin operator. This state is highly mixed. We need to create a pseudo-pure state (PPS) which mimics a pure state  $|000\rangle$  for initializing the quantum register to implement any algorithms. The circuit for realizing PPS in 3-qubit homo-nuclear NMR spin systems[15] is as follows:

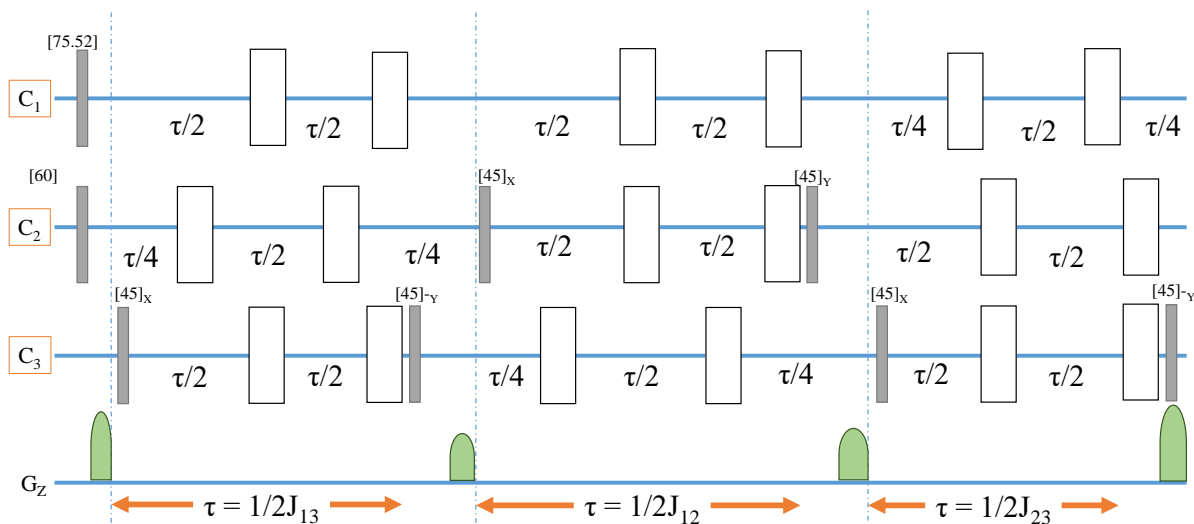


Figure 4.6: The quantum circuit for creating PPS.

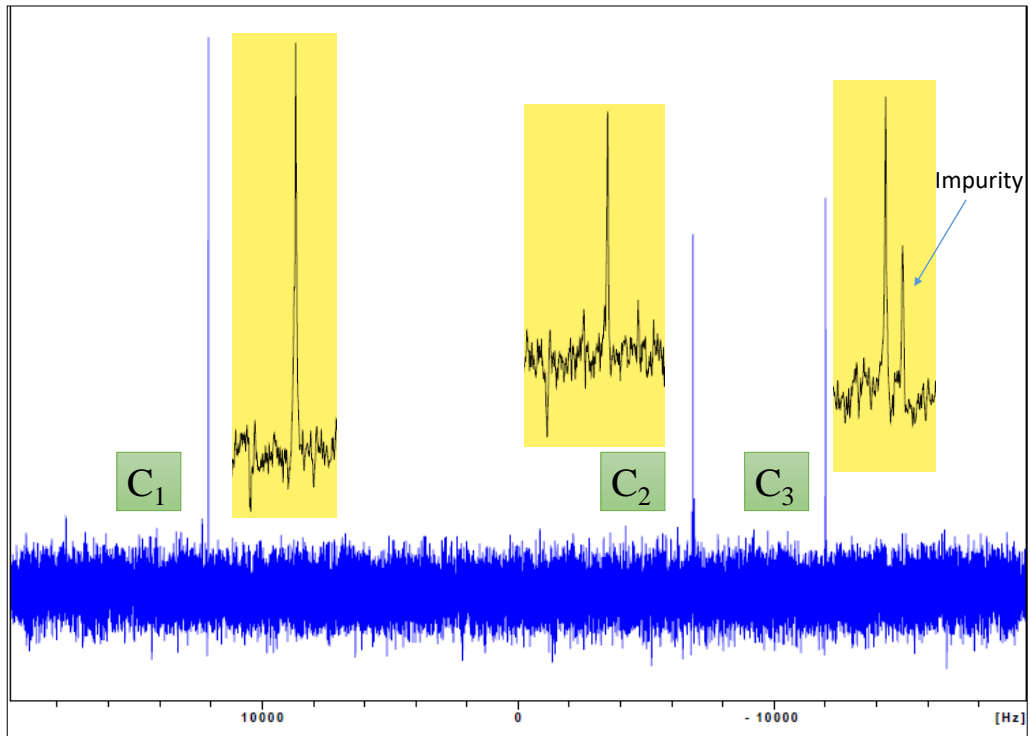


Figure 4.7: The spectra obtained after applying the PPS circuit. Note that there is only a single peak corresponding to every qubit after obtaining the pseudo-pure state which mimics the  $|000\rangle$  state.

### 4.1.3 Super-adiabatic quantum state transfer in spin chains

Here, we implement the super-adiabatic quantum state transfer protocol(SAQST) as described in [9]. The SAQST protocol for an odd-numbered spin chains for 3 spins is described below.

- Consider the Hamiltonian for the 3-spin chain with  $J_L$  and  $J_R$  as the couplings between 1<sup>st</sup> and 2<sup>nd</sup> spin and 2<sup>nd</sup> and 3<sup>rd</sup> spin respectively.
- The Hamiltonian for such a system is:

$$\mathcal{H}(t) = \tilde{J}_L \left( \sum_{i=x,y,z} \sigma_1^i \otimes \sigma_2^i \right) + \tilde{J}_R \left( \sum_{i=x,y,z} \sigma_2^i \otimes \sigma_3^i \right) \quad (4.1)$$

where,

$$\tilde{J}_L(t) = \left( J_L \sec \theta_R, J_L \sec \theta_R, J_L + \frac{1}{2} \frac{\partial \theta_L}{\partial t} \right) \quad (4.2)$$

$$\tilde{J}_R(t) = \left( J_R \sec \theta_L, J_R \sec \theta_L, J_R + \frac{1}{2} \frac{\partial \theta_R}{\partial t} \right) \quad (4.3)$$

$$J_L(t) = J_M \sin^2 \left( \frac{\pi t}{2T} \right) \quad (4.4)$$

$$J_R(t) = J_M - J_L(t) \quad (4.5)$$

$$\tan \theta_L = \frac{J_M \frac{\partial J_R}{\partial t}}{4(2J_M - 2J_L)(3J_L^2 + J_M^2 - 3J_M J_L)} \quad (4.6)$$

$$\tan \theta_R = \frac{-J_M \frac{\partial J_L}{\partial t}}{8J_L(3J_L^2 + J_M^2 - 3J_M J_L)} \quad (4.7)$$

$$(4.8)$$

Here,  $T$  is the total time of adiabatic evolution and  $\Delta = 2J_M$  is the effective energy gap between the ground state and excited state of the Hamiltonian.

- If we want to transport the state  $\alpha |0\rangle + \beta |1\rangle$  from the 1<sup>st</sup> spin to the 3<sup>rd</sup> spin, we encode  $\alpha |G_0\rangle + \beta |G_1\rangle$  where  $|G_0\rangle$  and  $|G_1\rangle$  are the degenerate ground states of the Hamiltonian  $\mathcal{H}(t = 0)$ .
- Now, we adiabatically evolve the Hamiltonian  $\mathcal{H}(t = 0)$  to  $\mathcal{H}(t = T)$  by varying the couplings  $\tilde{J}_L(t)$  and  $\tilde{J}_R(t)$  to transport any state  $\alpha |0\rangle + \beta |1\rangle$  from the 1<sup>st</sup> spin to 3<sup>rd</sup>.



The results for transporting the state  $\alpha |0\rangle + \beta |1\rangle$  with  $\alpha = 1/\sqrt{2}$  and  $\beta = 1/\sqrt{2}$  are shown below. We have performed the adiabatic evolution in 5 steps.

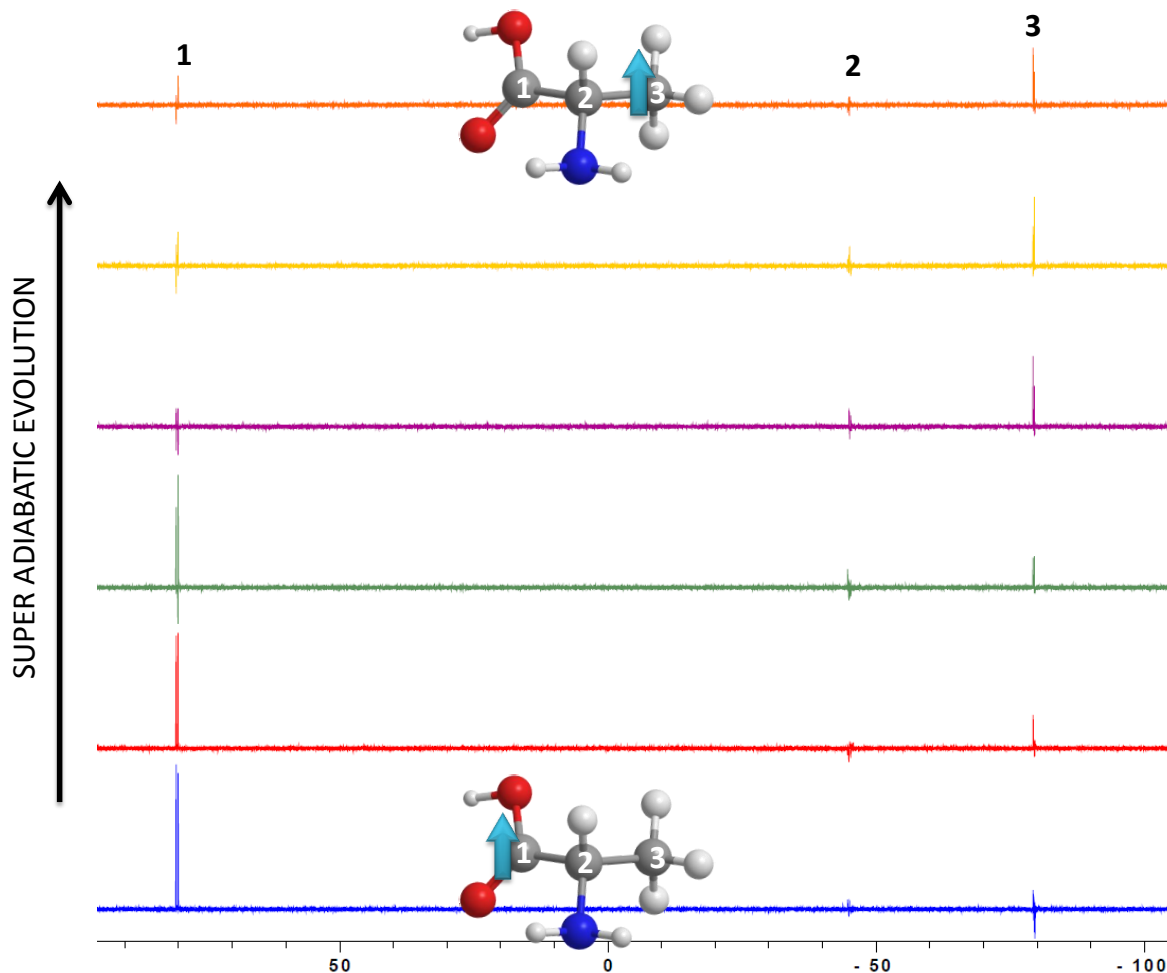


Figure 4.8: Experimental realisation of SAQST on 3-spin chain. The bottom-most spectra is recorded after encoding the state  $1/\sqrt{2} |0\rangle + 1/\sqrt{2} |1\rangle$ . Note that the first spin has maximum polarisation whereas the other two are nearly zero. From bottom to top shown is the spectra at each step of adiabatic evolution. In the top most spectra, we can see that the first spin has negligible polarisation compared to the third spin, implying that the state  $1/\sqrt{2} |0\rangle + 1/\sqrt{2} |1\rangle$  has been transferred to the 3<sup>rd</sup> spin.

## 4.2 Applications in spectroscopy

Here, we design single and multi-band selective inversion pulses using GRAPE. Such pulses are used to selectively invert (equivalently apply a  $\Pi$  pulse) peaks lying within a particular bandwidth without affecting peaks at any other frequencies (equivalent of an  $\mathbb{I}$  operation).

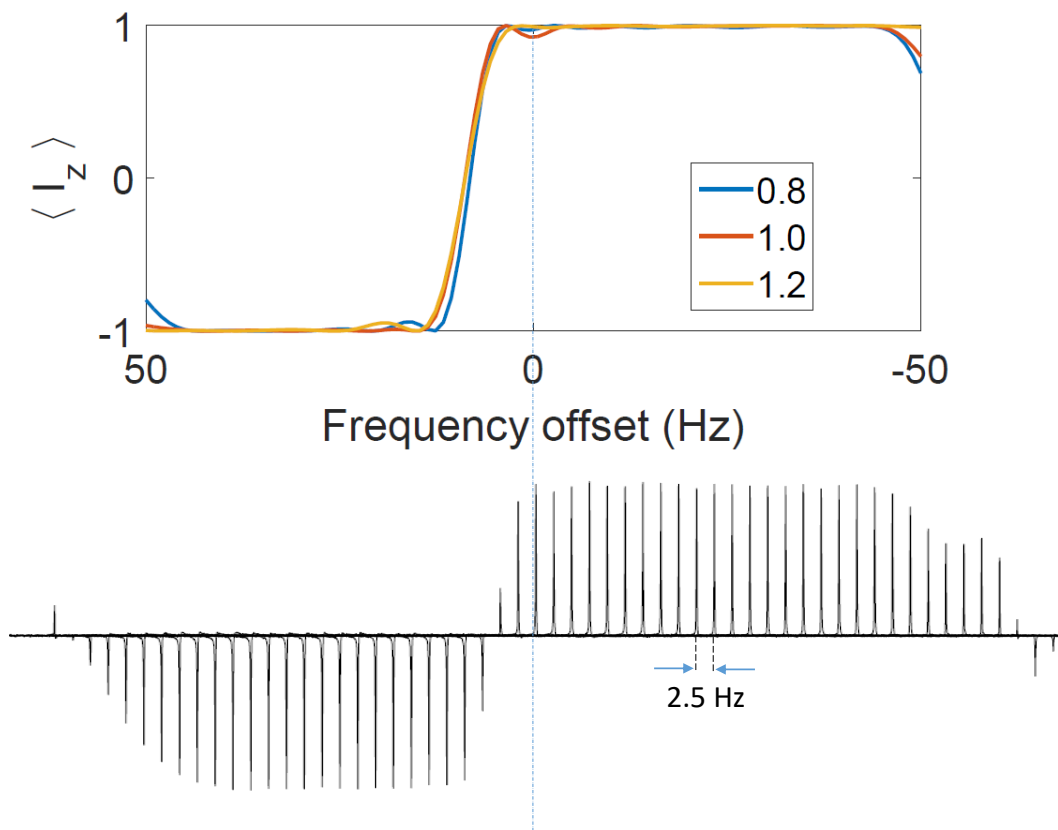


Figure 4.9: Top trace shows the inversion profile of the designed single band selective inversion iGRAPE pulse. To illustrate its robustness w.r.t. RF inhomogeneity, three inversion profiles with 0.8, 1.0, and 1.2 times the nominal RF amplitudes are displayed. The bottom trace shows how a particular peak is affected by the iGRAPE pulse when it is applied at various frequency offsets.

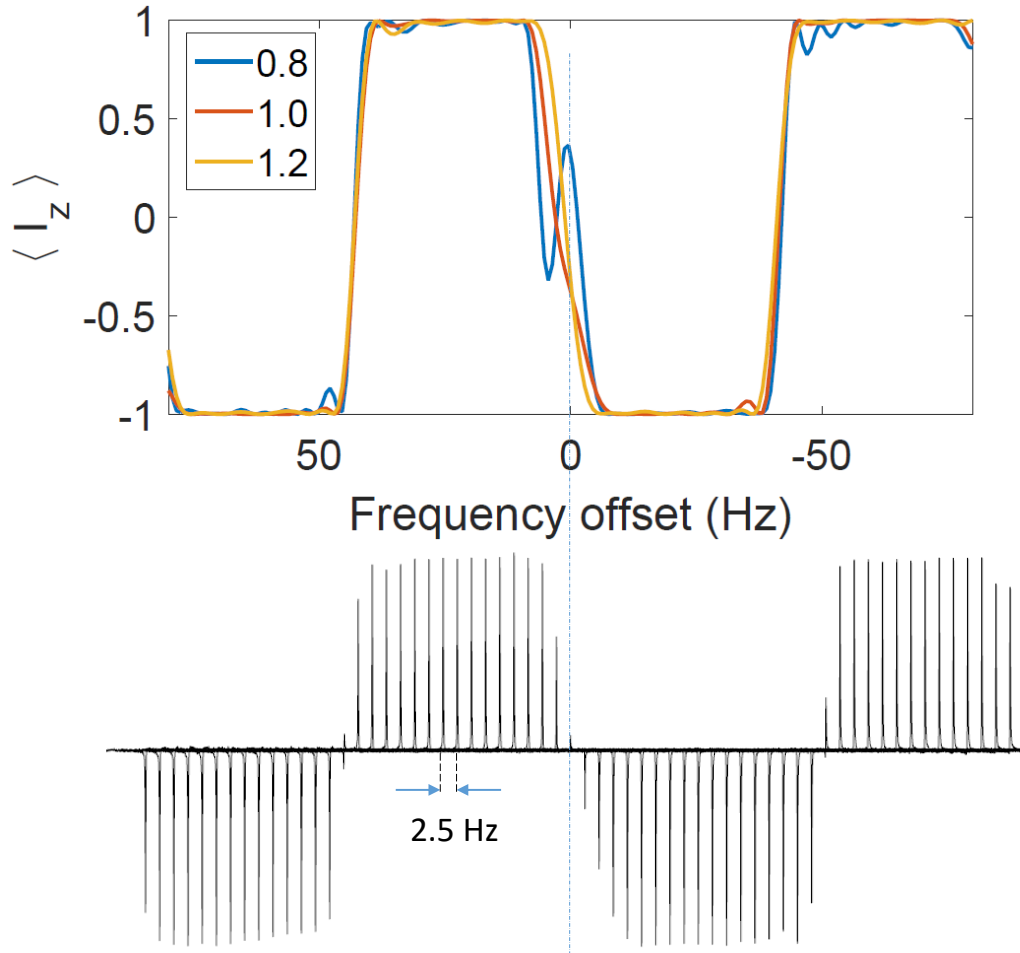


Figure 4.10: Top trace shows the inversion profile of the designed multi-band selective inversion iGRAPE pulse. To illustrate its robustness w.r.t. RF inhomogeneity, three inversion profiles with 0.8, 1.0, and 1.2 times the nominal RF amplitudes are displayed. The bottom trace shows how a particular peak is affected by the iGRAPE pulse when it is applied at various frequency offsets.

Such bandwidth limited selective inversion pulses have been recently applied for heat-bath algorithmic cooling in large star-topology quantum registers[10]. Heat-bath algorithmic cooling is used to enhance the polarisation of low natural abundance nuclear isotopes.

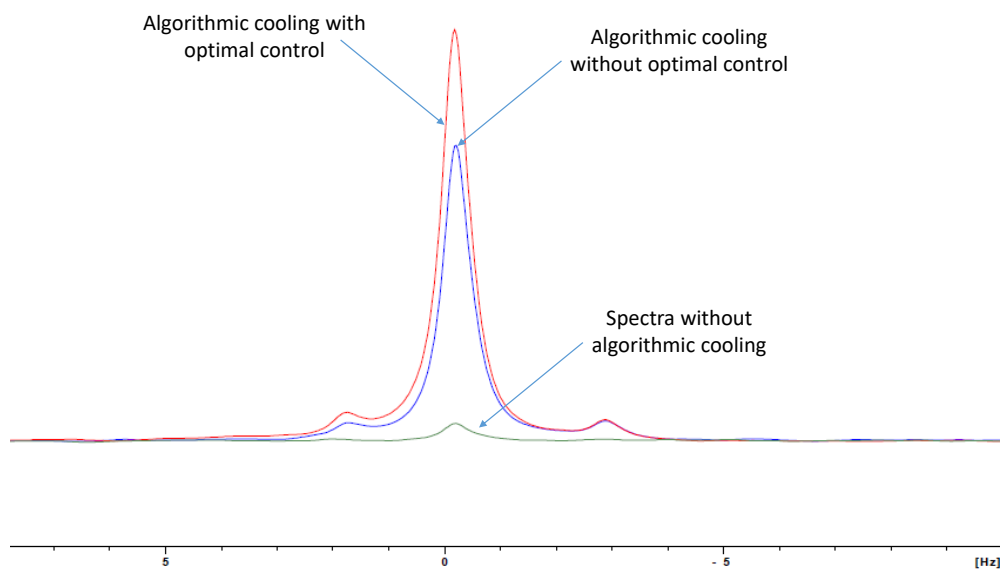


Figure 4.11: Shown here is the  $^1\text{H}$ -decoupled  $^{29}\text{Si}$  spectra of tetrakis(trimethylsilyl)silane – before algorithmic cooling (green), after algorithmic cooling (red and blue). The spectra in blue was obtained using standard NMR techniques while one in red was obtained using optimal control techniques. Clearly, the increase in magnetisation after algorithmic cooling is much higher in experiments performed using optimal control techniques than standard NMR techniques.

# Chapter 5

## Summary and Conclusions

- In this thesis, we studied how to control the dynamics of a quantum system using amplitude modulated electromagnetic pulses. Our focus was on improving the efficiency of the numerical optimization algorithms which are used to design these amplitude modulated electromagnetic pulses.
- In particular, we applied these numerical optimization algorithms to design shaped radio-frequency pulses which control the dynamics of spin-1/2 nuclei in NMR.
- We studied and implemented two well known numerical algorithms for quantum control, namely GRAPE (Gradient Ascent Pulse Engineering) and Krotov.
- GRAPE algorithm involves a concurrent update and is a local optimization algorithm. On the other hand, Krotov involves sequential update and is a global optimization algorithm.
- Krotov being a global optimization algorithm, it can rapidly make coarse adjustments to the controls and reach high fidelities within a few iterations. On the contrary, GRAPE is a local optimization algorithm and needs more iterations to reach higher fidelities. However, once high fidelities are attained, GRAPE can apply finer adjustments to the controls efficiently than Krotov.
- This contrasting nature of GRAPE and Krotov algorithms allows us to use a hybrid algorithms wherein one utilizes Krotov for initial few iterations to attain a certain

threshold fidelity and from there on switches to GRAPE to make finer adjustments to controls and thereby reach higher fidelities.

- Repeated evaluation of propagators which involves matrix exponentiation is the computational bottle-neck of these optimal control algorithms. We devised a novel method named REDO (**R**apid **E**xponentiation by **D**iscrete **O**perators), to speed up the matrix exponentiation specifically in optimal control algorithms making use of coarse graining, matrix recycling and interaction picture of quantum mechanics.
- We benchmarked our REDO method against the standard method of matrix exponentiation in MATLAB and observed a speed-up of nearly 2 – 3 times in the computation time.
- We incorporated the REDO method of matrix exponentiation in GRAPE and Krotov optimal control algorithms (named iGRAPE and iKrotov respectively) and used these algorithms for experimental applications in NMR quantum information processing and spectroscopy.
- Alanine was used as a 3-qubit NMR quantum register. Firstly, we initialized the 3-qubit quantum register to the  $|000\rangle$  pseudo-pure state. Further, we demonstrated the first experimental implementation of the super-adiabatic quantum state transfer protocol wherein a known quantum state was super-adiabatically transferred from one end of the spin chain to the other end.
- As an application in spectroscopy, we designed single and multi band-selective inversion pulses to selectively invert or equivalently apply a  $\Pi$  pulse within a particular range of frequencies without affecting any other frequencies. Such band-selective inversion pulses were further used in heat-bath algorithmic cooling protocol to increase the polarisation of low natural abundance nuclear isotopes.
- We observed that the controls designed using optimal control algorithms deliver a much better performance than standard NMR techniques.

# Bibliography

- [1] J. A. Jones, “Quantum computing and nuclear magnetic resonance,” *PhysChemComm*, vol. 4, no. 11, pp. 49–56, 2001.
- [2] D. P. Divincenzo, “The Physical Implementation of Quantum Computation,” *Fortschritte der Physik*, vol. 48, pp. 771–783, 2000.
- [3] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, “Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms,” *Journal of magnetic resonance*, vol. 172, no. 2, pp. 296–305, 2005.
- [4] T. Caneva, T. Calarco, and S. Montangero, “Chopped random-basis quantum optimization,” *Physical Review A*, vol. 84, no. 2, p. 022326, 2011.
- [5] S. Hou, L. Wang, and X. Yi, “Realization of quantum gates by lyapunov control,” *Physics Letters A*, vol. 378, no. 9, pp. 699–704, 2014.
- [6] E. M. Fortunato, M. A. Pravia, N. Boulant, G. Teklemariam, T. F. Havel, and D. G. Cory, “Design of strongly modulating pulses to implement precise effective hamiltonians for quantum information processing,” *The Journal of chemical physics*, vol. 116, no. 17, pp. 7599–7606, 2002.
- [7] I. I. Maximov, Z. Tošner, and N. C. Nielsen, “Optimal control design of nmr and dynamic nuclear polarization experiments using monotonically convergent algorithms,” *The Journal of Chemical Physics*, vol. 128, no. 18, p. 05B609, 2008.
- [8] G. Bhole, V. Anjusha, and T. Mahesh, “Steering quantum dynamics via bang-bang control: Implementing optimal fixed-point quantum search algorithm,” *Physical Review A*, vol. 93, no. 4, p. 042339, 2016.
- [9] R. R. Agundez, C. D. Hill, L. C. L. Hollenberg, S. Rogge, and M. Blaauuboer, “Superadiabatic quantum state transfer in spin chains,” *Phys. Rev. A*, vol. 95, p. 012317, Jan 2017.
- [10] V. R. Pande, G. Bhole, D. Khurana, and T. Mahesh, “Strong algorithmic cooling in large star-topology quantum registers,” *arXiv preprint arXiv:1702.04992*, 2017.

- [11] D. G. Lucarelli, “Quantum optimal control via gradient ascent in function space and the time-bandwidth quantum speed limit,” *arXiv preprint arXiv:1611.00188*, 2016.
- [12] D. Goodwin and I. Kuprov, “Auxiliary matrix formalism for interaction representation transformations, optimal control, and spin relaxation theories,” *The Journal of chemical physics*, vol. 143, no. 8, p. 084113, 2015.
- [13] D. Goodwin and I. Kuprov, “Modified newton-raphson grape methods for optimal control of spin systems,” *The Journal of chemical physics*, vol. 144, no. 20, p. 204107, 2016.
- [14] M. H. Levitt, *Spin dynamics: basics of nuclear magnetic resonance*. John Wiley & Sons, 2001.
- [15] A. Mitra, T. Mahesh, and A. Kumar, “Nmr implementation of adiabatic sat algorithm using strongly modulated pulses,” *The Journal of chemical physics*, vol. 128, no. 12, p. 124110, 2008.