JAVA APPLETS FOR SIMULATION OF MAGNETIC RESONANCE IMAGING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÇAĞDAŞ ALTIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2008

Approval of the Thesis:

JAVA APPLETS FOR SIMULATION OF MAGNETIC RESONANCE IMAGING

submitted by **ÇAĞDAŞ ALTIN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural Applied Sciences**

Prof. Dr. İsmet Erkmen _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Nevzat G. Gençer _____
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members

Prof. Dr. Murat Eyüboğlu _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Nevzat G. Gençer _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Ziya İder _____
Electrical and Electronics Engineering Dept., Bilkent University

Assist. Prof. Dr. İlkay Ulusoy _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Yeşim Serinağaoğlu _____
Electrical and Electronics Engineering Dept., METU

Date : 02 / 12 / 2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Çağdaş Altın

Signature :

ABSTRACT

JAVA APPLETS FOR SIMULATION OF
MAGNETIC RESONANCE IMAGING

Altın, Çağdaş
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Nevzat G. Gençer

December 2008, 88 pages

The aim of this study is to develop an easily accessible and realistic magnetic resonance imaging (MRI) simulation tool for educational and research purposes. With this aim, NMR (nuclear magnetic resonance imaging) phenomenon is simulated based on the physical principles, starting from the motion of a spin under the influence of external magnetic fields to pulse sequences generating the image. The inputs of the simulation are a 3D virtual object and a pulse sequence definition. The simulation software generates slice images using Fourier reconstruction method. To perform a more realistic simulation, the Bloch equation, which explains the behavior of a spin under external magnetic fields, is solved by using numerical methods. This enables to observe the behavior of a spin system under any magnetic field influence, not only for resonance condition. The simulation successfully simulates T_2^* affect by using inhomogeneous static magnetic field distribution over the entire volume of the object. The software is implemented in Java language and developed as a Java applet. A support tool is developed which allows observing NMR phenomenon. The simulation can produce realistic images, generate many of the artifacts in MRI, like intra-voxel dephasing, chemical shift, cross-talk (since it simulates the whole process), and

has the advantage of being web-based compared to the existing stand-alone MRI simulations.

Keywords: Magnetic resonance imaging, Bloch equation, simulation, Java, applet.

ÖZ

MANYETİK REZONANS GÖRÜNTÜLEME BENZETİMİ
İÇİN JAVA UYGULAMACIKLARI

Altın, Çağdaş
Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Nevzat G. Gençer

Aralik 2008, 88 sayfa

Bu çalışmanın amacı, eğitim ve araştırma amaçlı kullanılabilecek, gerçekçi ve kolay erişilebilir bir manyetik rezonans görüntüleme benzetimi geliştirmektir. Bu amaçla, NMR (nükleer manyetik rezonans) fenomeni, bir spinin dış manyetik alanlar altındaki hareketinden başlayıp, darbe dizileri ile görüntü oluşturulmasına kadar olan bütün süreçleri için, dayandığı fiziksel prensipler kullanılarak benzetilmiştir. 3 boyutlu bir sanal nesne ve bir darbe dizisi tanımı, bezetimin girdileridir. Benzetim yazılımı, Fourier geri dönüşüm tekniğini kullanarak kesit görüntüleri oluşturur. Benzetimin daha gerçekçi sonuçlar üretmesi için, bir spinin manyetik alan altındaki hareketini açıklayan Bloch denklemi, sayısal yöntemlerle çözdürülmüştür. Bu sayede spin sisteminin sadece rezonans durumundaki davranışını değil, her türlü girdi işareti altındaki davranışını görmek mümkün olmuştur. Benzetimde T_2^* etkisi, bütün hacimde homojen olmayan statik manyetik alan dağılımı kullanarak başarılı bir şekilde benzetilmiştir. Benzetim , Java dilinde yazılmıştır ve bir Java uygulamacığı olarak geliştirilmiştir. Ayrıca NMR fenomeninin gözlemlenebilmesi için bir destek yazılımı geliştirilmiştir. Benzetim gerçekçi görüntüler üretebilmekte, bütün işleyişi benzettiği için MR'daki voksel içi faz dağılması, kimyasal kayma, cross-

talk gibi bozulmaları oluşturabilmekte ve hali hazırda tek başına çalışan benzetimlere göre internet üzerinden çalışma avantajına sahiptir.

Anahtar Kelimeler: Manyetik rezonans görüntüleme, Bloch denklemi, benzetim, Java, uygulamacık.

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to my supervisor Assoc. Prof. Dr. Nevzat Güneri Gençer for her guidance, advice, criticism, encouragements and insight throughout the research.

I would like to express my thanks to my friends Suat Gümüşsoy and Cem Şafak Şahin. Their support and assistance was invaluable.

I would like to thank my dear friends Erol Aran and Yusuf Bediz. It is a great feeling to know that somebody cares about you and will be by your side in every situation.

I would like to give my special thanks to Yıldızfer Kemaloğlu for her great support.

Finally, I would like to thank my family for their understanding, support and patience; especially to my father and mother.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
MRI	Magnetic Resonance Imaging
NMR	Nuclear Magnetic Resonance
PC	Personal Computer
RF	Radio Frequency
TR	Time of Repetition
TE	Time of Echo
RAM	Random Access Memory
GHz	Giga Hertz
MHz	Mega Hertz
API	Application Programming Interface

CHAPTER 1

INTRODUCTION

1.1 General

Magnetic Resonance Imaging (MRI) has been a widely used imaging system after its invention two decades ago, due to its high capability and safety. It has a number of advantages compared to other non-invasive imaging modalities. These can be counted as high spatial resolution, high contrast on soft tissues, flexibility to image more than one parameter of biological tissues, and ability to generate two-dimensional images at any orientation without changing the position of the patient. MRI does not use ionizing radiation which can be harmful to the patients. It operates in radio-frequency (RF) range.

The growth in the usage of MRI scanners clinically, increased the need for educational and research tools in this area. In MRI, like other non-invasive imaging systems, physical and chemical properties of the object to be imaged are measured by applying external inputs. MRI is based on the nuclear magnetic resonance phenomenon (NMR). Basically, the protons in the body are excited by external magnetic fields according to the NMR phenomenon and these protons emit magnetic signals which are processed and transformed into an image. The whole process, starting from a motion of a proton under external magnetic fields to pulse sequences that generate the image, is quite complex and has a lot of parameters that have effects on the information content and quality of the image.

A software simulation of an MRI system can be very helpful to fully understand the underlying mechanisms that generate the images in MRI. MRI simulations generate images of perfectly known objects using the parameters defined by the user. The simulation can show the effect of these parameters on the image and can allow the artifacts created by the data collection method to be observed. Thus MRI simulations can be used for educational purposes in clinical environments. The simulations can also help students to better understand MRI by visualization of different blocks. Pulse sequences in MRI are still a research area. MRI simulations can also be helpful in observing the effectiveness of new pulse sequences before trying it on a real hardware. There is no universal MRI image database which can be used by investigators to test post-processing applications. An MRI simulation can also provide sample data for post-processing applications which are developed to reduce the artifacts and improve image quality.

1.2 Scope of the thesis

This thesis deals with the problem of developing a realistic and easily accessible MRI simulator. Besides the goal of being a research tool for engineers, it also aims to be an education tool for students and clinicians. The simulator in this study does not aim to simulate all the actors in an MRI system including hardware components, but it aims to simulate NMR phenomenon in a realistic way and generate many of the artifacts encountered in practice.

1.3 Outline of the dissertation

In chapter 2, basic principles of MRI are described starting from the NMR phenomenon to some widely used pulse sequences. This chapter provides the theoretical background for the simulations.

The implementation methods and the flowchart of the simulation are presented in chapter 3. This chapter also includes a literature survey giving a comparison of this study with previous works about this subject.

The user interfaces and the software architecture of the simulation software are discussed in chapter 4. This chapter provides a detailed user manual for the applications developed in this study.

Chapter 5 presents the results of the simulation software for different input parameters.

Finally, Chapter 6 reports the summary of the study and provides concluding remarks. Some future work is also suggested in this chapter.

CHAPTER 2

MRI BASIC PRINCIPLES

This chapter provides a brief theoretical background which is needed to understand the simulation methods. It is organized in two main sections. The first chapter explains how the NMR phenomenon works and gives definitions of MRI terms. The second part explains how the NMR phenomenon is used to generate images.

2.1 Nuclear Magnetic Resonance Phenomenon

MRI is based on the NMR phenomenon which was found in 1946 by Felix Bloch and Edward Purcell, both of whom were awarded the Nobel Prize in 1952. After its discovery, NMR is used by scientists to analyze chemical and physical properties of molecules. After 1970s, MRI has been discovered by using the NMR phenomenon with phase and frequency encoding to generate images of the body.

2.1.1 Nuclear Magnetic Moments

All materials consist of nuclei. Nuclei with an odd atomic number have a net electrical charge due to the unpaired nucleon. Nuclei also rotate around its own axis possessing an angular momentum $\vec{J}$. This spinning charged object is called a

spin and creates a magnetic moment $\vec{\mu}$ which can be related to the angular momentum $\vec{J}$ with the following relationship.

$$\vec{\mu} = \gamma \vec{J} \tag{2.1}$$

γ is a nucleus dependent physical constant known as *gyromagnetic ratio*. A related constant is $\not\gamma = \gamma / 2\pi$ and its value is 42.58 MHz/Tesla for a proton (^1H). All clinical use of MRI systems is based on ^1H which is one of the main atoms forming our body. An ensemble of spins of a specific atom constitutes a "spin system".

The magnitude of the magnetic moment μ ($A_{mpere} m^2$) is constant under any condition but its direction is completely random due to thermal motion unless an external magnetic field is applied. The following equation explains the physical motion of a spin under an external magnetic field.

$$\frac{d\vec{\mu}}{dt} = \vec{\mu} \times \vec{B} \tag{2.2}$$

If $\vec{B}$ is chosen a static magnetic field defined by $\vec{B}_0 = B_0 \vec{a}_z$, the spins start to rotate around $\vec{B}_0$ with a frequency which is proportional to the magnitude of $\vec{B}_0$.

$$w_o = \gamma B_0 \tag{2.3}$$

This precessional frequency w_0 is called the Larmor frequency or the resonant frequency of the spin system. In practice, all the spins of a specific atom may not have the same Larmor frequency. The group of spins that share the same resonance frequency is called *isochromat*. The reasons for having multiple isochromats for a specific spin are the inhomogeneity of $\vec{B}_0$ field, the static magnetic field distortion due to gyromagnetic ratio γ difference between tissues, and the chemical shift effect.

2.1.2 Bulk Magnetization

The net magnetization of an object being imaged is the vector sum of all magnetic moments in that object. If $\vec{M}$ represents the net magnetization and μ_n ($A_{mpere}m^2/m^3$) represents the magnetic moment of the n_{th} spin, then

$$\vec{M} = \sum_{n=1}^{Ns} \mu_n \tag{2.4}$$

where N_s is the total number of spin in the object.

In the absence of an external magnetic field, the net magnetization $\vec{M}$ over a volume is zero due to random orientation of individual magnetic moments. When an external magnetic field $\vec{B}_0$ is applied, the magnetic moments align themselves into discrete energy levels. Since ^1H is a ½ spin, there will be two discrete positions, one in the direction of $\vec{B}_0$, and the other in the opposite direction of $\vec{B}_0$. These two directions correspond to two different energy states. The spin population in these two states is slightly different because of the tendency of the spins to be in the lower energy state. This uneven distribution creates a net observable magnetic field along the direction of the applied magnetic field which is given by:

$$M_z^0 = \left|\vec{M}\right| = \frac{\gamma^2 \hbar^2 B_0 N_s}{4KT_s} \tag{2.5}$$

where $\hbar$ is the Planck's constant, K is the Boltzman constant and T_s is the temperature in Kelvin. As indicated by the Equation (2.5), the net magnetization of the object increases with the magnitude of applied static magnetic field $\vec{B}_0$. Although there is a net magnetization along the direction of $\vec{B}_0$, the magnetization on the transverse plane is zero since the spins in the two states are out of phase.

2.1.3 Bloch Equation

NMR is a quantum phenomenon if considered in atomic level. But classical physics can be applied to describe the collective behavior of spins in an object. Bloch equation describes the time dependent behavior of $\vec{M}$ under any applied magnetic field $\vec{B}(t)$.

$$\frac{d\vec{M}}{dt} = \gamma \cdot (\vec{M} \times \vec{B}) - \begin{bmatrix} M_x/T_2 \\ M_y/T_2 \\ (M_z - M_0)/T_1 \end{bmatrix} \tag{2.6}$$

Here, $\vec{M} = (M_x, M_y, M_z)$ and $\vec{B} = (B_x, B_y, B_z)$ are vectors. M_0 is the magnetization at thermal equilibrium in the presence of $\vec{B}_0$ only. T_1 and T_2 are time constants determining the relaxation process of $\vec{M}$ after excitation.

2.1.4 Rf Excitation and Resonance

A spin system generates an observable net magnetization along the direction of an external static magnetic field. However, in order to receive signals from an object, this magnetization vector $\vec{M}$ should be tipped to the transverse plane. This is achieved by applying an RF magnetic field perpendicular to $\vec{B}_0$ and exciting the spins. The RF magnetic field can be defined as follows:

$$\vec{B}_1(t) = B_1^e(t)\cos(w_{rf}t + \theta)\vec{a}_x \tag{2.7}$$

where w_{rf} represents excitation carrier frequency, and θ represents the initial phase angle. $B_1^e(t)$ is the pulse envelope function which determines the shape and the duration of the RF pulse. Its main purpose is to select the region to be excited.

The behavior of the magnetization vector $\vec{M}$ under excitation pulse $\vec{B}_1(t)$ can be studied by Bloch Equation (2.6). If typical T_1 and T_2 (on the order of msec) values and typical excitation durations (on the order of μsec) are considered, the effect of T_1 and T_2 can be ignored during excitation. Also the motion of $\vec{M}$ can be

observed more simply in a rotating frame. So if we consider a reference frame rotating around the z axis with angular velocity w, Equation (2.6) takes the following form for $\vec{M}$.

$$\frac{d\vec{M}_{rot}}{dt} = \gamma \cdot \vec{M}_{rot} \times \vec{B}_{eff} \tag{2.8}$$

where

$$\vec{B}_{eff} = \vec{B}_{rot} + \frac{\vec{w}}{\gamma} \tag{2.9}$$

Here, $\vec{M}_{rot}$ and $\vec{B}_{rot}$ are the magnetization vector and the net external magnetic field vector at rotating frame respectively. If the rotating frame is rotating at the Larmor frequency w_0, then $\vec{w} = \gamma B_0 \vec{a}_z$. If only $\vec{B}_0$ is applied, then $\vec{B}_{rot}$ becomes $\vec{B}_{rot} = B_0 \vec{a}_z$ and the effective magnetic field $\vec{B}_{eff}$ becomes $\vec{B}_{eff} = B_0 \vec{a}_z - \frac{\gamma B_0 \vec{a}_z}{\gamma} = 0$. If $\vec{B}_{eff}$ is zero, then $\vec{M}_{rot}$ seems stationary in the rotating frame according to Equation (2.8).

If $\vec{B}_1(t)$ is applied with a carrier frequency equal to w_0, $\vec{B}_{eff}$ becomes $\vec{B}_{eff} = B_0 \vec{a}_z + B_1(t) - \frac{\gamma B_0 \vec{a}_z}{\gamma} = \vec{B}_1^e(t)$. This input causes a precession motion about the $\vec{B}_1$ field and the magnetization vector tilts away from the $\vec{z}$ axis. The direction of $\vec{B}_1(t)$ should be perpendicular to static magnetic field $\vec{B}_0$. If the direction of $\vec{B}_1(t)$ is along $\vec{x}$, the motion of $\vec{M}$ in the rotating frame is governed by:

$$\begin{aligned} M_x(t) &= 0 \\ M_y(t) &= M_z^0 \sin\left(\int_0^t \gamma \cdot B_1^e(t)dt\right) \\ M_y(t) &= M_z^0 \cos\left(\int_0^t \gamma \cdot B_1^e(t)dt\right) \end{aligned} \tag{2.10}$$

If $\vec{B}_1^e(t)$ is a rectangular envelope function, the rotational frequency of this precession will be $\vec{w}_1 = \gamma\vec{B}_1$. The amount of the tip angle α is determined by the area under the envelope function:

$$\alpha = \int_0^{T_p} \gamma B_1^e(t)dt \tag{2.11}$$

The carrier frequency w_{rf} being equal to Larmor frequency w_0 is called as the resonance condition. From a quantum perspective, the resonance excitation causes the spins at the lower energy state to switch to the higher energy state. This is achieved by exciting the spins with an amount of energy exactly equal to cause a transition from one state to another.

If the carrier frequency w_{rf} is different from the resonance frequency w_0, then the effective field for the spin in w_{rf} rotating frame becomes:

$$\vec{B}_{eff} = (B_0 - \frac{w_{rf}}{\gamma})\vec{a}_z + \vec{B}_1(t) = \Delta w_0 \vec{a}_z + \vec{B}_1(t) \tag{2.12}$$

In this case, the effective field has a vertical and horizontal component. For a rectangular shaped $\vec{B}_1(t)$, the effective field $\vec{B}_{eff}$ points along the resultant vector of Δw_0 and the magnitude of $\vec{B}_1(t)$. The magnetization vector again rotates around this effective field. If $\Delta w_o >> B_1$, then the effective field becomes close to $\vec{B}_0$ and the magnetization vector does not tilt away from the $\vec{z}$ axis significantly. This is called off-resonance excitation.

2.1.5 Relaxation

The RF magnetic field $\vec{B}_1(t)$ is the external force that perturbs the magnetized spin system from its thermal equilibrium value by flipping the magnetization vector into the transverse plane. When this external energy is turned off, the magnetization vector returns to its initial position by making a precession motion about $\vec{B}_0$. This process is called relaxation and it is characterized by the recovery

of longitudinal magnetization M_z and the decay of transverse magnetization M_{xy}.

When the RF field is turned off, the effective field $\vec{B}_{eff}$ seen by the magnetization vector in the Larmor rotating frame becomes $\vec{B}_{eff} = B_0\vec{a}_z - \frac{\gamma B_0 \vec{a}_z}{\gamma} = 0$. Then the Bloch equation reduces to:

$$\begin{aligned} dM_z &= -\frac{M_z - M_z^0}{T_1} \\ dM_{xy} &= -\frac{M_{xy}}{T_2} \end{aligned} \tag{2.13}$$

where T_1 is called the spin-lattice relaxation time and T_2 is spin-spin relaxation time. The following are the solution of the above equation.

$$M_{xy}(t) = M_{xy}(0_+)e^{-t/T_2}e^{-iw_0t} \tag{2.14}$$

$$M_z(t) = M_z^0(1-e^{-t/T_1}) + M_z(0_+)e^{-t/T_1} \tag{2.15}$$

where $M_{xy}(0_+)$ and $M_z(0_+)$ are magnetizations on the transverse plane and along the $\vec{z}$ axis just after the RF pulse. Equation (2.14) indicates that the transverse magnetization M_{xy} makes a precession motion on the transverse plane around $\vec{z}$ axis and its magnitude decays exponentially at a rate determined by T_2. As stated in Equation (2.15), the longitudinal magnetization M_z recovers its equilibrium value M_z^0 exponentially at a rate determined by T_1. The decay of M_{xy} and the recovery of M_z during relaxation are shown on the Figure (2.1) below.

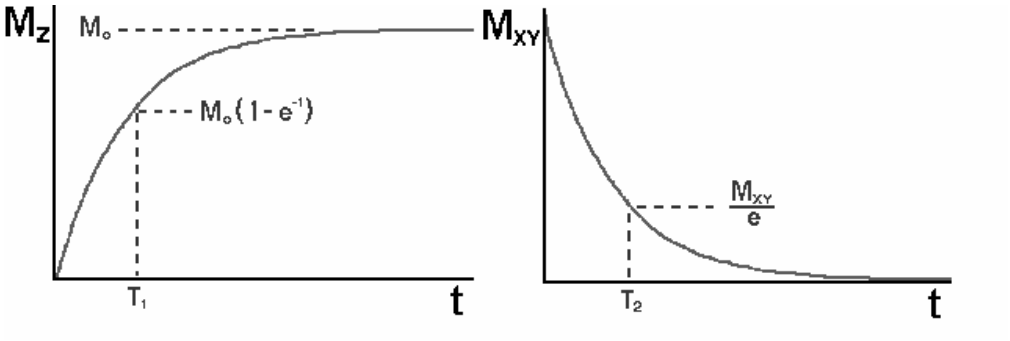

Figure 2.1 – The change of M_z and M_{xy} during relaxation [2]: the plot on the left shows the recovery of longitudinal component. After a relaxation duration of T_1, M_z reaches approximately %63 of its steady value. The plot on the right shows the decay of transverse magnetization, M_{xy} loses approximately %63 of its initial value after an interval of T_2.

T_1 and T_2 are tissue dependent parameters and they enable identification of different tissues on MR images. T_1 is about 300 to 2000 ms and is always longer than T_2 which is about 30 to 150 ms [1]. This implies that M_{xy} goes to zero before M_z grows to M_z^0 along the $\vec{z}$ axis. If all the spins in an object had the same rotational frequency on the transverse plane, the net magnetization would have a phase coherence and the decay of M_{xy} would be at the same rate with the recovery of M_z (T_1 = T_2). But this is not the case due to spin to spin interactions. Each spin experiences a slightly different static magnetic field and this causes them to rotate at slightly different rotational frequencies which results in a phase dispersion and the loss of transverse magnetization M_{xy} before T_1.

Besides the spin to spin interactions, if the applied static magnetic field $\vec{B}_0$ is not homogenous on the object, the spins experience more different local magnetic fields and have more different rotational frequencies. This results in decay of

M_{xy} earlier than T_2. This decay rate is called T_2^*. In practice, the net magnetization on the transverse plane M_{xy} never decays with T_2 but with T_2^*. The inhomogeneity of $\vec{B}_0$ is not the only reason behind T_2^*. The chemical shift effect and the magnetic susceptibility variations on the boundaries of the tissues are the other reasons of T_2^* decay [1, 3].

2.2 Image Generation

If an object is placed in an external magnetic field $\vec{B}_0$ and excited with an RF magnetic field $\vec{B}_1(t)$, the magnetization vector $\vec{M}$ flips into transverse plane. After the alternating magnetic field $\vec{B}_1(t)$ is turned off, the magnetization vector $\vec{M}$ returns to its equilibrium position by making a precessional motion around $\vec{B}_0$. This is briefly the NMR phenomenon. Now the question is how to generate an image using NMR.

This section explains the process that produces the MR image starting from the signal detection to pulse sequences.

2.2.1 Signal Detection

In MRI, the net magnetization vector $\vec{M}$ starts rotating at an RF frequency on the transverse plane after the RF magnetic field $\vec{B}_1(t)$ is turned off. This magnetization can be detected by placing a coil along the object. By Faraday's law of electromagnetic induction, this rotating magnetization will induce a voltage at the receiver coil that is equal to the rate at which the flux through the coil is changing. The voltage in the coil can be expressed as:

$$V(t) = -\frac{\partial \Phi(t)}{\partial t} = -\frac{\partial}{\partial t}\int_{object} \vec{B}_r(\vec{r}) \cdot \vec{M}(\vec{r},t) dr \qquad (2.16)$$

where $\Phi(t)$ is the flux through the coil and $\vec{B}_r(\vec{r})$ and $\vec{M}_r(\vec{r})$ are the magnetic flux density and magnetization vector at the spatial position $\vec{r}$ respectively. Assuming that the receiver has a homogenous reception sensitivity over the region of interest and considering that $M_z(t)$ is a slowly varying function, the received signal equation turns into the integration of transverse magnetization M_{xy} over the entire volume.

$$S(t) = \iiint M_{xy}(x,y,z,t)dxdydz = \int_{object} M_{xy}(r,t)dr \tag{2.17}$$

where $S(t)$ is the received signal. Considering the relaxation equation of transverse magnetization (2.14), this equation can be written as:

$$S(t) = \int_{object} M_{xy}(r,0_+)e^{-t/T_2(r)}e^{-iw(r)t}dr \tag{2.18}$$

The signal received at the coil is a high frequency signal which can cause problems for electronic circuitries [1]. In practice, this signal is moved to baseband by signal demodulation method. This method multiplies the input signal with a reference signal and removes the high frequency component by low-pass filtering. In MRI, this reference signal is a sinusoid at the Larmor frequency. This demodulation operation converts the signal equation into:

$$S(t) = \int_{object} M_{xy}(r,0_+)e^{-t/T_2(r)}e^{-i\Delta w(r)t}dr \tag{2.19}$$

where $\Delta w(r) = w(r) - w_0$.

2.2.2 Signal Localization

Equation (2.17) tells us that the received signal is the sum of all local magnetization vectors in the object. If all these magnetization vectors had the same rotating frequency $w(r)$, the received signal $S(t)$ would have only frequency component and would not give any spatial information. To be able to obtain spatial information, gradient fields are used which are defined as:

$$\vec{B}_G(x,y,z) = (G_x x + G_y y + G_z z)\vec{a}_z \tag{2.20}$$

Gradient fields are always applied along the direction of static magnetic field $\vec{B}_0$ and change the net static magnetic field along x, y, z directions. This is illustrated in Figure 2.2 In MRI, gradient fields are used for slice selection and spatial encoding. The net static magnetic field at any point along with the application of gradient fields is calculated with the below equation.

$$\vec{B}(r) = (B_0 + B_G \cdot r)\vec{a}_z \tag{2.21}$$

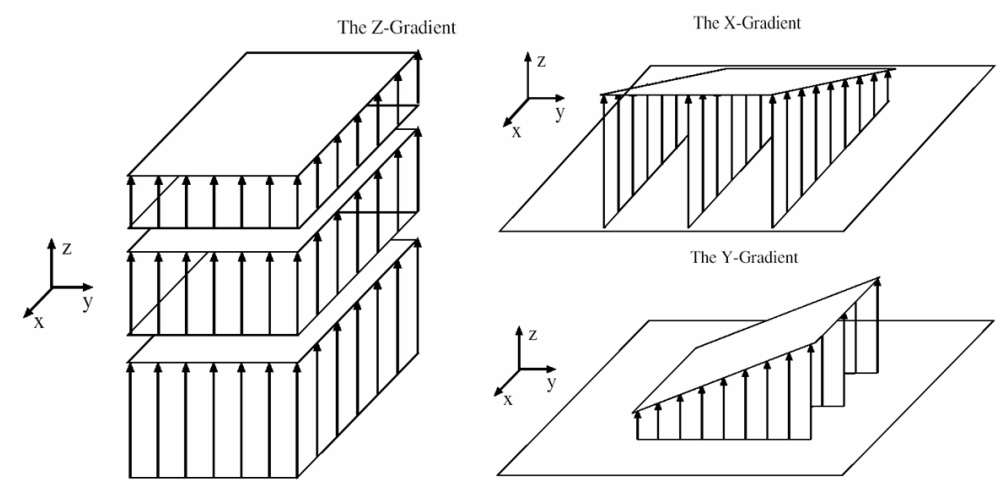

Figure 2.2 – The affect of gradient fields on the net static magnetic field [2].

2.2.2.1 Slice Selection

In MRI, it is usually desired to obtain the image of a slice of the object. This is achieved by exciting the spins only in that slice as shown in Figure 2.3. To select a slice, two things are necessary: a gradient field along $\vec{z}$ direction and a shaped RF pulse. If the slice to be selected is along $\vec{z}$, then $\vec{B}_G = (G_z z)\vec{z}$ can be applied. Considering the RF pulse Equation (2.7), the RF frequency w_{rf} and the envelope function $B_1^e(t)$ determine the slice center $z = z_c$ and the slice width Δz. The Larmor frequency distribution in a slice of width Δz and centered at $z = z_c$ is given by :

$$p(w) = \Pi(\frac{w - w_c}{\Delta w}) \tag{2.22}$$

where $w_c = \gamma(B_0 + G_z z_c)$ and $\Delta w = \gamma G_z \Delta z$. This rectangular distribution in frequency domain can be met by a sinc shaped envelope function in time domain. The resultant RF magnetic field becomes:

$$\vec{B}_1(t) = A\sin c(\pi \Delta f t) e^{-iw_c t} \vec{a}_x \tag{2.23}$$

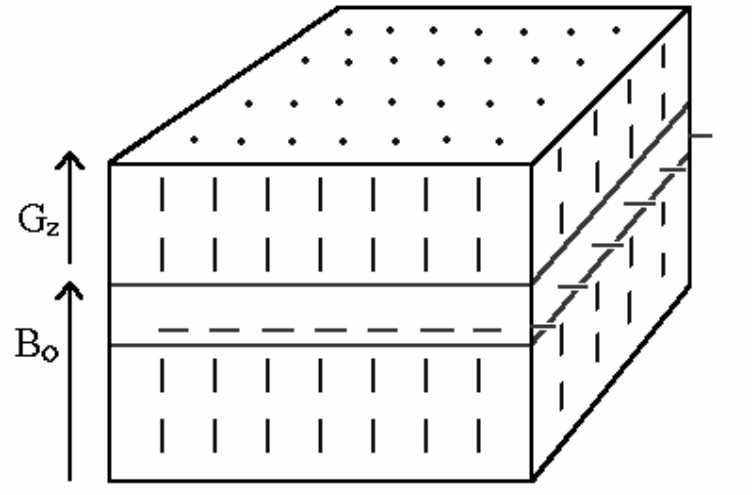

Figure 2.3 – Slice selection selectively excites spins in a region [2].

2.2.2.2 Spatial Encoding and K-space

Spatial information can be obtained from the MR signal by encoding the magnetization vectors based on their position. This is achieved by frequency encoding and phase encoding.

Frequency encoding, as the name applies, is changing the frequency of precession linearly with position. If a constant gradient G_x is applied for frequency encoding, the Larmor frequency at (x, y) can be expressed as:

$$w(x, y) = w_0 + \gamma G_x x \tag{2.24}$$

When frequency encoding is applied, contributions of magnetization vectors at different locations along $\vec{x}$ will have different precessional frequencies.

Phase encoding is the same as frequency encoding except that it is applied for a short interval T_{pe} and then it is turned off. As a result of this, signals from different locations will have different phase angles. If a constant gradient G_y is applied for phase encoding, the Larmor frequency will be $w(x,y) = w_0 + \gamma G_y y$ during the interval $0 \le t < T_{pe}$. The total phase accumulated at the end of this interval will be:

$$\theta(T_{pe}) = \gamma G_y y T_{pe} \tag{2.25}$$

If both frequency encoding and phase encoding are applied during an MRI signal acquisition, and if T_2 relaxation is ignored, the baseband signal equation becomes:

$$S(t) = \int_{object} M_{xy}(r,0_+) e^{-t/T_2(r)} e^{-iw(r)t+\theta(r)} dr = \iint M_{xy}(r,0_+) e^{i(\gamma G_x xt + \gamma G_y y T_{pe})} dxdy \tag{2.26}$$

If we denote $k_x = -\lambda G_x t$ and $k_y = -\lambda G_y T_{pe}$, then the signal equation turns into:

$$S(t) = S(k_x(t), k_y) = \iint M_{xy}(x,y,0_+) e^{i(k_x x + k_y y)} dxdy \tag{2.27}$$

As seen in Equation (2.27), the signal equation is the Fourier transform of magnetization at time t. After phase encoding interval, each voxel along the $\vec{y}$ direction has a distinct phase although they have the same precession frequency. The frequency encoding causes every location along $\vec{x}$ direction to have a distinct precession frequency during readout. This distinct pair of a phase and a frequency of the magnetization precession converts the received signal into the Fourier transform of the magnetization.

2.2.2.3 Pulse Sequences

The signal measured in Equation (2.27) contains samples from one row of spatial frequency space (k_x, k_y), also called as the K-space in MRI literature. The

relationship between frequency domain and time domain is same for K-space and spatial domain. If all rows of K-space are filled by taking measurements of $S(t)$ with different values of G_y, the Fourier transform of the magnetization function $M(x,y)$ is obtained. Then one can apply inverse Fourier transform operation to obtain the magnetization function $M(x,y)$:

$$M(x,y) = F^{-1}\left(S(k_x,k_y)\right) \tag{2.28}$$

To fill another row in K-space, the system should be excited again and a readout must be performed witch switches to another row in K-space by use of gradients. The sequence of these excitation and readout events are called as a pulse sequence. An example of a pulse sequence is shown in Figure 2.4. The time between two consecutive excitations is called time of repetition (TR).

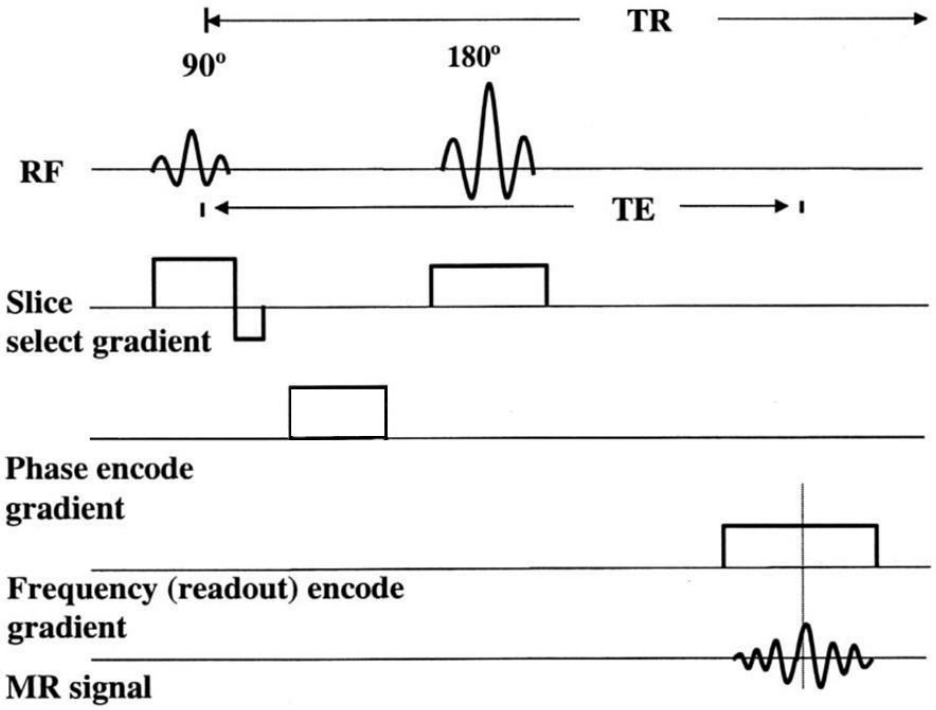

Figure 2.4 – Spin Echo Pulse Sequence [2]: the echo time TE is the duration where the readout signal reaches its peak value, the pulse sequence above is applied number of phase encoding steps times where each step performed in TR duration.

It is more convenient to express $M(x, y)$ in (2.28) as the imaging function rather than the magnetization function since it is a function of proton density p and relaxation constants T_1 and T_2. By changing the parameters of the pulse sequence, it is possible to weight the received signal $M(x, y)$ with one of these tissue properties.

CHAPTER 3

SIMULATION METHODS

Simulations should be close to reality in order to be useful. The simulation system proposed in his thesis, aims to create a flexible and expandable simulation framework which takes into account most of the physical processes in MRI.

Many MRI simulations have been developed so far by different research groups [3-7]. These simulators have different approaches in modeling the reality, methods of implementation and software design. The previous simulators can mainly be divided into two groups.

Some simulators generate new images from known images by using the image intensity functions of pulse sequences. The image intensity functions depend on spin parameters; proton density, T_1 and T_2 and pulse sequence parameters; flip angle, time of echo T_E and time of repetition T_R [7]. These simulations are helpful in observing the image contrast variation with pulse sequence parameters. However, this approach does not simulate the whole process of MRI and thus is not able to simulate all the artifacts like chemical-shift, intra-voxel dephasing, imperfection of slice selection, aliasing, non-linear gradients, B_0 inhomogeneity and susceptibility artifacts. Some of these simulators generate k-space data by taking the Fourier transform of the input parameter image and modify the k-

space data to generate the artifacts encountered in MRI. Then by taking the inverse Fourier transform, output images are obtained.

Another group of simulators is based on the solution of Bloch equation for virtual objects which are composed of small volume elements called voxels, each representing a magnetization for a specific spin [3, 4, 6]. Depending on the pulse sequence parameters selected, the magnetization for each voxel is calculated according to the Bloch equation and the summation of these magnetizations yield the MR signal. This approach is the closest model to reality since it is a discrete representation of the real process. In this thesis, this model is followed. This model is only limited by the Bloch equation which ignores some physical events like diffusion. This approach suffers from high computation times due to the need of large number of magnetization vectors to correctly represent the virtual object. Some simulators make use of the computation power of parallel processing to decrease the high computation times [4,6]. The spin model and the Bloch equation are appropriate for such a distributed node implementation.

The Bloch equation based simulators take the virtual object definition and pulse sequence parameters from the user and generate the MR signal by a Bloch equation solver. These simulators differ in the solution of Bloch equation and generation of echoes. Some simulators calculate the magnetization values using rotational and scaling matrix operations corresponding to excitation, relaxation and gradient field affects. Analytical solution of Bloch equation is used for calculation of flip angle values which is an acceptable approach if the step size is chosen small enough [3,6]. In [3], a hybrid approach is used where numerical methods are used when the analytic solution does not exist for the Bloch equations and analytic solution is preferred other times.

A simulation should be able to generate the artifacts created by the actors in a physical process to be close to reality. In MRI, these artifacts can be counted as: chemical-shit effect, B_0 inhomogeneity, RF magnetic field inhomogeneity, non-

linear gradients, aliasing, imperfection of slice selection, intra-voxel dephasing, and Gibbs phenomenon. To simulate the artifacts, the sources that create these artifacts must be modeled in a realistic way.

To generate echoes, Benoit [4] uses a tricky method which reduces the number of magnetization vectors per voxel but is based on the assumption of a Lorentzian distribution for static magnetic field inhomogeneity. This approach is very interesting since it allows generation of echoes using only one magnetization vector per voxel. Yoder [3] includes magnetic susceptibility variations besides static magnetic field inhomogenity to simulate local magnetic field variations. This is a more realistic approach to simulate intra-voxel dephasing but requires processing of the virtual objects to calculate susceptibility changes before the simulation. In this study, intra-voxel dephasing is realized by changing local magnetic fields using B_0 inhomogeneity.

There is only one web based MRI simulator in the literature [8]. It is also a Java applet which is accessible through internet. This simulator is very helpful for teaching MRI contrast behavior but it is in the group of k-space based simulators which lacks generality. The other Bloch equation based simulators [3,4] perform realistic simulations but are either stand-alone PC application for specific platforms or require special hardware to run simulations. The simulation software developed in this study is the first Bloch equation based, web accessible MRI simulator in the literature.

3.1 Overview of Simulator

The simulation software is composed of four main building blocks shown in Figure 2.2. Among these blocks the 3D virtual object definition and Pulse sequence definition blocks are the ones that the user provides as input to the simulation. The Bloch equation simulation block is the kernel of the simulation

and generate the MRI signal by using the input blocks. The image construction block processes the received signal and generates the image function.

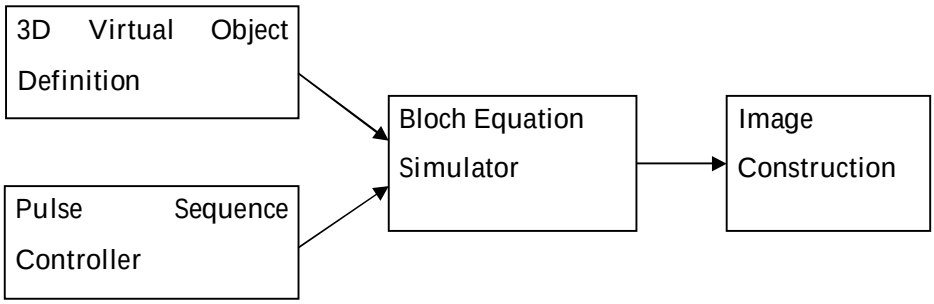

Figure 3.1 – Main building blocks of the simulator: the 3D virtual object definition and the pulse sequence definition are the inputs of the system, the Bloch equation simulator block is the kernel of the simulation.

3.2 3D Virtual Object Definition

The structure of the virtual object in MRI simulations is very important since it determines the features of the data set which the simulation will be tested on. The structure should be flexible and should not limit the capabilities of the simulation. In this study, the 3D virtual object is a discrete definition of a real 3D object. It is composed of $MxNxK$ voxels. A voxel is a pixel with some finite thickness. In MRI, the images are obtained from slices with a finite width, so it is better to compose the object from voxels instead of pixels. A 3D object is composed of many slices and slices are composed of voxels as in Figure 3.2.

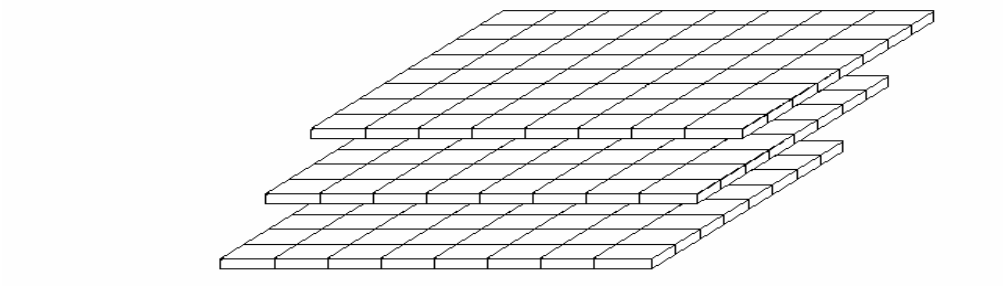

Figure 3.2 – 3D Virtual Object composed of three slices.

Voxels are defined cubical so one dimension is enough. A voxel has a 3D position information defined in Cartesian coordinates (x, y, z) and contains magnetization information for a spin. The voxel holds a proton density value p_i and the spin information which is defined with the following properties: gyromagnetic ratio γ, and the relaxation constants T_1 and T_2. The spin information changes from tissue to tissue. In a voxel, the magnetization is represented by one or a group of magnetization vectors. Each magnetization vector has 3D position information and holds a 3D vector for magnetization value. The positions of the magnetization vectors are determined by distributing them evenly in the voxel. A clearer picture of the structure of the virtual object is given with the following UML class diagram in Figure 3.3.

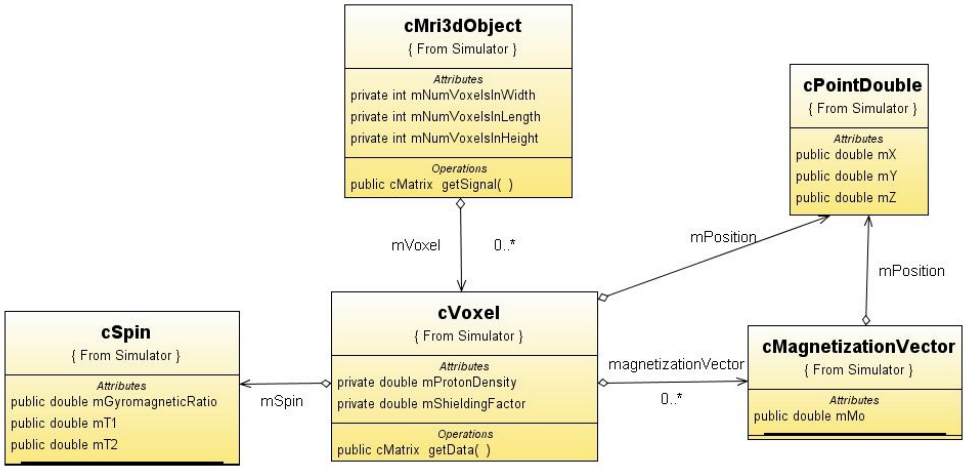

Figure 3.3 – Virtual MRI 3D object class diagram: as shown on the figure, the 3D virtual object class holds multiple voxel objects and the voxel class holds multiple magnetization vector objects, each voxel object has one spin and voxel and magnetization vector objects have position information.

If $\vec{m}_i$ represents i_{th} magnetization vector of the voxel, the net magnetization of a voxel at position (x, y, z), will be the summation of all magnetization vectors in that voxel divided divided by the number of magnetization vectors.

$$\vec{M}(x,y,z) = \frac{1}{N}\sum_{i=1}^{N}\vec{m}_i \tag{3.1}$$

where N is the total number of magnetization vectors in the voxel. The total magnetization of the virtual object will be the summation of net magnetization values from each voxel:

$$\vec{M}_{object} = \sum_{z=1}^{K}\sum_{y=1}^{N}\sum_{x=1}^{M}\vec{M}(x,y,z) \tag{3.2}$$

where K is the number of voxels along the $\vec{z}$ direction, N is the number of voxels along the $\vec{y}$ direction and M is the number of voxels along the $\vec{x}$ direction of the object. $MxNxK$ gives the total number of voxels in the object.

Representing the magnetization of the voxel by more than one magnetization vector creates a more realistic model since it allows intra-voxel dephasing by giving a different precession frequency to each magnetization vector. The dephasing affect created by gradient fields is also supported in this voxel model since magnetization vectors are distributed spatially in the voxel.

To be able to simulate the chemical-shift affect, each voxel holds a shielding factor δ. This factor is a value between 0 and 1. It generates an effective gyromagnetic ratio γ_{eff} by scaling the gyromagnetic ratio as in Equation (3.3). As a result it changes the net static magnetic field experienced by a spin. The shielding factor δ value is usually a few parts per million [1].

$$\gamma_{eff} = \delta\gamma \tag{3.3}$$

3.3 Pulse Sequence Controller

This block controls the timing and order of the events and sets the magnetic input signals for each event based on the pulse sequence type and parameters selected. It also calculates the data acquisition parameters according to the input object and pulse sequence parameters. It takes the pulse definition as input, splits the total sequence into MRI events which will be explained below, and processes each event in the sequence consecutively.

3.3.1 Pulse Sequence Definition

The pulse sequence parameters provided by the user are TE (time of echo), TR (time of repetition) and flip angle value. Three types of pulse sequences are implemented in the simulation. These are spin echo, gradient echo, and echo planar pulse sequences.

There are two types of events for a magnetized spin system from the point of applied magnetic fields. These are excitation and relaxation events. A combination of these events creates a pulse sequence. Excitation event is the one which the system is excited with an RF magnetic field. Relaxation event corresponds to the relaxation of the system with the RF magnetic field off. Gradient fields may be on or off in both events. A special case of relaxation event is the readout case where samples are collected from the system and stored in a buffer for processing. Excitation event is defined mostly by the RF magnetic field parameters including frequency, amplitude, duration, envelope function type and bandwidth. Gradient field values are also parameters for excitation event. Relaxation event is defined with the gradient field parameters and duration. If readout is going to be performed during relaxation, sampling time is also provided. The simulation software has 3 classes that handle the processing of these events shown on Figure 3.4. A combination of these event processing functions executed consecutively for a virtual object generate a pulse sequence.

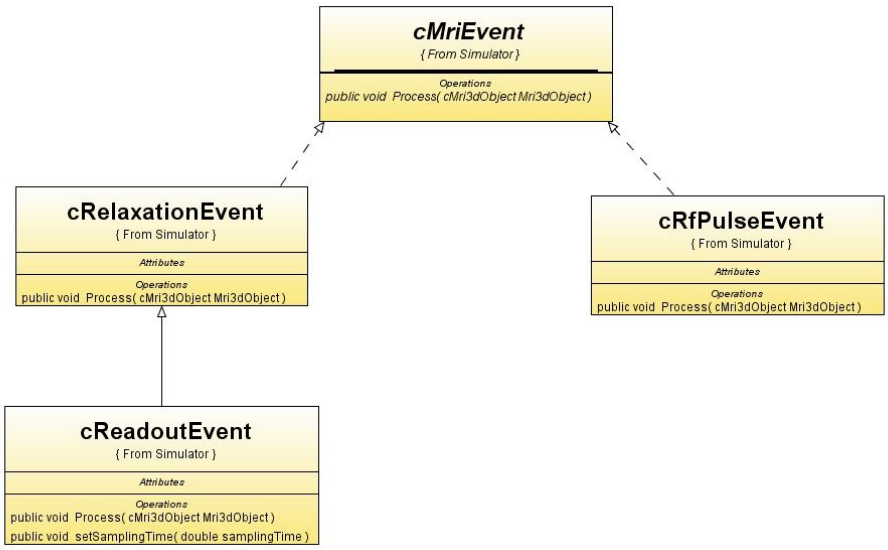

Figure 3.4 – MRI event processing classes: the base class cMriEvent class is an interface for relaxation and excitation event classes, the polymorphic structure for MRI event classes enables the event objects to be put in a list and executed consecutively, the cReadoutEvent class is derived from cRelaxationEvent class since it is a special form of relaxation event.

To perform a simulation, the pulse sequence controller creates events according to pulse sequence type selected and its parameters. Figure 3.5 shows how a pulse sequence is divided into events. Excitation events are only defined for time regions where RF magnetic field is enabled. The remaining time regions are divided into relaxation events, each corresponding to a different combination of gradient fields. If k-space is to be filled during relaxation, it becomes a readout event.

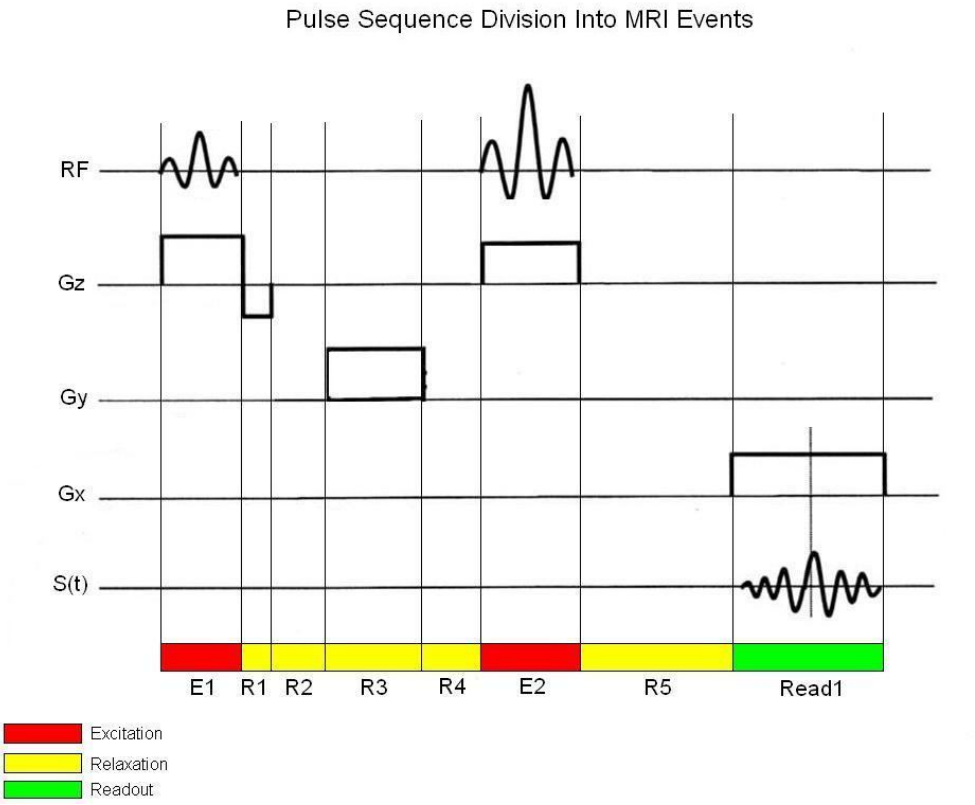

Figure 3.5 – Event generation from a pulse sequence: the spin echo pulse sequence above is represented with 2 excitation events, 5 relaxation events and 1 readout event, each relaxation event corresponds to a different combination of gradient fields.

Three types of magnetic inputs exist in MRI. These are static magnetic field $\vec{B}_0$, RF magnetic field $\vec{B}_1$ and the gradient field $\vec{B}_G$. The RF magnetic field is defined by its carrier frequency, amplitude, duration, and envelope function shape. The simulation program allows two types of envelope functions to be selected: rectangular and sinc. If the envelope function is sinc shaped, the bandwidth value Δf in Equation (2.23) is calculated. The gradients fields are defined by their amplitude $G = (G_x, G_y, G_z)$ and their duration.

3.3.2 Simulation Flow

Figure 3.6 explains how the simulation software treats each event in a pulse sequence.

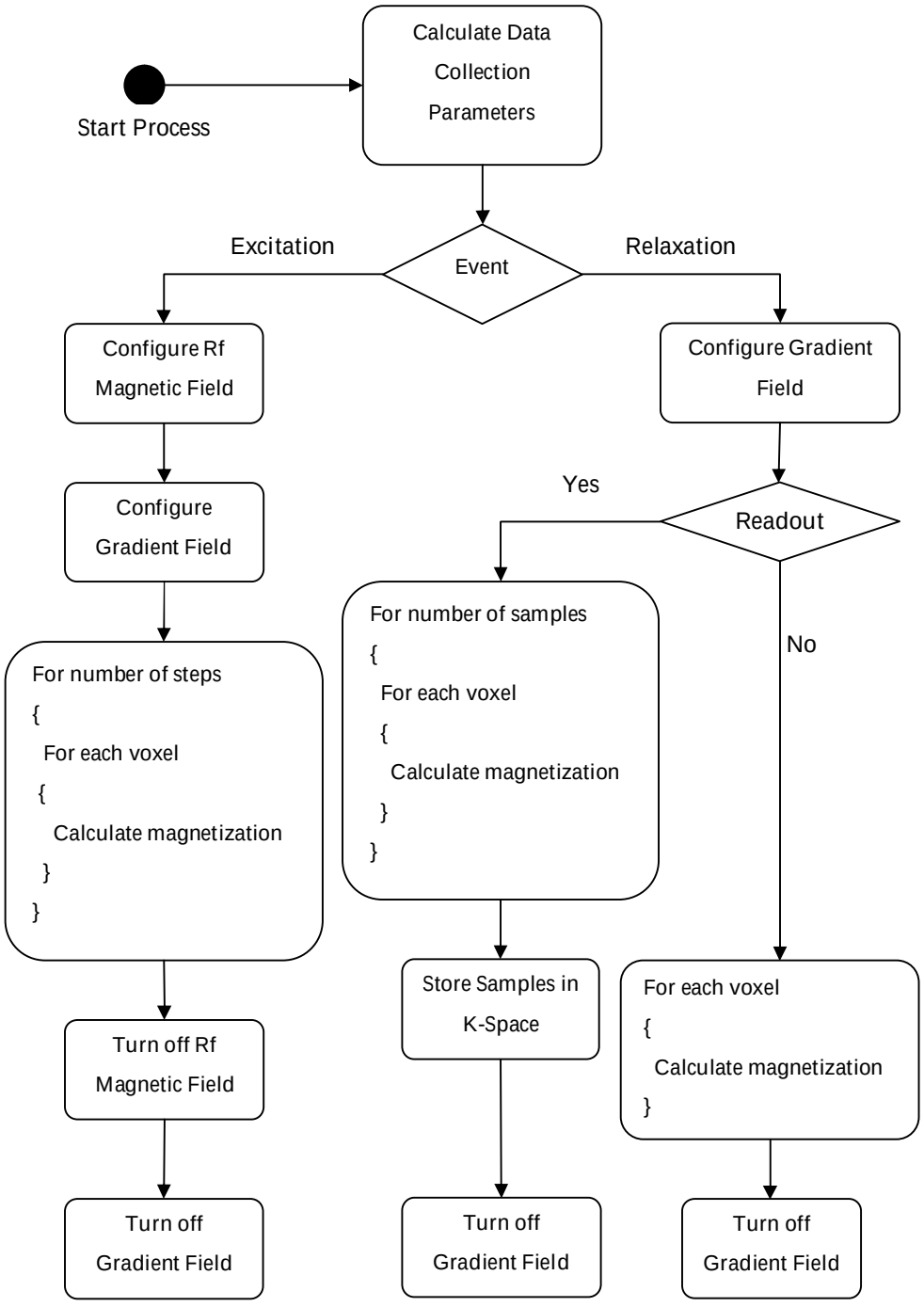

Figure 3.6 – Pulse Sequence Controller Flow.

The simulation software executes the events in a pulse sequence consecutively. Before a pulse sequence process is started, firstly data collection parameters are calculated. After this the first event is taken from the list. Each event configures the magnetic inputs before processing and turns them off after processing. The RF magnetic field $\vec{B}_1$ and the gradient field $\vec{B}_G$ are configured in an excitation event. Relaxation and readout events only configure gradient fields. After the magnetic inputs are configured, the new magnetization values for each voxel in the object are calculated by the Bloch equation simulator block.

After all the events are processed in a pulse sequence, the k-space buffer is processed to obtain the image function.

3.3.3 Data Collection Parameters

Data collection parameters are calculated to meet the Nyquist theorem for K-space sampling. Nyquist theorem states that a signal should be sampled at a rate at least twice the bandwidth of the signal in order to reconstruct the signals from its samples again. The signal K-space sampling intervals Δk_x and Δk_y are determined as follows:

$$\begin{aligned} \Delta k_x &= 1/W_x \\ \Delta k_y &= 1/W_y \end{aligned} \tag{3.4}$$

where W_x and W_y are the width and the length of the virtual object. The sampling time is found using the equation $k_x = -\lambda G_x t$. Frequency encoding gradient G_x is defined by the user. Then sampling time Δt becomes:

$$\Delta t = \Delta k_x / \lambda G_x \tag{3.5}$$

After this, the step size of phase encoding gradient value is determined by using the equation below:

$$\Delta G_y = \Delta k_y / \lambda T_{pe} \tag{3.6}$$

where T_{pe} is the phase encoding gradient duration. This parameter is chosen to be half of the data acquisition time. Data acquisition time is equal to the number of voxels along the frequency encoding direction times the sampling time Δt.

3.4 Bloch Equation Simulator

The Bloch equation simulation is the heart of an MRI simulation program since it calculates the net magnetization from the virtual object and generates the K-space data. This block performs the task of "calculation magnetization values" in the pulse sequence flow in Figure 3.6. The input-output scheme of Bloch equation simulator is shown on Figure 3.7.

The inputs of the Bloch equation simulator are the magnetization vector $\vec{M}(t,\vec{r})$ at position $\vec{r}$, the spin parameters of this magnetization vector (γ, T_1, T_2), the net external magnetic field at position $\vec{r}$ and time t and the duration of the event Δt. The net external magnetic field is obtained from the summation of static magnetic field, RF magnetic field and gradient fields. Static magnetic field $\vec{B}_0$ only depends on spatial position, RF magnetic field only depends on time and the gradient fields depend both on time and spatial position.

$$\vec{B}_{net}(t,\vec{r}) = \vec{B}_0(\vec{r}) + \vec{B}_1(t) + \vec{B}_G(t,\vec{r})\cdot\vec{r} \tag{3.7}$$

The output of the simulator is the magnetization vector $\vec{M}(t+\Delta t,\vec{r})$ which is calculated by using the input vector $\vec{M}(t,\vec{r})$, the net external magnetic field and spin parameters. The Bloch equation simulator is composed of two processing blocks: one used for excitation and the other used for relaxation.

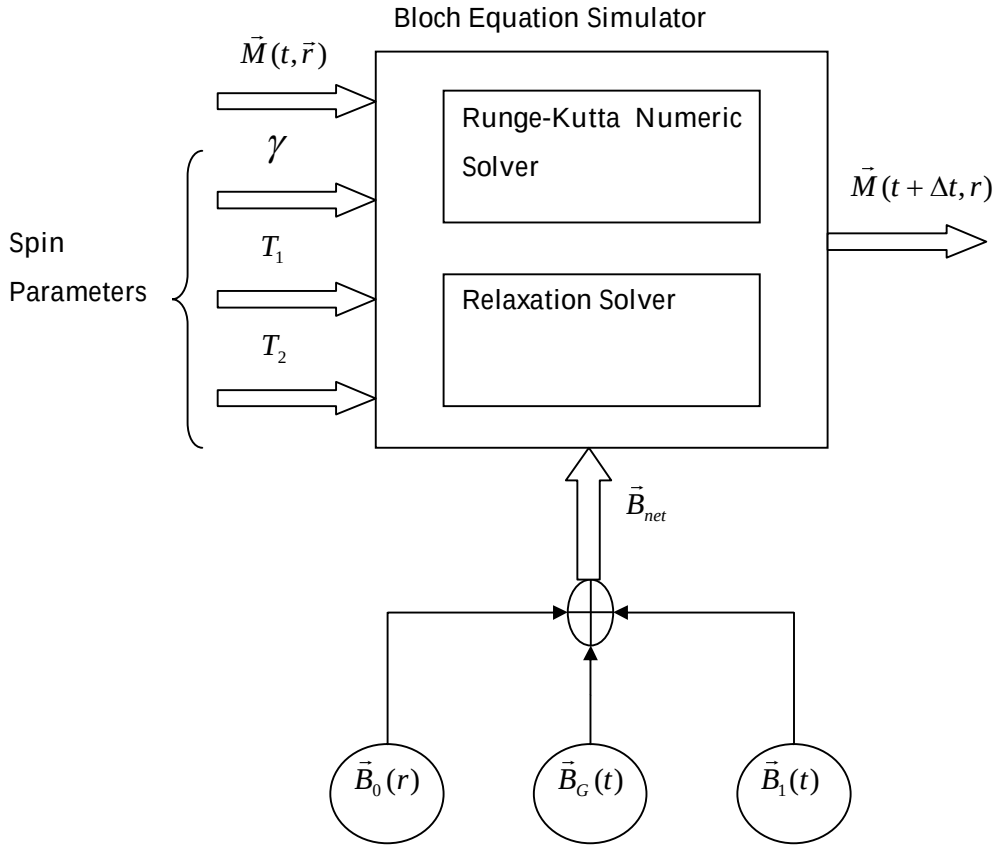

Figure 3.7 – Bloch Equation Simulator Input/Output Scheme: the inputs of the simulator are $\vec{M}(t,\vec{r})$, spin parameters, the net external magnetic field at position $\vec{r}$ and time t, and the duration of the event Δt. The output of the simulator is the magnetization vector $\vec{M}(t+\Delta t,\vec{r})$.

During relaxation, the solution of the Bloch equation is trivial and it is given by Equation (2.15). When a constant gradient is applied during relaxation, the solution does not change from (2.15) but w_o is replaced by:

$$w(\vec{r}) = \gamma \cdot B_{net}(\vec{r}) = \gamma \cdot (B_0(\vec{r}) + \vec{B}_G \cdot \vec{r}) \tag{3.8}$$

So the magnetization vector at time $t+\Delta t$ can be found by the following equations:

$$M_{xy}(t+\Delta t,\vec{r}) = M_{xy}(t,\vec{r})e^{-\Delta t/T_2}e^{-iw(\vec{r})\Delta t} \quad (3.9)$$

$$M_z(t+\Delta t,\vec{r}) = M_z^0(1-e^{-\Delta t/T_1}) + M_z(t,\vec{r})e^{-\Delta t/T_1} \quad (3.10)$$

This equation is used in the relaxation solver block of Bloch equation simulator. The simulator calculates the net external magnetic field using the spin position r and the given time t by Equation (3.7). The result will be the summation of the static magnetic field and the gradient field which is then used in Equation (3.8) to calculate the precession frequency of the spin $w(\vec{r})$. Finally $\vec{M}(t+\Delta t,\vec{r})$ is determined according to Equations (3.9) and (3.10), and the whole calculation is finished in one time step.

During excitation, the solution of Bloch equation gets more complicated. For a rectangular shaped envelope function the Bloch equation has an analytical solution for both on-resonance and off-resonance cases. But for an arbitrary envelope function $B_1^e(t)$, it is not possible to find a closed-loop analytic solution, therefore numerical methods must be utilized. This is performed by the Runge-Kutta solver of Bloch equation simulator.

Bloch equation is an ordinary differential equation which can be written in the form of $dy/dt = f(x,y)$ and $y(x_0) = y_0$. Considering Equation (2.6), the Bloch equation is already in this form with $\vec{M}(0) = \vec{M}_0$. These type of differential equations can be solved with numerical methods like Euler's method, Mid-point method and Runge-Kutta method. In this thesis the 4_{th} order Runge-Kutta method is preferred since it is the most used one in practice because of its high accuracy and less computation time. The ode45() method in Matlab is the most suggested method in numerical solvers package and this methods also implements the 4_{th} order Runge-Kutta method. The 4_{th} order Runge-Kutta method is a modified version of Euler's method which uses 4 derivative values to determine the value of the next point. The value of the function at the next step is calculated as follows:

$$y_{t+h} = y_t + h \times (k_1/6 + k_2/3 + k_3/3 + k_4/6) + O(h^5) \tag{3.11}$$

where the coefficients are determined by:

$$\begin{aligned} k_1 &= \mathrm{f}(x_n, y_n), \\ k_2 &= \mathrm{f}(x_n + h/2, y_n + k_1/2), \\ k_3 &= \mathrm{f}(x_n + h/2, y_n + k_2/2) \\ k_4 &= \mathrm{f}(x_n + h, y_n + k_3) \end{aligned} \tag{3.12}$$

Here the function f is the discrete representation of Bloch equation and calculates the value of the magnetization $\vec{M}(t+\Delta t)$ using the given magnetization $\vec{M}(t)$ vector and the given magnetic field $\vec{B}(t)$ vector as shown in Equation (3.13). The relaxation affects are ignored during excitation.

$$f(M(t), B(t)) = \begin{bmatrix} M_x(t+h) \\ M_y(t+h) \\ M_z(t+h) \end{bmatrix} = \begin{bmatrix} \gamma \cdot (M_y(t) \cdot B_z(t) - M_z(t) \cdot B_y(t)) \\ \gamma \cdot (M_x(t) \cdot B_z(t) - M_z(t) \cdot B_x(t)) \\ \gamma \cdot (M_x(t) \cdot B_y(t) - M_y(t) \cdot B_x(t)) \end{bmatrix} \tag{3.13}$$

The step size h value is critical in the success of these types of numerical methods. If a big step size is chosen, you may be off the track and have big error, on the other hand if a small step size is chosen, you can have small error but the computation time increases. MRI signal is a high frequency signal around 42.58 MHz and the value of $h = 1e-9$ is found to be a good value in terms of accuracy and computation time.

3.4.1 T2* and T2** Simulation

To stimulate T_2^* decay, additional dephasing factors for M_{xy} should be included in the simulation. Among these factors, the static magnetic field inhomogeneity is the main reason behind dephasing. To enable intra-voxel dephasing, a voxel should contain multiple magnetization vectors and each should have a different precession frequency. This can be achieved by changing the static magnetic field

experienced by magnetization vectors in a voxel. Two models are investigated to model B_0 inhomogeneity and simulate T_2^*. In the first one, the static magnetic field B_0 is changed linearly for magnetization vectors, which can be expressed as:

$$B_o(m_i) = B_0 + i\Delta B \tag{3.14}$$

where $B_0(m_i)$ represents the local magnetic field on the i_{th} magnetic vector. This approach produces a smooth intra-voxel dephasing and result in a T_2^* decay for the transverse magnetization but it has been observed that spurious echoes are produced after the signal is completely vanished as mentioned in reference [3]. The spurious echoes can be avoided by using a large number of magnetization vectors per voxel at the cost of increased computation time. Another disadvantage of this method is that it generates T_2 weighted echoes instead of T_2^* weighted echoes for a gradient echo sequence. This is due to the recovery of phase dispersion of the linearly changing precession frequencies which gradient fields which also change the precession frequencies of magnetization vectors linearly. This problem is shown in Results section.

The second model is using a random distribution for B_0 inhomogeneity. This model can be expressed with the following relationships:

$$B_o(m_i) = B_0 + A\Delta B_i \tag{3.15}$$

where ΔB_i values are generated by a pseudo-random generator. This approach is closer to reality and does not generate spurious echoes as in the case of linear changing model. The only problem of this model is that it does not produce a smooth T_2^* decay. The smoothness of T_2^* decay can be enhanced by using more magnetization vectors per voxel at the cost of increased computation time.

3.5 Image Construction

The magnetization values sampled during readout are stored in a K-space buffer. The MRI signal received is a high frequency signal as in Equation (2.18). In

practice, this signal is moved to baseband by a demodulation operation. In the simulation, instead of a demodulator implementation, the static magnetic field B_0 is set to zero during readout. This operation removes the high frequency resonance component from the signal and the baseband signal in Equation (2.19) is obtained. The static magnetic field B_0 is set to its previous value after readout.

K-space data is a complex data. The samples collected during readout are stored in the K-space buffer as follows:

$$Kspace(m,n) = S(m,n)_X + S(m,n)_Y\, j \tag{3.16}$$

where m is the m_{th} phase encoding cycle and n is the n_{th} sample collected in this phase encoding cycle. As seen in Equation (3.15), the x component of transverse magnetization M_{xy} corresponds to the real part and the y component of transverse magnetization corresponds to imaginary part of a k-space sample.

In order to obtain the image, simply a 2D inverse Fourier transform operation is applied to the K-space buffer. The magnitude of the resultant matrix gives us the parameter image function. Depending on the selected pulse sequence parameters, the output image may be the proton density function, T_1 function or T_2 function of the selected slice.

CHAPTER 4

SIMULATION SOFTWARE ARCHITECTURE

AND USER INTERFACE

One of the motivations behind this study was to create an educational tool for students and clinicians. For this aim, the simulation application is designed with an educational perspective in terms of user interface and software design. The simulation is composed of two applications. One is called “Spin simulator” and the other is called “MRI Simulator”. The reason behind developing two applications is to show all aspects of MRI as an imaging technique. The “MRI Simulator” is the final product of this study and it generates images from 3D virtual objects based on the pulse sequence parameters selected by the user. On the other hand, the “Spin simulator” allows one to study NMR phenomenon by applying excitation and relaxation events to a virtual object composed of one voxel.

This chapter explains the software architecture and user interface of the simulation software. The first section focuses on the software design of the simulation in terms of java, applets and object oriented approach. The second section explains the user interfaces of the applications in the simulation software.

4.1 Software Architecture

Besides the aim of being an educational tool, this application also aims to be easily accessible and practical for users. The most accessible environment in today's technology is the web environment. Therefore the applications of simulation are developed as Java applets which are executed inside web browsers. Java applets are java applications which run completely at the client side and can easily be embedded into a web page. Most of the today's popular web browsers support java applets. So only an internet connection is needed to run this simulation.

Java is an object oriented, platform independent and high performance software language. By its high level and well-organized APIs, it provides a fast and reliable development environment for developers. With all these advantages, java became a popular language for academic applications.

By use of the Java technology, the simulation software has been developed with an object-oriented approach. The user interface components and the kernel of the simulation are nicely separated. If desired, the simulation kernel easily be integrated with another user interface. The user interface and the simulation kernel run in separate tasks so that the user interface elements are accessible when the simulation is running. This allows the simulation to be terminated anytime.

The simulation kernel has been designed to be an MRI simulation framework rather than a specific application code. Any pulse sequence type can easily be realized in this framework using the following utility classes.

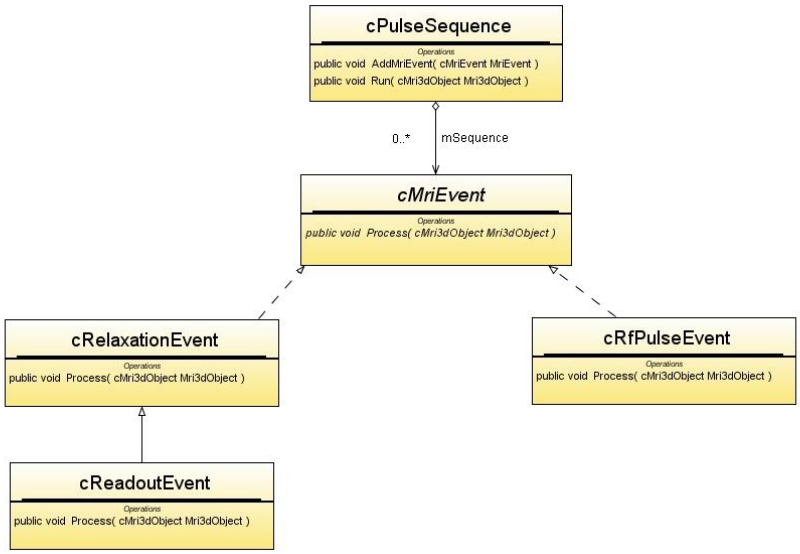

Figure 4.1 – Utility classes to implement a pulse sequence

As shown in Figure 4.1, the cPulseSequence class holds a list of MRI events. By *AddMriEvent()* method, relaxation or excitation can be added to the list to form a series of events. The events in the list are processed consecutively by the *Run()* method by calling the *Process()* method of each event in the list. With this structure, any pulse sequence can easily be implemented and executed.

4.2 Application User Interfaces

The user interfaces of both applications have been designed to be interactive and informative for the user. For this purpose, when a parameter is updated by the user, the affected parameters are updated immediately by the application to inform the user what that parameter affects.

4.2.1 Spin Simulator

The user interface of the "Spin simulator" application is shown in Figure 4.2.

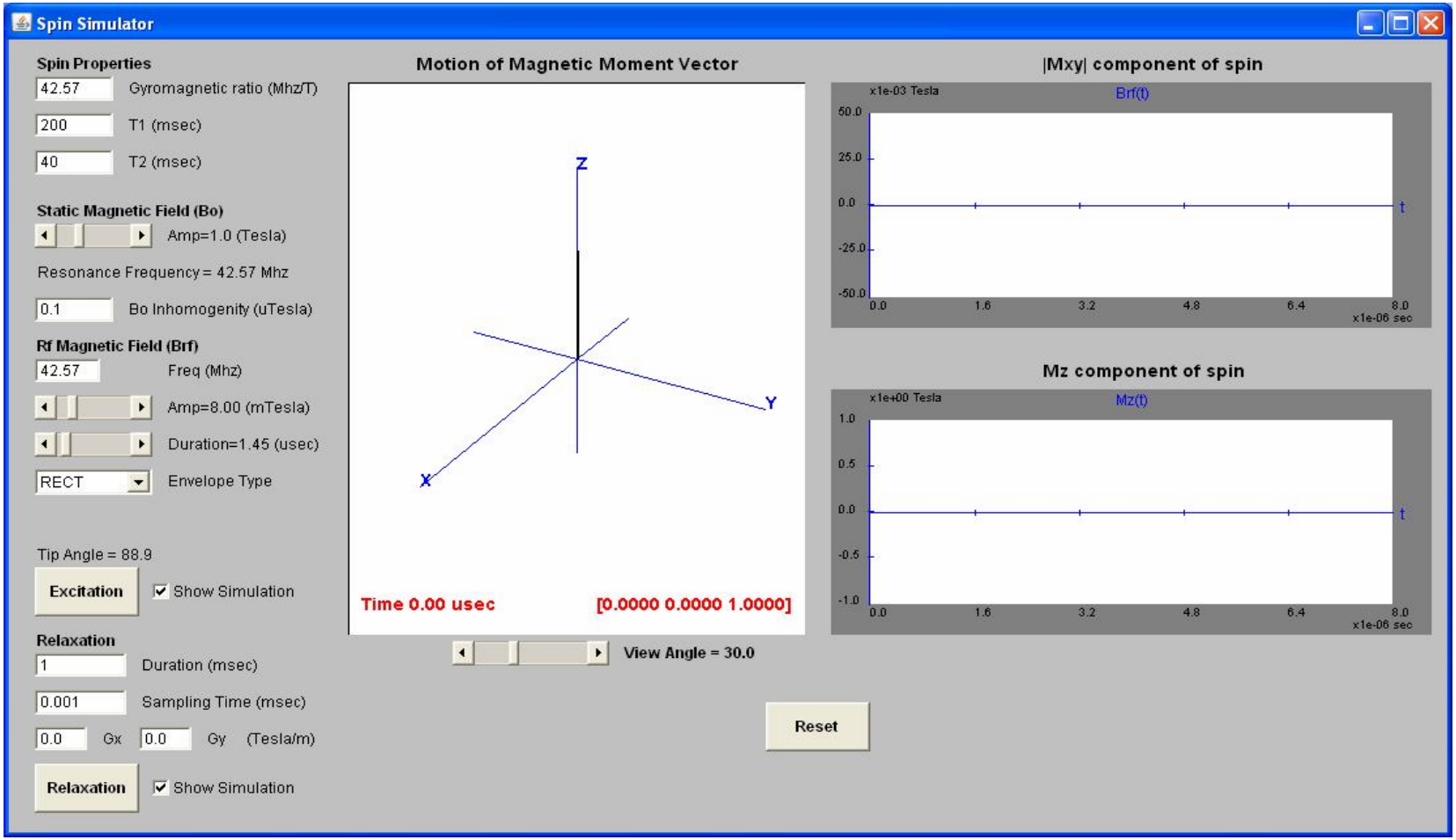

Figure 4.2 – User interface of the Spin Simulator

This application provides user the chance to apply RF magnetic field and relaxation to a virtual object consecutively and observe the magnetization vector $\vec{M}$ on 3D and 2D displays. As an example, the user can see the affect of a spin-echo on a voxel by first applying a 90 degree excitation pulse, then a relaxation for some duration, then a 180 degree pulse and finally a relaxation which has a longer duration than the previous relaxation. The magnetization of the voxel is not reset unless "Reset" button is pressed. As the name of the application implies, this simulation works on one voxel which is composed of only one spin. The voxel has 64 magnetization vectors which are distributed evenly in the voxel volume. The voxel width is chosen to be 0.01 meters.

Through the user interface, the user can set the spin parameters, static magnetic field value, static magnetic field inhomogeneity value, RF magnetic field parameters and relaxation parameters. The RF magnetic field parameters include frequency, amplitude, duration, and envelope function type. Possible choices are rectangular and sinc shaped envelope functions. If the selection is sinc, the bandwidth is asked to the user as an additional parameter. Duration, sampling time and gradient field values in the direction of $\vec{x}$ and $\vec{y}$ (Gx, Gy) are the parameters of relaxation. The sampling time parameter defines the sampling period or readout period during relaxation.

The application has three displays. The one at the left with the label "Motion of Magnetic Moment Vector" shows the 3D motion of the magnetization vector. It has a view angle scroll bar which the user can change the view angle from 0 to 90 degrees. At the bottom left corner of the display, the simulation time is shown and at the bottom right corner, the magnetization vector $\vec{M}$ is displayed in a vector format (M_x, M_y, M_z). The display at the top right of application window is a 2D display and its displays different inputs for excitation and relaxation. The display at the bottom left is also a 2D display and shows the value longitudinal magnetization M_z versus time. The 2D displays have axis, scale and label

information which provide the user, time information for the x coordinate and unit information for the y coordinate.

The “Excitation” button starts an RF excitation to the magnetized virtual object with the RF magnetic field parameters selected. The parameters are passed to the simulation kernel at the instance the button is pressed. Thus changing the parameters during the simulation will have no effect. The static magnetic field and the static magnetic field inhomogeneity values are regarded when the application is first invoked and get updated if the user changes these fields. A change in static magnetic field value updates “Resonance Frequency” field’s value. Before starting excitation, the user first configures the excitation parameters by playing with slide bars and text boxes. As the user changes amplitude and duration of the RF pulse, the tip angle field is updated with the new tip angle value. If the selected envelope function is sinc, changes in bandwidth field, also updates the tip angle. The tip angle is calculated according to (2.10) if the envelope function type is rectangular. If the sinc shaped envelope function is selected, the tip angle is calculated by the integration of the sinc function over the excitation interval. The integration is performed using trapezoid method. During excitation, the top right display shows the amplitude of RF magnetic field amplitude versus time. This is shown in Figure 4.2. The 3D display and 2D displays are updated at every 100 steps of the simulation kernel during excitation. The Bloch equation simulator is configured with a step size of $h = 2e-10$ for this application. Gradient field values are also taken into account during excitation. If they are different than zero, it will be observed that the flipped magnetization vector will have a phase dispersion.

The “Relaxation” button turns off the RF magnetic fields and starts a simulation by relaxing the voxel for the specified duration with a step size equal to sampling time field. The effect of gradient fields can be observed by setting the gradient field values. During relaxation, the top right display shows the value of

transverse component $\vec{M}_{xy}$ of magnetization vector versus time. Displays are updated at the period of sampling time field during the process of relaxation.

The "Reset" button stops the simulation, resets the magnetization vector to its initial value and clears the displays. It can be clicked anytime. It also sets the simulation time to zero. The 3D display shows magnetization vector after it is reset and the 2D displays are cleared after "Reset" button is clicked. "Reset" operation does not bring the parameters back to their default values.

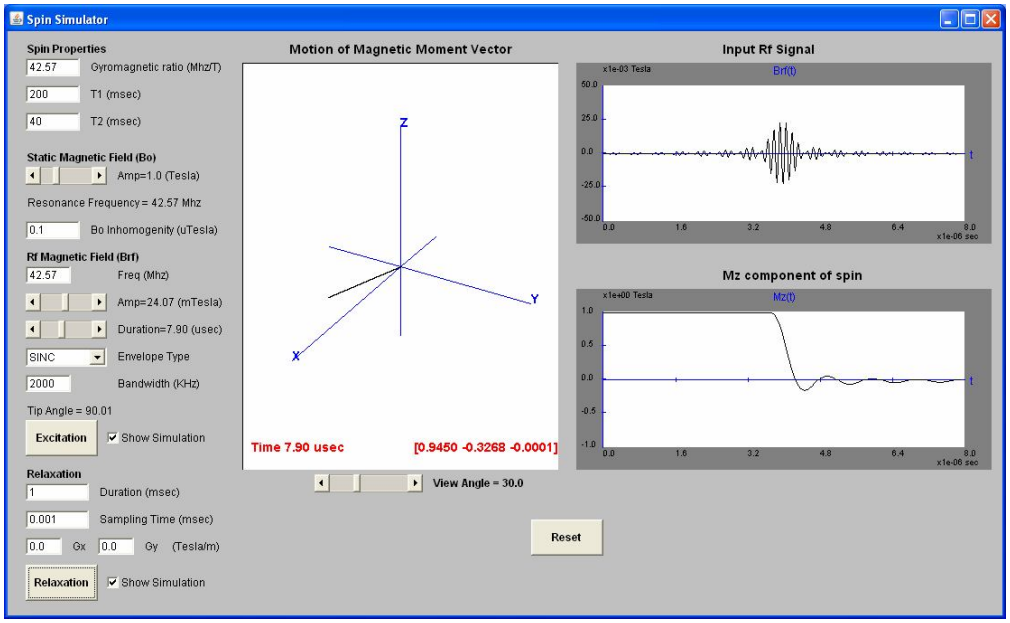

Figure 4.3 – User interface of the Spin Simulator while the simulation is running

4.2.2 MRI Simulator

The user interface of the MRI simulator application is shown in Figure 4.4.

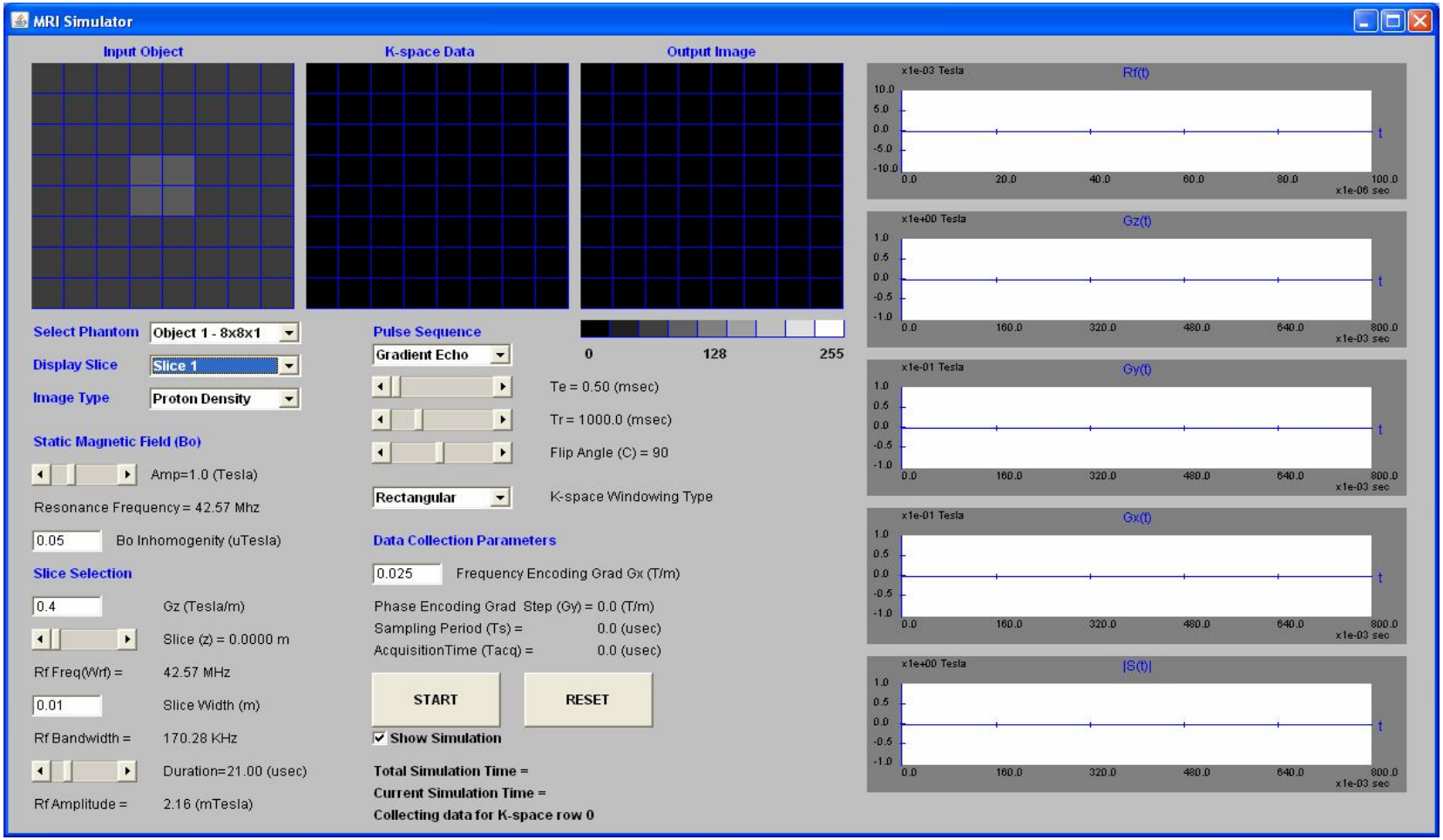

Figure 4.4 – User interface of the MRI Simulator

This application is the main output of this study. It allows the user to apply a pulse sequence and slice selection pulse on a virtual 3D object and see the output image. The user is able to select a 3D virtual object inside a set of predefined virtual 3D objects.

As seen in Figure 4.4, the application does not have a menu and all the displays and user interface elements are put into one window. On the upper left side, there are 3 image displays showing the selected slice and parameter image of input object, the k-space data and the output image respectively. On the right side of the window, there are five 2D displays placed from top to bottom. These windows show the change of RF signal versus time ($Rf(t)$), gradient field change along $\vec{z}$ direction versus time ($G_z(t)$), gradient field change along $\vec{y}$ direction versus time ($G_y(t)$), gradient field change along $\vec{x}$ direction versus time ($G_x(t)$), and the magnitude of received signal versus time ($|S(t)|$) respectively. When the application is started up, all the image displays and 2D displays are cleared except the input object display. The application is simple to use. The user selects an input image, configures the slice selection and pulse sequence parameters and presses the "START" button to start the simulation. Except the input object display, all the displays are updated during the simulation according to their content. The user can stop the simulation anytime by pressing the "RESET" button.

Below the input object display, there are three combo boxes which allow the user to select an input object and to view the image of the selected parameter and selected slice. Through the "Select Phantom" combo box, the user chooses the virtual 3D input object which the simulation will be performed for. Each item of the combo box shows the dimensions of the input object in the $MxNxK$ format where M is the number of voxels in $\vec{x}$ direction, N is the number of voxels in $\vec{y}$ direction and K represents the number of slices. The parameter images of 3D virtual objects are shown slice by slice. The slice selection is done with "Display

slice" combo box. The number of items in the "Display slice" combo box is updated according to the number of slices in the selected input object. The selected slice is displayed in the input object display. MRI is capable of imaging proton density, T_1 and T_2 maps of tissues. As explained before, each voxel in the virtual object has proton density, T_1 and T_2 information. The third combo box "Image Type" allows user to select which parameter image will be displayed.
The user can change the static magnetic field value using the "static magnetic field" scroll bar. The resonance frequency field is updated as the static magnetic field value is changed. The static magnetic field inhomogeneity value is the inhomogeneity constant for B_o distribution. The B_o inhomogeneity distribution is calculated each time a simulation is started .

In the slice selection part of the user interface, the user can configure the $\vec{z}$ gradient G_z, the slice center and the slice width. By changing the "Slice(z)" scroll bar, the user chooses the center of the slice along the $\vec{z}$ direction. The height of a virtual object is determined by the multiplication of number of slices and the voxel width which is chosen as 0.01m. For a virtual object with 4 slices, the height of the object will be 0.04 m. The slice center should not be above the object height in order to receive signal from the object. The slice center determines the frequency of the RF magnetic field according to the formula $w_{rf} = \gamma(B_0 + G_z z_c)$ where z_c is the center of the slice and G_z is the gradient field along $\vec{z}$. The slice width specifies the region around the slice center to be excited by the RF pulse and determines the bandwidth of the RF pulse according to the formula $\Delta w = \gamma G_z \Delta z$ where Δz is the slice width. The slice width should also be selected carefully considering that the voxel width is 0.01m. If the slice width is chosen a value higher than voxel width, the output image becomes the addition of two more slices. Besides these configurable parameters, there are RF pulse frequency, RF pulse bandwidth, RF pulse duration and RF pulse amplitude fields in the slice selection part. The values of these fields are calculated by the simulation software and shown to the user when the "START" button is pressed.

The pulse sequence part of the user interface provides to user to select a pulse type and configure its parameters. The configurable parameters are time of echo (TE), time of repetition (TR), and flip angle. TE and TR determine which parameter of tissue will be weighted among proton density, T_1, and T_2. TE vale cannot be chosen greater than TR.

The "K-space Windowing Type" field determines the type of windowing operation to be applied on the k-space data. If "Rectangular" type is selected, no windowing operation is applied. If Hamming is selected, a Hamming window is applied to each row of k-space. As the name implies, the "Frequency Encoding Grad Gx" field is used as the frequency encoding gradient value during pulse sequence processing.

When the user press "START" button, the application takes the currently selected input object, pulse sequence parameters, slice selection parameters, the k-space windowing type and G_x value, and performs the simulation using these settings. Changing the parameters will not have any affect during the simulation. When the simulation is started, firstly the simulation software calculates the data collection parameters and displays them in the "data collection parameters" part of the user interface. These parameters are phase encoding gradient step size, sampling period used in readout, and total data acquisition time. The acquisition time calculated as the number of samples taken in readout times the sampling period. Also the total collection time is calculated as the number of phase encoding steps times TR and written into the "Total Simulation Time" field.

There is a check box under "START" button called "Show Simulation". If this check box is not selected, the image displays and the 2D displays are not updated during simulation. Only the "Current Simulation Time" and "Collecting Data for K-space Row" fields are updated throughout the simulation. At the end of the simulation, the k-space data and the output image are displayed. This mode enables to obtain the output image in a shorter time. If the "Show Simulation"

check box is selected, the 2D displays and the k-space image display are updated during the simulation. This is shown in Figure 4.5. The 2D displays show the current state of the magnetic inputs and the readout signal. During excitation, the displays are updated every microsecond. For relaxation events, the displays are updated before and after each event. The duration of the gradient fields determine the width of the gradient pulses on the displays. The pulse widths are calculated by dividing the duration of the gradient fields to 1 μsec. Since free relaxation times are on the order of milliseconds, they cannot be drawn on 1 microsecond scale otherwise one phase encoding cycle does not fit into display window. To overcome this problem, the free relaxation periods are shown with 50 pixels width on the displays. So the overall phase encoding cycle shown on the displays is not to scale. The 2D displays are updated every 1 μsec except free relaxation (no gradient field applied) times. The K-space data display shows which point of k-space is being filled during readout. This is performed by blinking the pixel of k-space which is being filled and then setting it to its value. After all the k-space is filled, the output image display is updated with the output image.

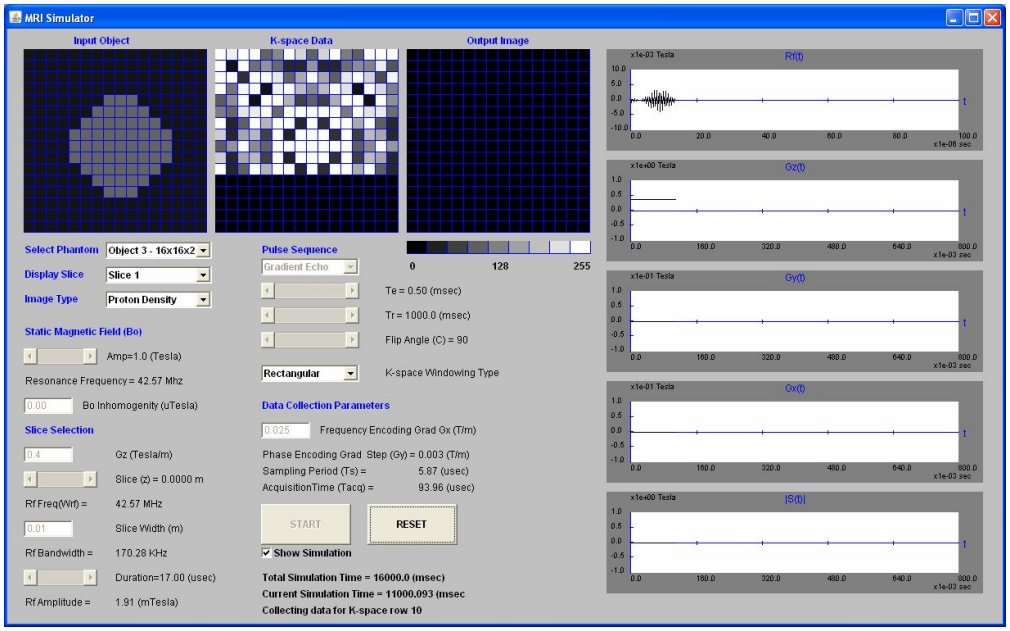

Figure 4.5 – User interface of the MRI Simulator while simulation is running

The “RESET” button stops the simulation, sets the virtual object to its initial state and clears the displays except the input object display. It can be clicked anytime. It also sets the simulation time to zero. “RESET” operation does not bring the parameters back to their initial values.

CHAPTER 5

SIMULATION RESULTS AND DISCUSSION

The first simulation program "Spin Simulator" allows the user to observe the NMR phenomenon with different excitation and relaxation parameters on a voxel. The second program "MRI Simulator" generates MRI images from a virtual 3D object by using the pulse sequence and slice selection parameters provided by the user. In this chapter, outputs of experiments from both applications are presented and compared with the expected results.

The results for the "Spin Simulator" application are classified according to specific NMR situations. The results for "MRI Simulator" contain MRI image outputs for different slice selection and pulse sequence parameters.

5.1 Results from Spin Simulator

The virtual object used in the "Spin Simulator" program is a voxel whose structure is defined in section 3.1. The voxel is represented by 64 magnetization vectors. The voxel has only one spin whose parameters are given by the user. The default spin parameters and proton density are chosen as $\gamma = 42.57$ MHz/Tesla, $T_1 = 200$ msec , $T_2 = 40$ msec and, proton density p=1.0 and these values are used for the experiments presented here.

5.1.1 Resonance

This experiment is performed with the following parameters: $B_0 = 1.0$ Tesla, a rectangular shaped RF pulse with frequency $f_{rf} = 42.57$ MHz, amplitude = 8.1 mTesla, and duration = 1.45 μsec which creates a $90°$ pulse.

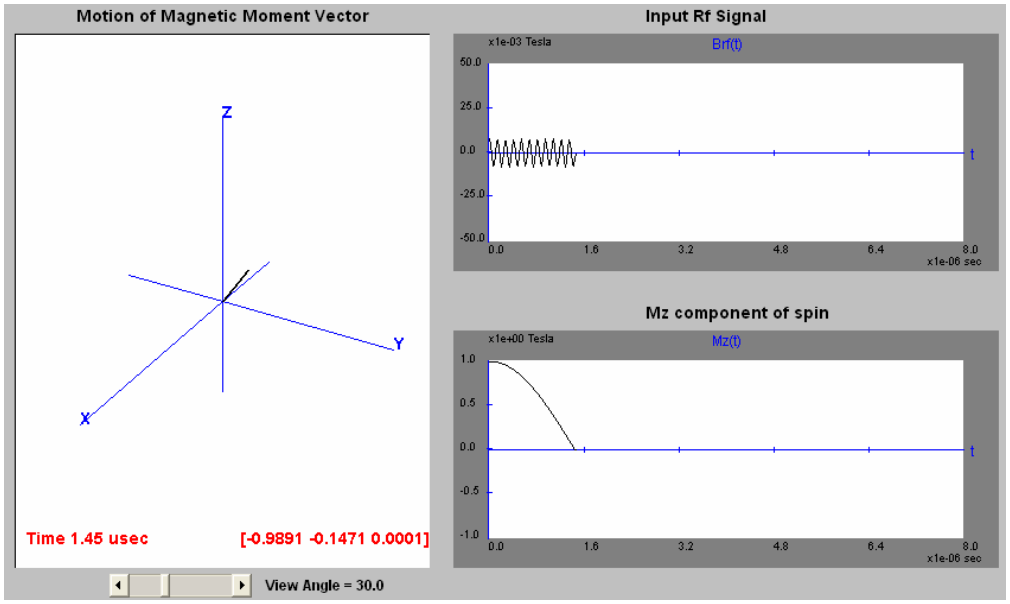

Figure 5.1 – On-Resonance excitation: Bo = 1.0 Tesla, a rectangular shaped RF pulse with frequency 42.57 MHz, amplitude = 8.1 mTesla, and duration = 1.45 μsec.

As seen in Figure 5.1, the magnetization vector was tilted away from $\vec{z}$ axis by the $90°$ pulse as expected. To verify the accuracy of the numerical method (Runge-Kutta 4th order) used in the solution of Bloch equation, the Bloch equation has been solved by the Matlab's *ode45 ()* method which uses also uses a Runge-Kutta implementation for solution of ordinary differential equations. Matlab simulation gave the same result for the magnetization vector [-0.9891 -0.1471 0.0001]. This result has also been compared with the analytic solution which produced the result of [-0.9891 -0.1471 0.0000]. The slight difference between the

analytic and numerical solution is due to T_1 relaxation which is ignored in the analytic solution.

It should also be noted that the simulation program uses a step size of $h = 2e-10$ in the Runge-Kutta solution which tracked the analytic solution successfully. When the step size is taken as $h = 1e-9$, the simulation produces the result of [-0.9857 -0.1569 0.0014] which deviates from the real solution slightly. Decreasing h increases the accuracy but also increases the simulation time by $O(n^3)$. To find out how much deviation occurs from the exact solution by increasing h, a $180°$ pulse is applied to the system with frequency $f_{rf} = 42.57$ MHz, amplitude = 8.1 mTesla, and duration = 2.90 μsec with $h = 1e-9$. The simulator produces the result [-0.0196 -0.0058 -0.9961] in this case. It is observed that the step size value can be taken as $h = 1e-9$ by sacrificing a %.004 error in the $\vec{z}$ component of magnetization vector.

To see what happens when the envelope function is not rectangular, the system is excited with a sinc shaped RF pulse with frequency $f_{rf} = 42.57$ MHz, amplitude = 24.07 mTesla, duration = 7.90 μsec and bandwidth = 2000 KHz which corresponds to $90°$ pulse. The excitation signal $\vec{B}_1(t)$ and the evolution of the $\vec{z}$ component of magnetization vector can be seen in the figure below.

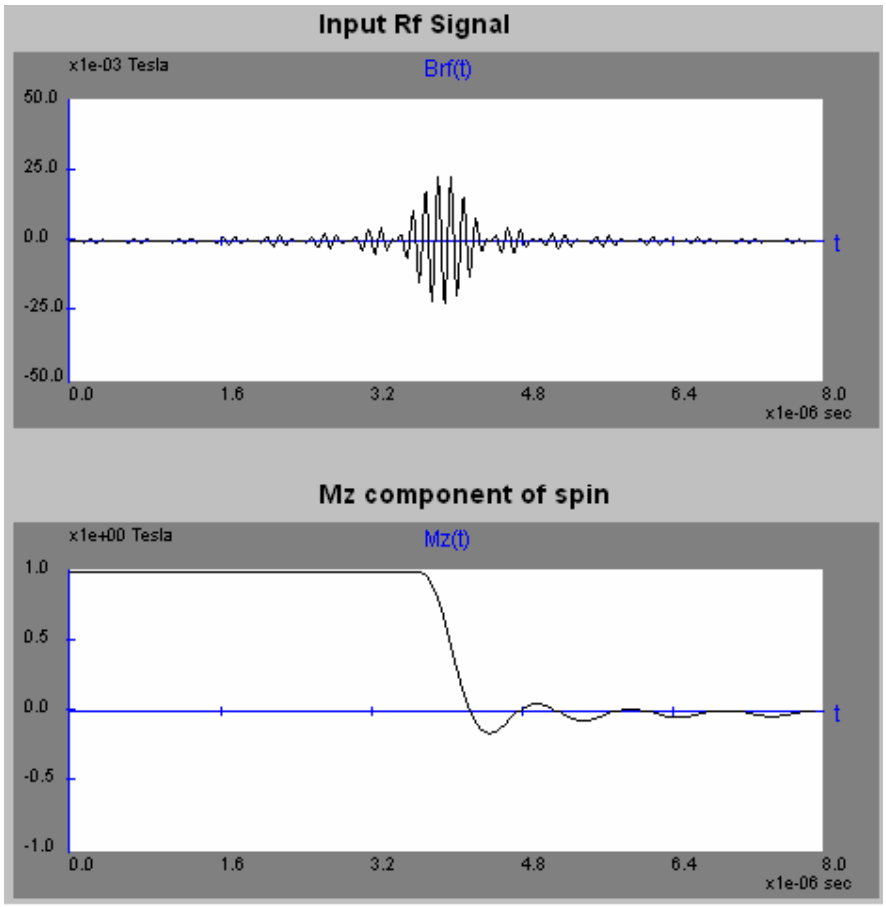

Figure 5.2 – On-Resonance excitation with sinc shaped envelope function: frequency 42.57 MHz, amplitude = 24.07 mTesla, duration = 7.90 μsec and bandwidth = 2000 KHz.

As seen in Figure 5.2, during the side lobes of $\vec{B}_1(t)$, $\vec{M}(t)$ keeps its equilibrium value. During the main lobe, $\vec{M}(t)$ moves to transverse plane quicker compared to rectangular pulse excitation (Figure 5.1). After the main lobe, a small fluctuation around xy plane is observed. It is interesting to observe that the side lobes of sinc pulse has little affect on the magnetization vector.

5.1.2 Off-Resonance

To observe the off-resonance case, the object is excited with the following parameters: $B_0 = 1.0$ Tesla, a rectangular shaped RF pulse with frequency $f_{rf} = 42.87$ MHz, amplitude = 8.1 mTesla, and duration = 1.45 μsec which would be a $90°$ pulse if it were an on-resonance excitation. The output of the simulator is shown in the figure below.

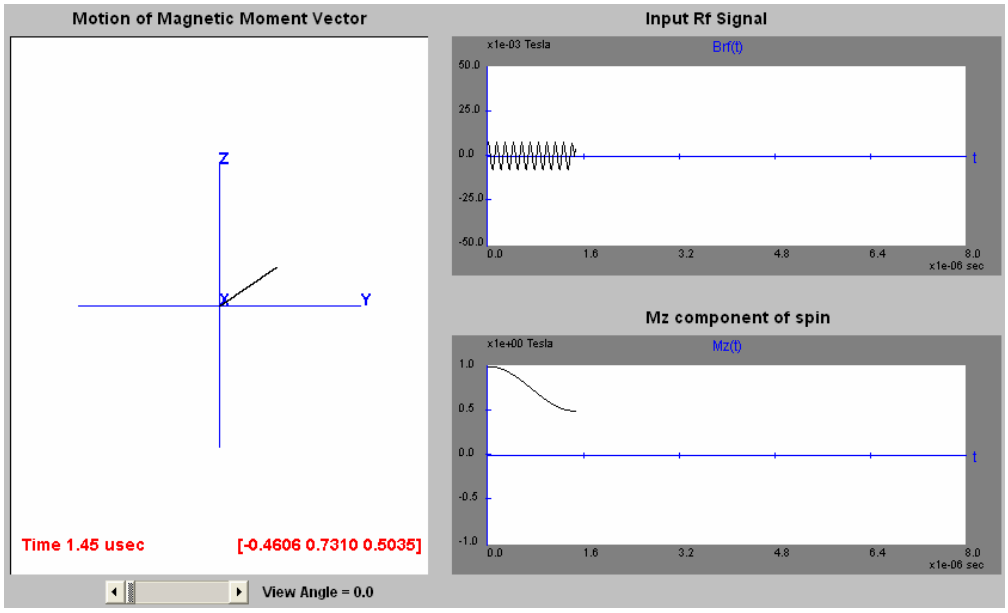

Figure 5.3 – Off-Resonance excitation, Δw small: performed with frequency $f_{rf} = 42.87$ MHz, amplitude = 8.1 mTesla, and duration = 1.45 μsec. The left display shows how much the magnetization vector tipped on the $\vec{z} - \vec{y}$ plane.

Although the excitation frequency is 0.30 MHz apart from resonance frequency, the magnetization vector $\vec{M}(t)$ still tilts into the transverse plane by a significant amount. This is expected as explained in Equation (2.12).

If the excitation frequency is selected further away from resonance frequency, $\vec{M}(t)$ moves away from $\vec{z}$ axis slightly as shown in Figure 5.4. The following

result is obtained by application of a rectangular shaped RF pulse with frequency $f_{rf} = 43.17$ MHz, amplitude = 8.1 mTesla, and duration = 2.90 μsec which would cause a $180°$ pulse if it were an on-resonance excitation. The 0.60 MHz deviation form resonance frequency resulted in almost no transverse magnetization for spin system.

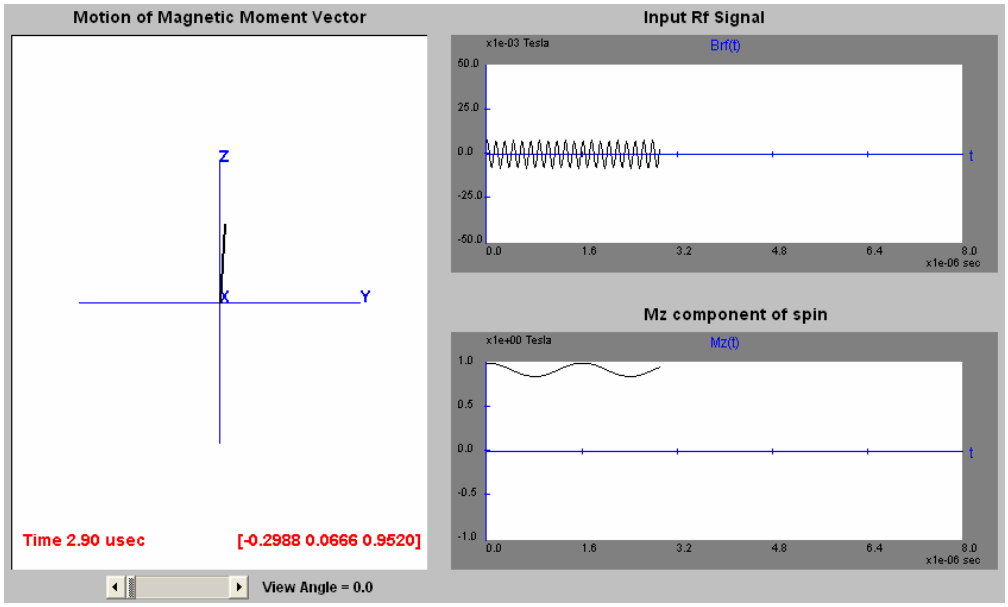

Figure 5.4 – Off-Resonance excitation, Δw big: performed with with frequency $f_{rf} = 43.17$ MHz, amplitude = 8.1 mTesla, and duration = 2.90 μsec. The left display shows how much the magnetization vector tipped on the $\vec{z} - \vec{y}$ plane.

5.1.3 Relaxation

In all the relaxation results presented below, the system is first excited with a $90°$ rectangular pulse with the following parameters: frequency $f_{rf} = 42.57$ MHz, amplitude = 8.1 mTesla, and duration = 1.45 μsec. Then samples are taken from the spin system for different scenarios.

5.1.3.1 T1 and T2 Relaxation

If the static magnetic field B_o was homogenous, then all the magnetization vectors in the object would be rotating with the same precession frequency and the net transverse magnetization vector would be decaying with T_2. To illustrate this, the B_o inhomogeneity constant is set to 0 for this experiment and the system is put into relaxation with a sampling period of 0.2 msec for 40 msec which equals to T_2 constant of the spin.

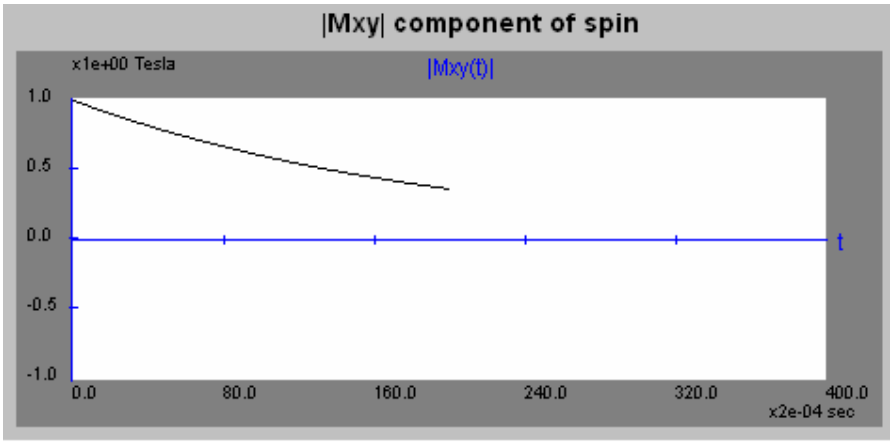

Figure 5.5 – T_2 relaxation for 40 msec with a sampling period of 0.2 msec.

As seen in Figure 5.5, the transverse magnetization vector loses %63 of its magnitude after 40 msec relaxation as expected.

To show T_1 relaxation, the system is relaxed for 200 msec which equals to T_1 constant of the spin and samples are taken at 0.5 msec intervals. The gradient fields are set to zero in both cases. This is shown on the figure below together with the transverse magnetization $|M_{xy}|$ change.

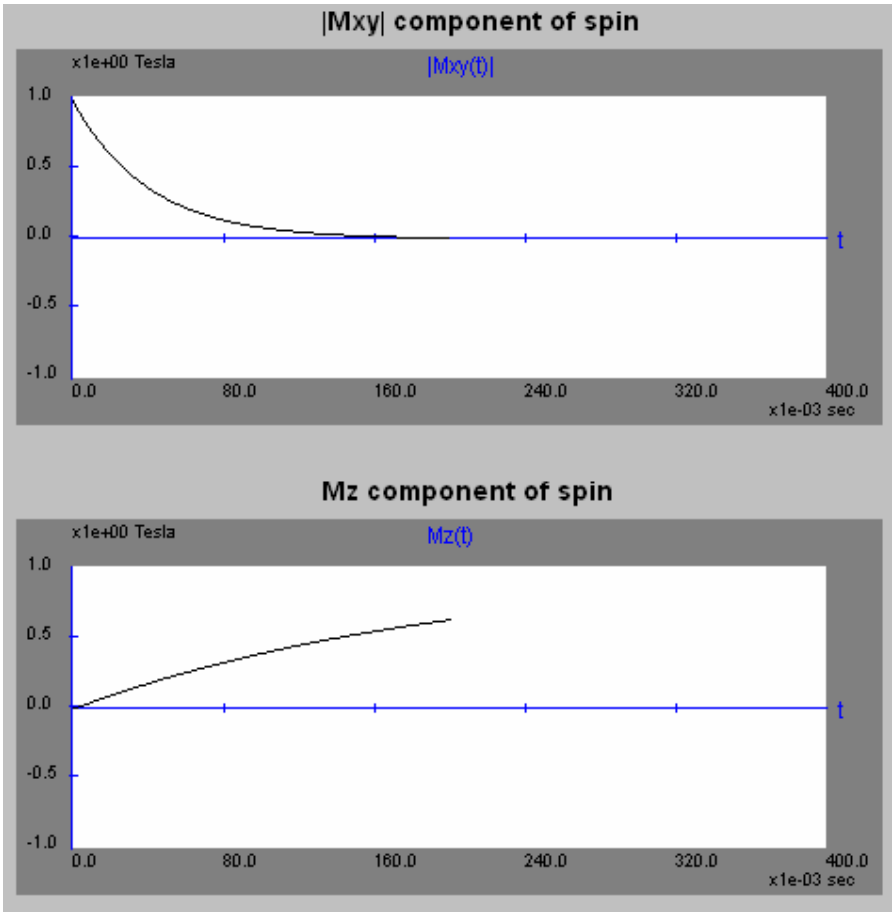

Figure 5.6 – T_1 and T_2 relaxation: the top display shows $M_{xy}(t)$ vs. t and presents T_2 relaxation, the bottom display shows $M_z(t)$ vs. t and presents T_1 relaxation. The system is relaxed for 200 msec. with a sampling period of 0.5 msec.

As seen on Figure 5.6, the longitudinal magnetization M_z gains %63 of its magnitude after a relaxation of 200 msec. which is the T_1 constant of the system. The transverse magnetization M_{xy} vanishes after 200 msec relaxation as expected. As observed, the transverse magnetization diminishes early before the longitudinal magnetization is recovered.

5.1.3.2 T2* Relaxation

During relaxation, the individual magnetic moments in a voxel have different precession frequencies because of B_0 inhomogeneity. To create this inhomogenity, the B_0 inhomogeneity value is set to 0.1 μTesla. As a result of this inhomogeneity, each magnetization vector in the voxel will have a different precession frequency determined by Equation (3.15). This is illustrated in the figure below.

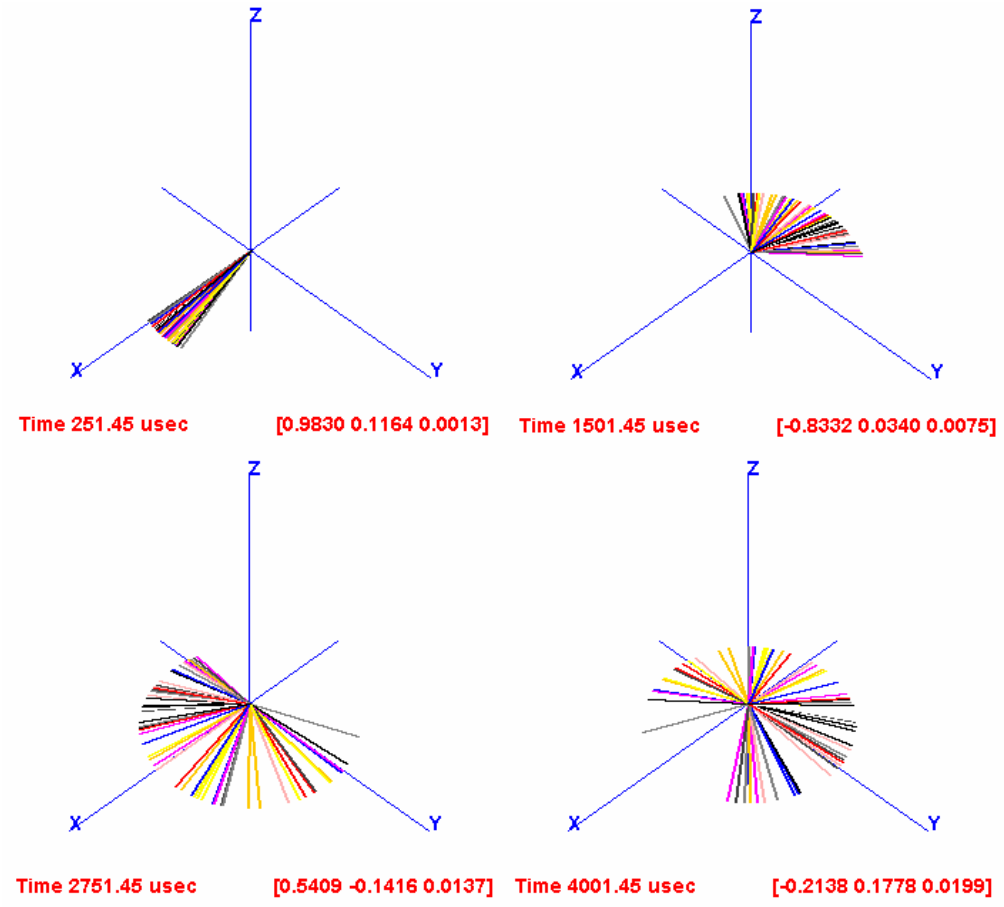

Figure 5.7 – The phase dispersion of magnetic moment vectors with time: the top-right display shows the distribution of magnetization vectors at 0.25th msec, the top right display shows the distribution at 1.50th msec, the bottom left display shows the distribution at 2.75th msec and the bottom right display shows the distribution at 4.0th msec.

Figure 5.7 contains four snapshots taken from the simulation program at different times. The first snapshot is taken after 0.250 msec with a sampling rate of 0.01 msec. Then the system is relaxed for 1.250 msec more and the second snapshot is taken. The third snapshot is taken after 1.250 msec from the second snapshot. The fourth snapshot is taken after another 1.250 msec of relaxation which makes a total of 4.0 msec relaxation.

The voxel is composed of 64 magnetization vectors. So the total magnetization vector will be composed of 64 different precession frequencies. This is shown in Figure 5.7 by displaying each of them with a separate color. As time progresses, the phase difference between magnetization vectors increases and this causes the transverse magnetization vector M_{xy} to decay at a rate of T_2^* instead of T_2. This is shown in the figure below where the system is relaxed for 40 msec with a sampling interval of 0.1 msec.

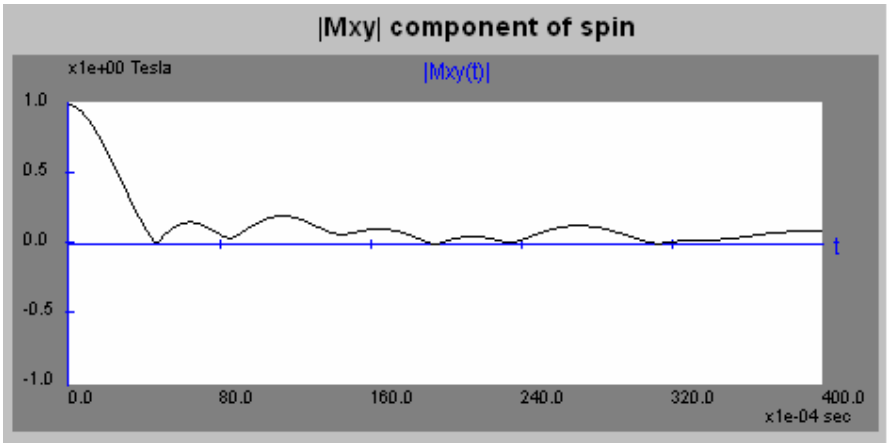

Figure 5.8 – The decay of $M_{xy}(t)$ vs. t with T_2^* when ΔB =0.1 μTesla: the system is relaxed for 40 msec with a sampling interval of 0.1 msec.

As seen in Figure 5.8, the magnitude of transverse magnetization vector vanishes after 5 msec which should normally vanish after two T_2 intervals. In a normal

MRI system, the transverse magnetization decays exponentially at a rate of T_2^*. Physically, there are millions of magnetization vectors in a voxel and this leads to an exponential decay for net transverse magnetization. But in the simulation, a voxel is represented by only 64 magnetization vectors because of computation issues and the phase dispersion of these magnetization vectors create side lobes after the main decay which is not seen in reality. If more magnetization vectors are used to represent a voxel, the decay function approaches to an exponential decay.

Another thing that should be underlined is that the B_0 inhomogeneity value determines the rate of decay. This is shown in Figure 5.9 where the B_0 inhomogeneity value is set to 0.2 μTesla and the system is relaxed for 40 msec with a sampling rate of 0.1 msec.

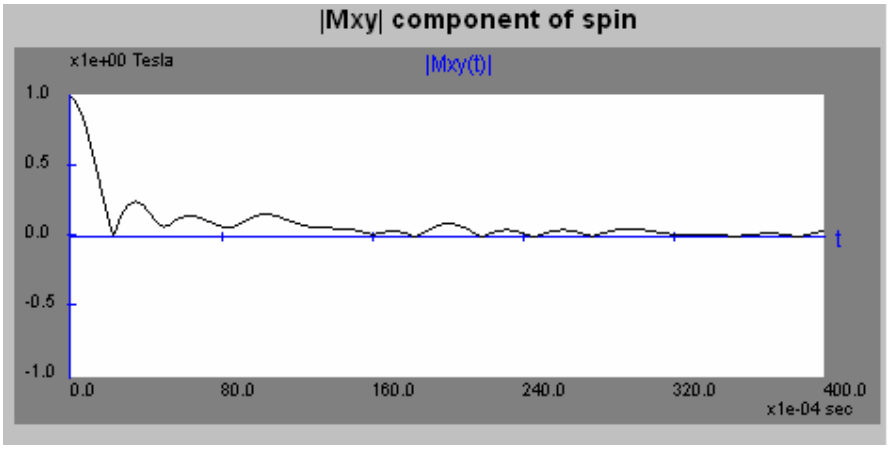

Figure 5.9 – The decay of $M_{xy}(t)$ vs. t with T_2^* with ΔB =0.2 μTesla: the system is relaxed for 40 msec with a sampling rate of 0.1 msec.

As seen in Figure 5.9, the transverse magnetization decays at a faster rate compared to Figure 5.8. The first zero crossing of M_{xy} occurs at 2.5 msec in this

case where it occurs at 5.0 msec when the B_0 inhomogeneity value is set to 0.1 μTesla.

5.1.4 Echo Generation

Echo generation is the key step of MRI process. In this part, two methods of echo generation spin echo and gradient echo are presented.

5.1.4.1 Spin Echo

To generate a spin echo, first the system is excited with a rectangular shaped RF pulse with frequency $f_{rf} = 42.57$ MHz, amplitude = 8.1 mTesla, and duration = 1.45 μsec which corresponds a $90°$ pulse. The B_0 inhomogeneity value is set to 0.2 μTesla. After the excitation, the system is relaxed for 5 msec with a sampling rate of 0.1 msec. The transverse magnetization vanishes after this relaxation due to T_2^* decay. After this a $180°$ pulse is applied to recover the phase dispersion. The parameters of the 180 pulse are as follows: frequency $f_{rf} = 42.57$ MHz, amplitude = 16.2 mTesla, and duration = 1.45 μsec. Finally the system is relaxed for duration of 10 msec in which an echo is generated at the 10th msec. Figure 5.9 shows the phase recovery of magnetization vectors just before the echo generation. This relaxation -$180°$ pulse - relaxation sequence is called a spin echo sequence. This sequence is applied to the system two times which generates the two echoes shown in Figure 5.10.

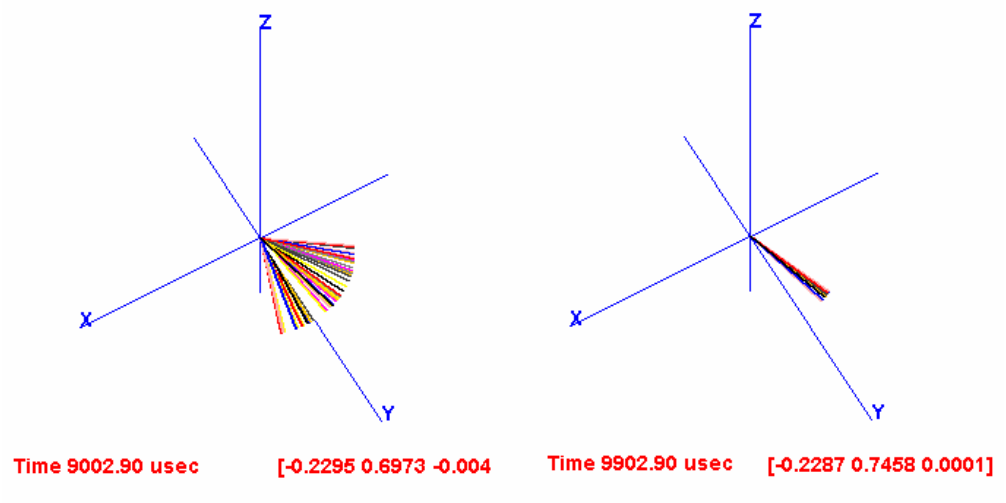

Figure 5.10 –Phase recovery after 180° pulse: the left display shows the distribution of magnetization vectors at time instance 9.0 msec and the right display shows the distribution of magnetization vectors at time instance 9.9 msec just before the echo generation.

As seen in Figure 5.10, the phase dispersion decreases as approaching to the echo time. The 180° pulse maintains the phase total coherence between magnetization vectors and causes echo generation.

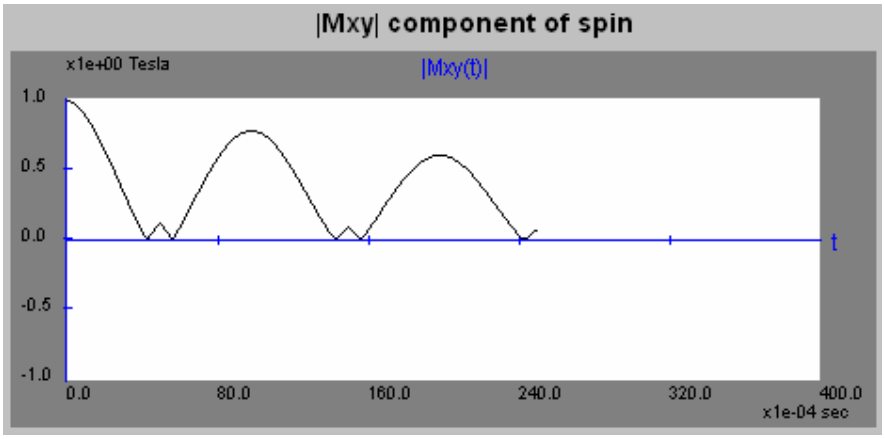

Figure 5.11 – Spin echo simulation: performed with a sequence of 5 msec relation - 180 pulse - 10 msec relaxation-180 pulse – 10 msec relaxation. Two echoes are observed at time instances 10 msec and 20 msec.

As seen in Figure 5.11, the peaks of echoes are at 10 msec and 20 msec as expected. The time instants where the $180°$ pulses are applied can be observed by the abrupt change of phase at 5 msec and 15 msec. If this sequence is applied many times consecutively, it is observed that the peaks of the echoes follow the exponential decay of magnetization vector with T_2. This is shown in Figure 5.12 where the $180°$ pulse - relaxation sequence is applied four times after an intial relaxation with an echo time of 8 msec.

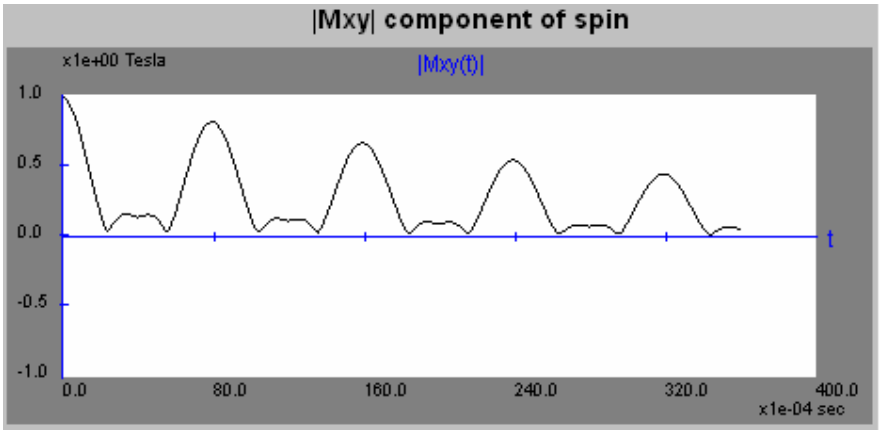

Figure 5.12 – Spin echoes following T_2 decay: performed with a sequence of 180 pulse - 8 msec relaxation applied 4 times after an initial relaxation of 4 msec.

5.1.4.2 Gradient Echo

Gradient echoes are generated by spoiling the transverse magnetization with a gradient field and recovering it with another gradient field. To generate a gradient echo, first the system is excited with a rectangular shaped RF pulse with frequency $f_{rf} = 42.57$ MHz, amplitude = 8.1 mTesla, and duration = 1.45 μsec which corresponds a $90°$ pulse. The B_0 inhomogeneity value is set to 0.1 μTesla. After the $90°$ pulse, the system is relaxed for 0.050 msec with a sampling time of 0.001 msec and Gx gradient field set to 0.1. After this 0.050 msec relaxation, the Gx gradient value is set to -0.1 Tesla/m and the system is put into relaxation for

duration of 0.100 msec sampled at an interval 0.001 msec. The phase dispersion and recovery of magnetization vectors and, the change of transverse magnetization $M_{xy}(t)$ for this sequence are shown in Figure 5.13 and 5.14, respectively.

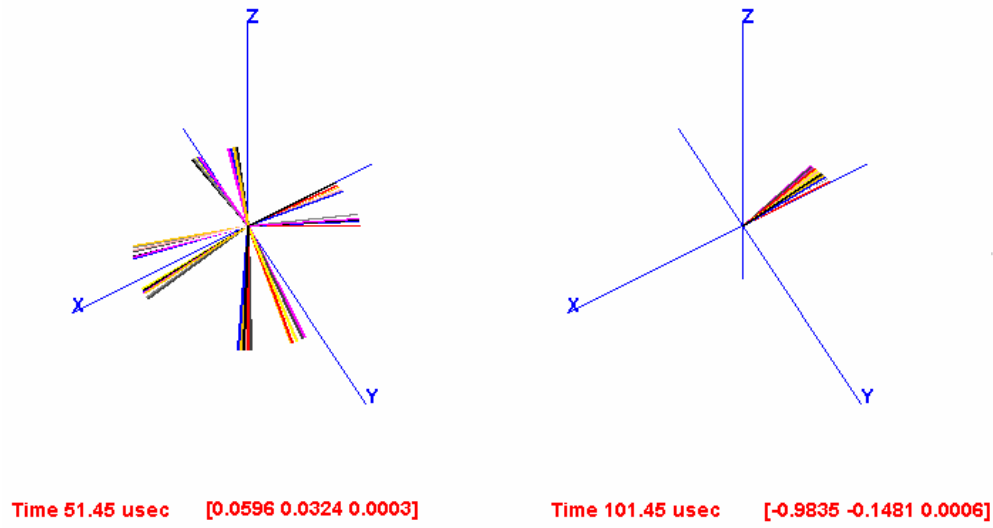

Figure 5.13 –Phase dispersion and recovery of magnetization vector during gradient echo: the left display shows the dispersion of magnetization vectors at 50 μsec and the right display shows the recovery of magnetization vectors at time instance 100 μsec.

The left side of Figure 5.13 shows the magnetization vectors after the positive gradient is applied for 0.050 msec. As seen in the figure, the precession frequencies of 64 magnetization vectors are grouped into 8 frequencies since the gradient field is applied only in x direction. If Gy gradient is applied together with Gx gradient, then 64 different frequencies are observed. The right display shows the phase recovery of magnetization vectors after the negative gradient is applied for 0.050 msec. At this point the net phase dispersion created by the gradient fields is zero. The small phase dispersion between magnetization vectors at this instant is caused by B_0 inhomogeneity.

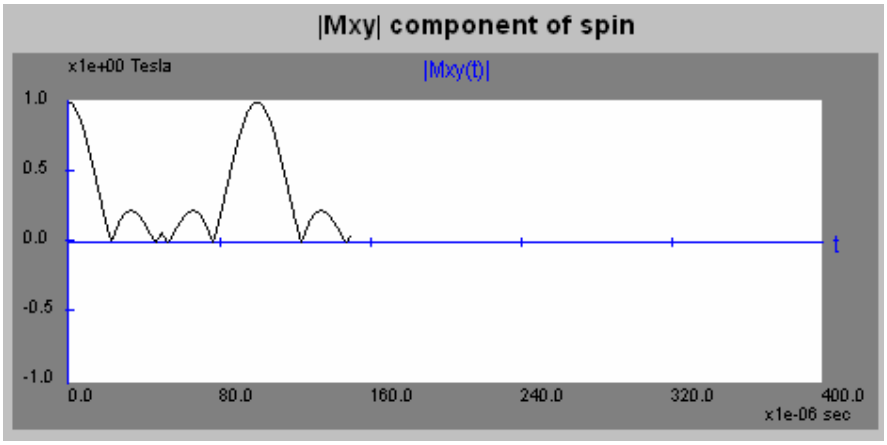

Figure 5.14 – Gradient echo simulation: a relaxation of 50 μsec is applied with Gx=0.1 Tesla/m and followed by a relaxation of 50 μsec with Gx=-0.1 Tesla/m.

As seen in Figure 5.14, the first positive gradient (Gx=0.1) applied vanishes the net transverse magnetization M_{xy} in a very short duration (0.050 msec). The negative gradient (Gx=-0.1) applied for the same duration recovers the phase dispersion and an echo is obtained with a peak value at the 100th μsec.

An interesting behavior for the gradient echo is observed when the B_0 inhomogeneity has a linearly changing distribution instead of a random distribution. The system is excited with a rectangular shaped RF pulse with frequency $f_{rf} = 42.57$ MHz, amplitude = 8.1 mTesla, and duration = 1.45 μsec which corresponds a $90°$ pulse. The B_0 inhomogeneity value is set to 0.1 μTesla. After the $90°$ pulse, the system is put into relaxation for 2 msec sampled at 0.01 msec with the gradient fields off. Then the system is relaxed for 0.050 msec with a sampling time of 0.001 msec and Gx gradient field set to 0.1. After this, the Gx gradient value is set to -0.1 Tesla/m and the system is put into relaxation for a duration of 0.100 msec sampled at an interval 0.001 msec. The transverse magnetization $M_{xy}(t)$ during this sequence is shown in Figure 5.15.

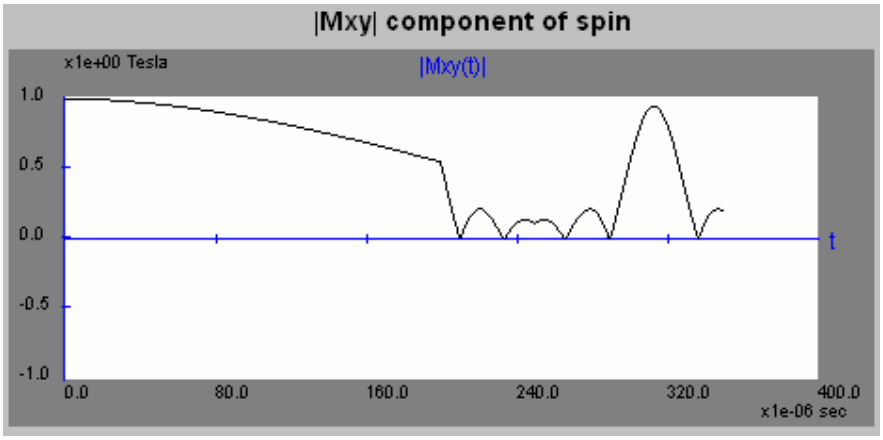

Figure 5.15 – Gradient echoes following T_2: the figure shows a gradient echo after a relaxation of 1 msec. The gradient echo is obtained with a relaxation of 0.050 msec with Gx=0.1 Tesla/m followed by a relaxation of same duration with Gx=-0.1. Bo inhomogenity has a linear changing distribution. The peak of the echo follows T_2 decay instead of T_2^* decay.

As seen in Figure 5.15, the first relaxation with gradient fields off is a normal T_2^* decay because of the B_0 inhomogeneity. After this, the positive gradient applied causes the net magnetization to diminish as expected. The negative gradient following the positive one recovers the phase coherence and generates the echo which is expected to follow T_2^* decay. But it is seen that the peak of the echo follows T_2 decay instead of T_2^* decay. It is understood that the negative gradient field also recovers the phase dispersion caused by B_0 inhomogeneity. The reason for this is found out to be the linear change of B_0 field among magnetization vectors. Since the precession frequencies of magnetization vectors has a linear change, a gradient field re-phases the spins since the gradient field also creates a linear change in precession frequencies of magnetization vectors. The linearly changing B_0 field creates a smooth T_2^* decay but it has the drawback of being neutralized by a gradient field applied in the opposite direction of B_0 field change. This disables the peaks of gradient echoes to follow a T_2^* decay.

To overcome this problem, the B_0 magnetic field inhomogeneity should be modeled with a random distribution. If the same experiment is carried out with a random B_0 inhomogeneity distribution, the following result in Figure 5.16 is obtained. The parameters and the sequence are same with the previous experiment except that the positive gradient followed by a negative gradient is continued with another positive gradient of 0.100 msec to obtain a second gradient echo.

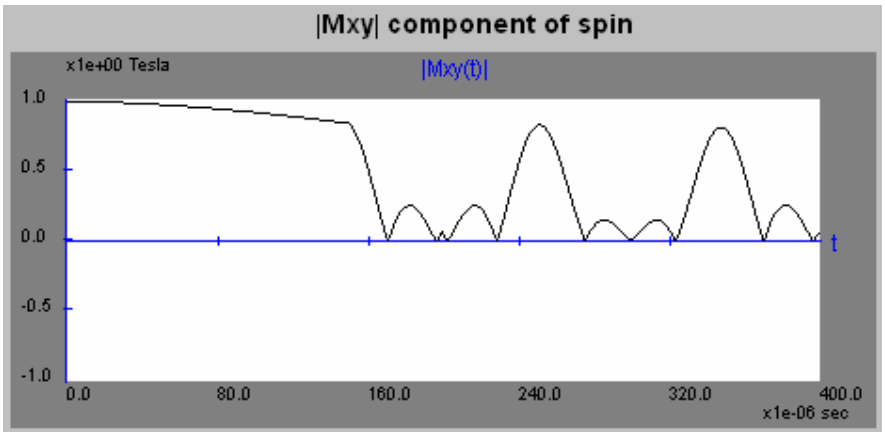

Figure 5.16 – Gradient echoes following T_2^*: the figure shows two gradient echoes after a relaxation of 1.5 msec. The gradient echoes are obtained with a relaxation of 0.050 msec with Gx=0.1 Tesla/m followed by a relaxation of 0.100 msec duration with Gx=-0.1 Tesla/m and another relaxation of 0.100 msec with Gx=0.1 Tesla/m. B_0 inhomogenity has a random changing distribution and the inhomogeneity constant is set to 0.1. The peak of the echo follows T_2^* decay as expected.

As seen in Figure 5.16, the peak values of the two echoes follow T_2^* decay as expected. These results show that a random B_0 inhomogeneity distribution should be used in order to simulate gradient echoes correctly.

5.2 Results from MRI Simulator

The experiments presented here are the simulation results of predefined 3D virtual phantoms with different pulse sequences and varying parameters. There four virtual objects in the application selectable by the user. The table below lists the specifications of each object.

OBJECTS			Proton Density	T1 (msec)	T2 (msec)
Object 1 – 8 x 8 x 1	Slice 1	Tissue-1 (Background)	60	500	40
		Tissue-2	90	50	80
Object 2 – 8 x 8 x 4	Slice 1	Tissue-1 (Background)	20	200	40
		Tissue-2	100	200	40
	Slice 2	Tissue-1 (Background)	20	200	40
		Tissue-2	100	200	40
	Slice 3	Tissue-1 (Background)	20	200	40
		Tissue-2	100	200	40
	Slice 4	Tissue-1 (Background)	20	200	40
		Tissue-2	100	200	40
Object 3 – 16 x 16 x 2	Slice 1	Tissue-1 (Background)	20	200	40
		Tissue 2	100	200	40

	Slice 2	Tissue-1 (Background)	20	200	40
		Tissue-2	100	200	40
Object 4 – 32 x 32 x 1	Slice 1	Tissue-1 (Background)	20	200	40
		Tissue-2	100	200	40
		Tissue-3	100	200	40
		Tissue-4	20	200	40
		Tissue-5	100	200	40

Figure 5.17 –Input objects specification

The table in Figure 5.17 lists the proton density, T_1 and, T_2 values of the tissues in the slices of the input objects. The dimension information of each is object is specified with their titles. The first tissue of each slice corresponds to the background tissue. These objects are displayed on the "MRI Simulator" application slice by slice using a 256 level gray scale color scheme.

Each of these four objects are created for different purposes. The first object is a 8x8x1 object which has a non-uniform T_1 and T_2 map. This single slice object allows the user to obtain T_1 weighted and T_2 weighted images by adjusting the pulse sequence parameters appropriately. All parameter (proton density, T_1, T_2) maps of this object has a higher square shaped density over a uniform background. So the k-space data of the output images should a sinc function which is the Fourier transform of rectangular window.

The second object is an 8x8x4 with all slices having a uniform T_1 and T_2 map. This object can be used to test the performance of the slice selection pulse. All the slices of this object have square shaped higher density regions and these regions are placed at different positions among slices so that contributions from other slices at the output image can easily be noticed.

The third object is a 16x16x2 object. The first slice of this object is designed to observe chemical-shift affect. For this purpose, the shielding factor of the voxels of the high proton density tissue in the first slice, is set to a value of 0.999990 which corresponds to 10 parts per million. If this slice is imaged with any pulse sequence type, the affect of chemical shift should be seen as a shift at the output image.

The fourth object is a simplified version of Shepp-Logan phantom. It is a 32x32x1 object with different elliptic tissues.

5.2.1 Slice Selection

The second phantom which has four slices is used to perform slice selection pulse with different parameters. In the simulation, sinc function is used as the envelope function of the excitation pulse. The sinc function should be applied infinitely to create a perfect slice selection profile. Since this is not realizable, the sinc pulse should be packed with as many side lobes as possible to create a nice slice selection profile. So the RF duration selection is critical for slice selection performance. The B_0 inhomogeneity value is set to 0.0 for slice selection tests to get rid of imperfections resulting from B_0 inhomogeneity and to focus only on slice selection. The first experiment aims to select the first slice of the virtual object and is performed with the following parameters:

Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla / m, Slice Center = 0.00 m,

Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.

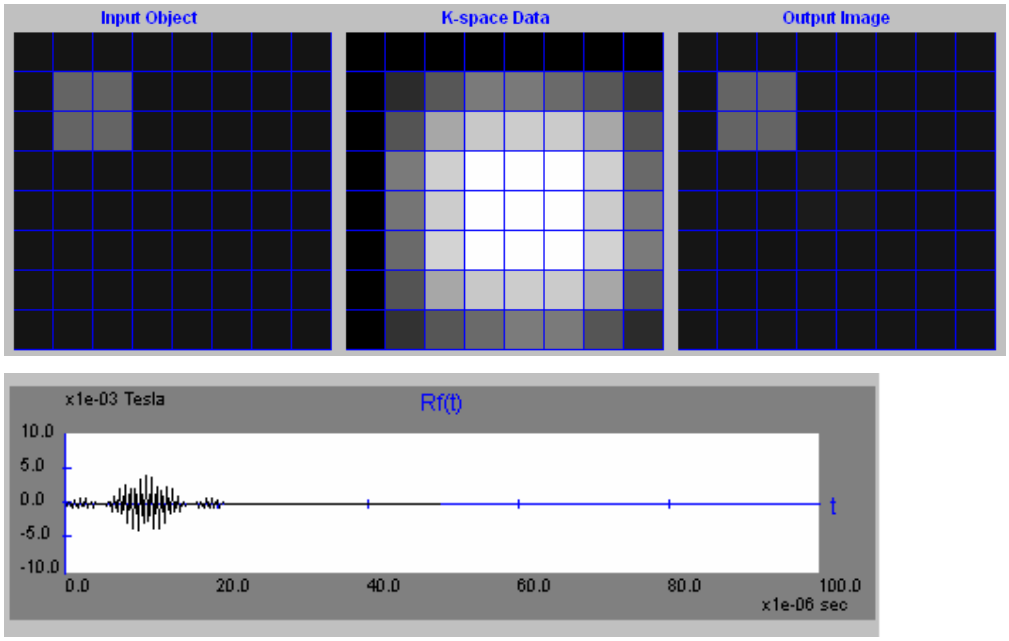

Figure 5.18 – Slice selection, one slice: The top display shows the gradient echo result for the second phantom. The selected slice of the input object, the k-space collected and the output object are shown from left to right. The bottom display shows the RF pulse used in the slice selection. Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The Bo inhomogeneity value is set to 0.

As seen in Figure 5.18, the output image is very close to the first slice as desired but the contribution from the second slice is noticeable. Figure 5.18 also shows the RF pulse used in the slice selection. As seen on the diagram, the sinc function has only one side lobe on each side. If the RF duration is made shorter, the selectivity of the RF pulse decreases and contributions from other slices increase.

The center of the slice and the width of the slice determine which region of the 3D object to be excited. If the previous simulation is repeated by only changing the slice center to 0.005 m which is the middle of the first and the second slice, the result in Figure 5.19 is obtained. The whole configuration parameters for this simulation are as follows:

Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.005 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.

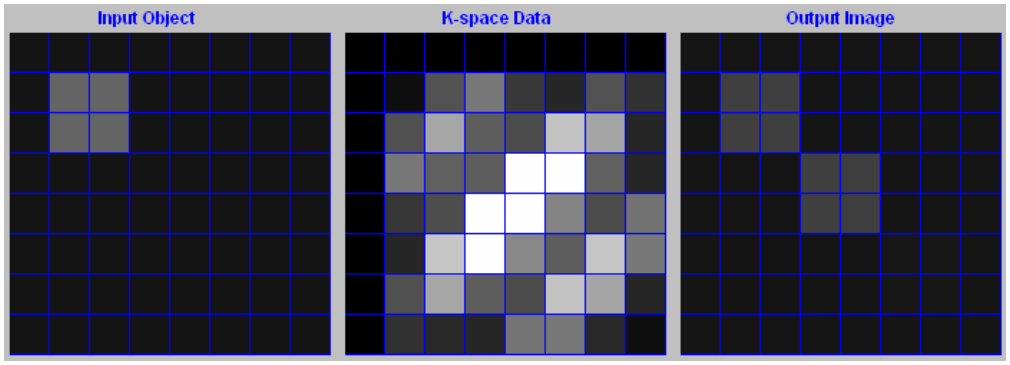

Figure 5.19 – Slice selection, middle of two slices is selected: the display shows the gradient echo result for the second phantom, the middle of the first two slices is selected as the slice center. Data from both slices exist at the output image. Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m/m, Slice Center = 0.005 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.

As seen in Figure 5.19, the output image is the addition of the first two slices. The excitation pulse designed to select the middle of the two slices excited both the slices the same amount, and the output image is formed by equal contribution of both slices.

If a hard pulse is applied to the object with the following configuration, the output image in Figure 5.20 is obtained. To apply a hard pulse to the system, the z gradient Gz is set to 0. This makes the bandwidth of the sinc pulse 0 which converts the sinc to pulse to a rectangular function.

Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.0 Tesla / m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 4 μsec}. The B_0 inhomogeneity is set to 0.

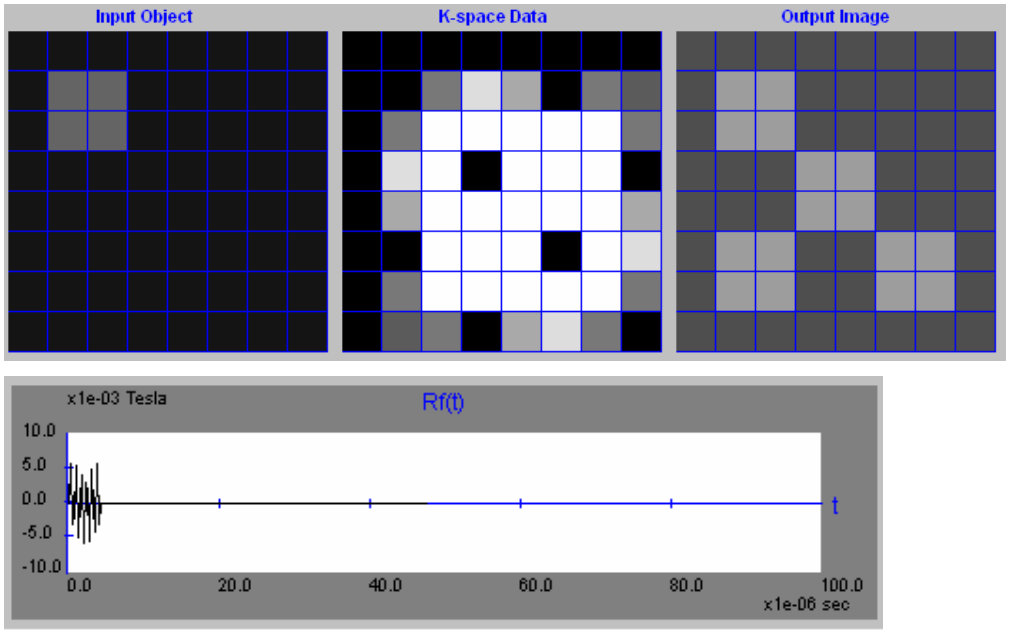

Figure 5.20 – Slice selection with a hard pulse: The top display shows the gradient echo result for the second phantom. The selected slice of the input object, the k-space collected and the output object are shown from left to right. The bottom display shows the hard RF pulse used in the slice selection. Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.0 Tesla / m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 4 μsec}. The B_0 inhomogeneity value is set to 0.

As seen in Figure 5.20, the output image is the addition of four slices. The background intensity of the output image is 4 times the background of one slice

as expected. The RF pulse is also shown in Figure 5.20 which is an unmodulated sinusoidal.

5.2.2 Spin Echo

Spin echo is a widely used pulse sequence type because of its ability to generate T_2 weighted images. The first result presented here is obtained for the first phantom which has a single slice. The pulse parameters are adjusted for a proton density image by selecting TE a low value and TR a high value. The parameters used for this experiment are as follows:

Pulse sequence parameters: {Spin Echo, TE = 0.50 msec, TR = 2500 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla / m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.1.

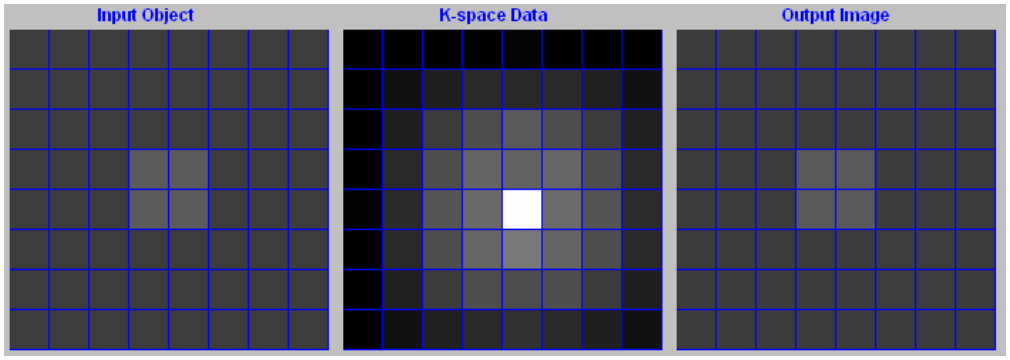

Figure 5.21 – Spin echo result for the first phantom, proton density weighted: Pulse sequence parameters: {Spin Echo, TE = 0.50 msec, TR = 2500 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla / m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The Bo inhomogeneity value is set to 0.1 .

As seen in Figure 5.21, the k-space data is sinc function as expected and the output image is the same as input image with a small intensity difference. The

peak of the k-space data is at the center as expected, which equals the sum of proton densities of voxels in the object.

To obtain T_2 weighted images, TR should be chosen long and TE value should be chosen carefully to differentiate between tissues. The first object has a T_2 map with two tissues whose values are 40 msec and 120 msec respectively. To weight the output image with T_2 values of these two tissues, a TE value which will suppress the signal coming from one of the tissues should be selected. TE is selected 100 msec for this purpose and the following simulation is performed with the configuration below.

Pulse sequence parameters: {Spin Echo, TE = 100 msec, TR = 2500 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.1.

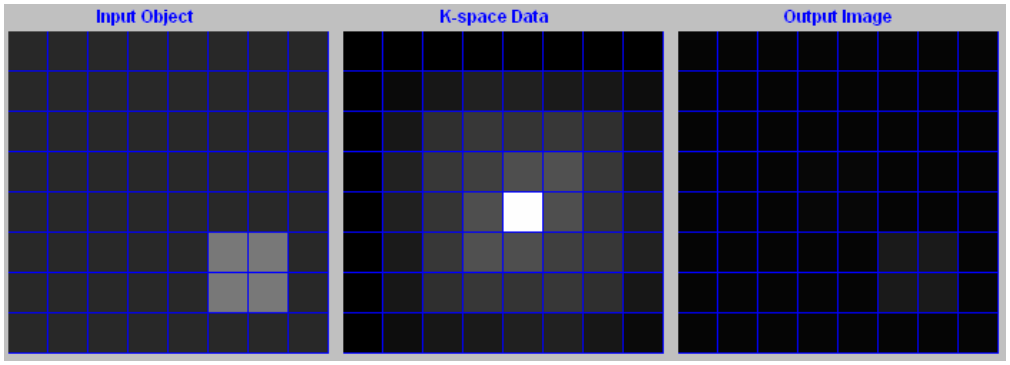

Figure 5.22 – Spin echo result for the first phantom, T_2 weighted: Pulse sequence parameters: {Spin Echo, TE = 100 msec, TR = 2500 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.1.

As seen in the figure above, the output image is T_2 weighted instead of proton density. Since the T_2 image of the object has a rectangular shape, the k-space data

is a sinc function. Another thing that should be remarked is that the intensity of the T_2 weighted image is the weighted image of the proton density map. There is a big intensity difference between the input image and the output image in the figure above. Because the input object display shows the T_2 map which is displayed according to the T_2 values of the image. On the other hand, the intensity of the output image is a T_2 weighted version of the proton density image.

The next simulation presents a spin echo result for the second phantom which is a multi-slice object. An unexpected artifact is observed in spin echo simulations for multi-slice objects. The configuration parameters for the simulation are as shown below:

Pulse sequence parameters: {Spin Echo, TE = 100 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.01 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.1.

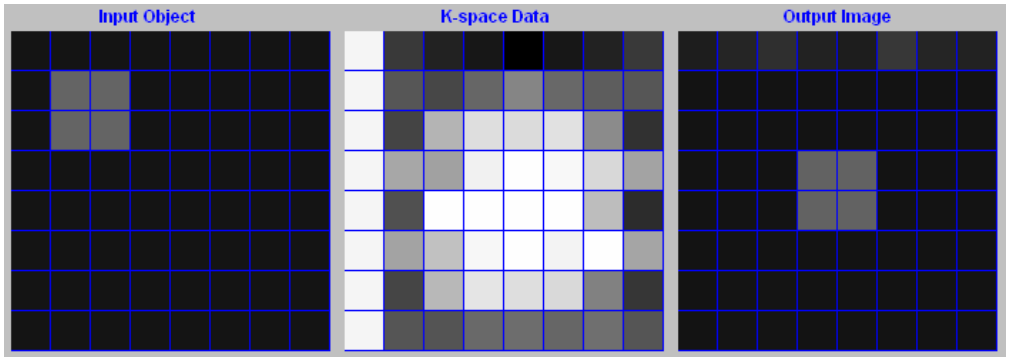

Figure 5.23 – Spin echo result for the second phantom, second slice selected: An unexpected bright line is observed at the most left column of k-space which causes a distortion at the top row of output image. Pulse sequence parameters: { Spin Echo, TE = 100 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.01 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.1.

As seen in Figure 5.23, the most left column of k-space is filled with a high value unexpectedly. The rest of the k-space data is like a sinc function as expected . This high frequency line in k-space creates a varying line at the first row of the output image. This unexpected result is not seen for single slice objects or for gradient simulations of multi-slice objects. By checking the magnetization values of all the voxels in the virtual object during simulation, it is found out that this artifact is caused by the neighbor slice. In spin echo, one $90°$ pulse and one $180°$ pulse is applied to the virtual object for each phase encoding step. Because of the imperfection of slice selection pulse, the $90°$ pulse creates a small amount of transverse magnetization in the neighbor slice, the $180°$ pulse increases this transverse magnetization and creates this artifact.

5.2.3 Gradient Echo

Gradient echo has the ability to generate T_2* weighted images. The first result presented here is obtained for the second phantom with B_0 inhomogeneity value set to 0. In this case, the magnetization vectors will decay with T_2 and gradient echo is expected to produce the same result with spin echo. The pulse sequence parameters are adjusted for proton density weighting, the whole configuration for this experiment is as follows:

Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.02 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.

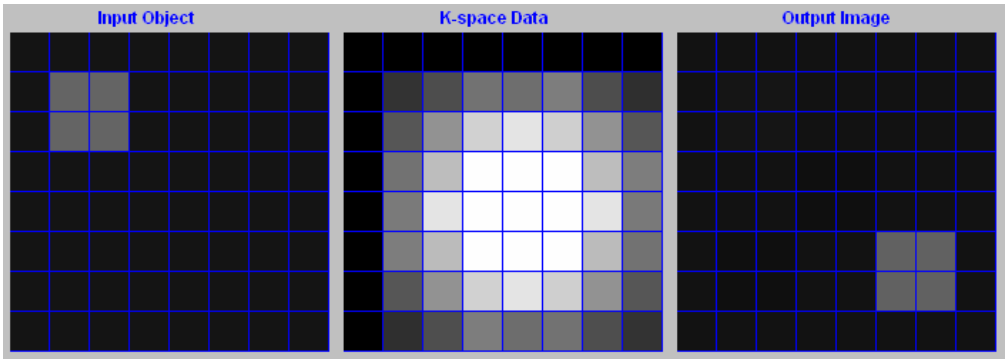

Figure 5.24 – Gradient echo result for the second phantom, third slice selected: Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla / m, Slice Center = 0.02 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The Bo inhomogeneity value is set to 0. The input object display shows the first slice of the object.

As seen in the figure above, the output image generated by the simulation is perfectly equal to the selected slice.

The next simulation presents a gradient echo result for the third phantom with an inhomogeneous B_0. The configuration for this experiment is as follows:

Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla / m, Slice Center = 0.02 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.1.

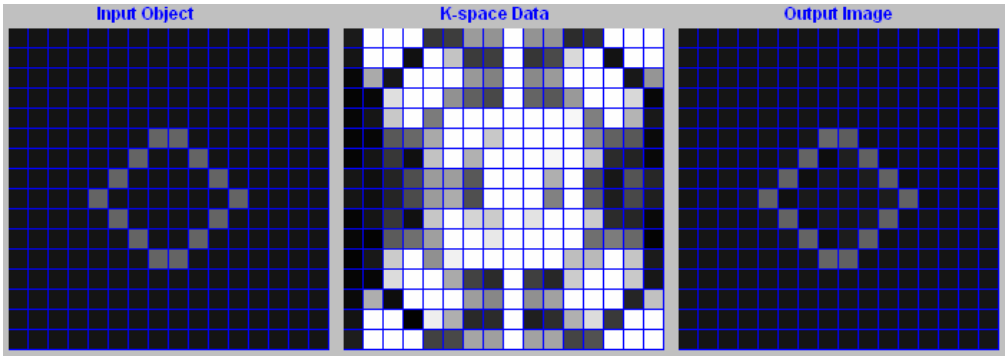

Figure 5.25 – Gradient echo result for the third phantom, first slice selected: Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.02 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The Bo inhomogeneity value is set to 0.

The Figure 5.25 shows the k-space data and output image for this simulation. The output image reflects the selected slice and a small contribution from the second slice.

5.2.4 Echo Planar

Echo planar pulse sequence is a fast MR imaging method. Echo planar imaging may have different k-space trajectories. The one implemented in this thesis uses a rectilinear k-space trajectory. Since there is only one excitation in echo planar imaging, it is the also the fastest pulse sequence in the simulation. An echo planar experiment is performed for the fourth object with the following configuration.

Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.

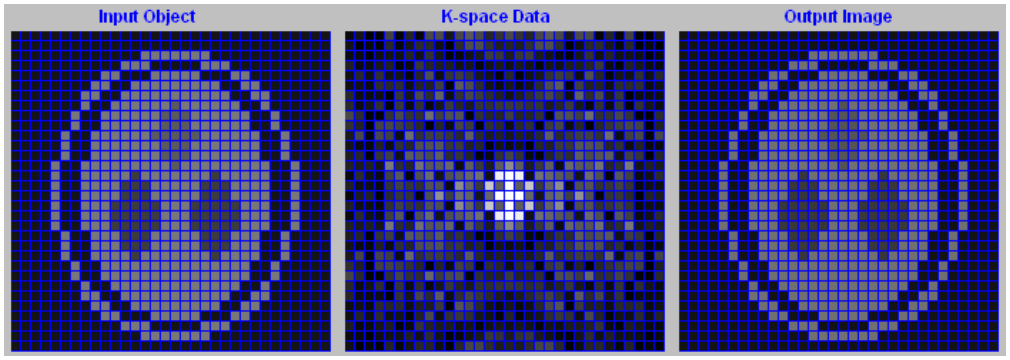

Figure 5.26 – Echo planar result for the fourth phantom with a homogenous B_0: Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The Bo inhomogeneity value is set to 0.

As seen in Figure 5.26, the output image is the same with input image as expected. The same experiment is repeated by setting B_0 inhomogeneity to 0.02 μTesla. The output of the simulator in this case is shown in Figure 5.27.

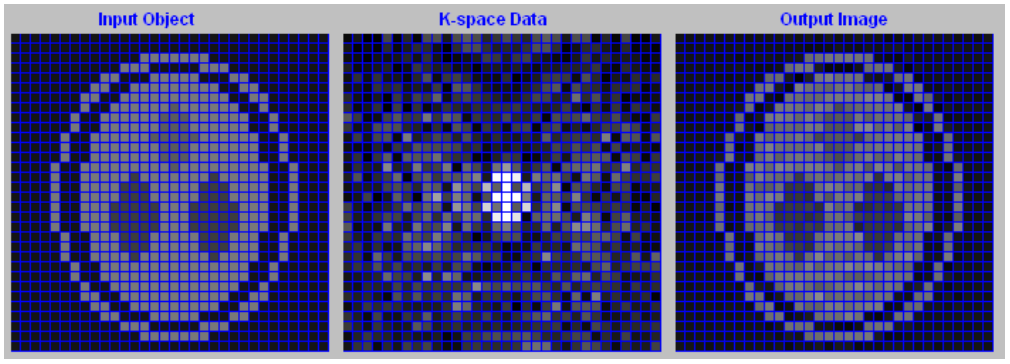

Figure 5.27 – Echo planar result for the fourth phantom with B_0 inhomogeneity equal to 0.02: Pulse sequence parameters: {Gradient Echo, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}.

As seen in the figure above, including B_0 inhomogeneity in the simulation generated degradation in the image compared to homogenous B_0 case. The degradation in the image increases further with increasing inhomogeneity constant.

5.2.5 MRI Artifacts

The simulation program has the ability to generate many of the artifacts in MRI. These artifacts can be counted as B_0 inhomogeneity, slice selection imperfection, Gibbs ringing artifact, aliasing and chemical –shift affect. Some of these artifacts are shown in the previous results.

A common artifact in MRI is aliasing which occurs when the k-space sampling parameters are not selected according to the Nyquist theorem. The result shown in Figure 5.28 presents such a case. It is obtained with the following configuration of the application.

Pulse sequence parameters: {Echo Planar, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla / m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0. Frequency encoding gradient Gx is set to 0.1 Tesla / m.

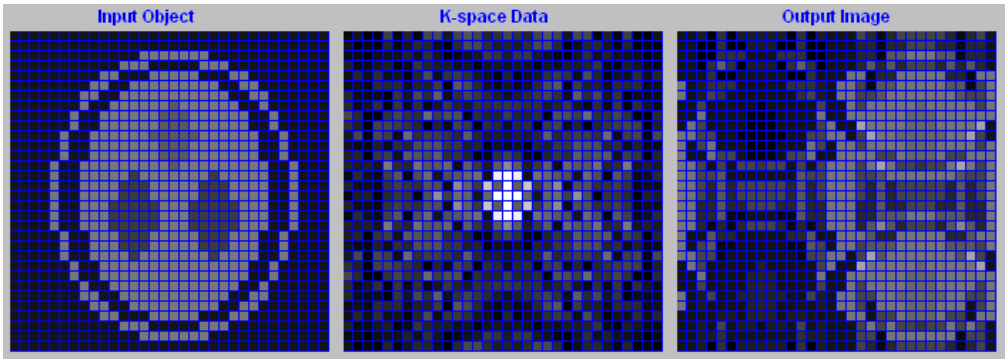

Figure 5.28 – Echo planar result for the fourth phantom with aliasing: Pulse sequence parameters: {Echo Planar, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}, Gx = 0.1 Tesla/m, the Bo inhomogeneity value is set to 0.

As seen in Figure 5.28, the output image is the addition of shifted, inverted portions of input image.

Another artifact in MRI arises from the change in resonance frequency of the spin due to the chemical environment around the nucleus called chemical-shift affect. To observe the chemical-shift affect, a simulation is performed with the configuration below for the third object which is specially prepared for this experiment.

Pulse sequence parameters: {Echo Planar, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The B_0 inhomogeneity value is set to 0.

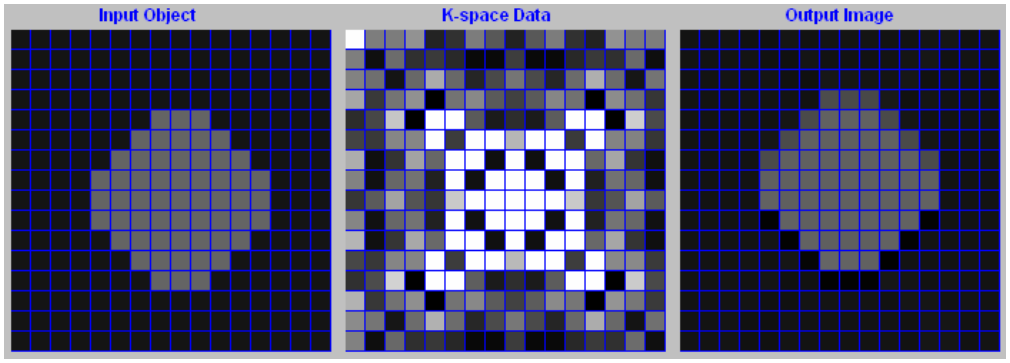

Figure 5.29 – Echo planar result for the third phantom, chemical-shift affect: Pulse sequence parameters: {Echo Planar, TE = 0.50 msec, TR = 1000 msec, Flip Angle = $90°$}, Slice selection parameters: {Gz = 0.4 Tesla / m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf $\mathrm{Duration} = 21$ μsec}. The Bo inhomogeneity is set to 0.

As seen in Figure 5.29, the output image is a 1 pixel upwards shifted version of the input image.

The Gibbs ringing artifact occurs due to the finite sampling in k-space. It appears as a ringing artifact around sharp changing borders. In the simulation, a noticeable ringing affect could not be observed. To avoid this artifact, windowing of k-space data is performed at the expense of spatial resolution loss. The "MRI Simulator" application allows two types of windowing: rectangular and hamming. The next simulation result shows how the output image is affected if Hamming window is selected.

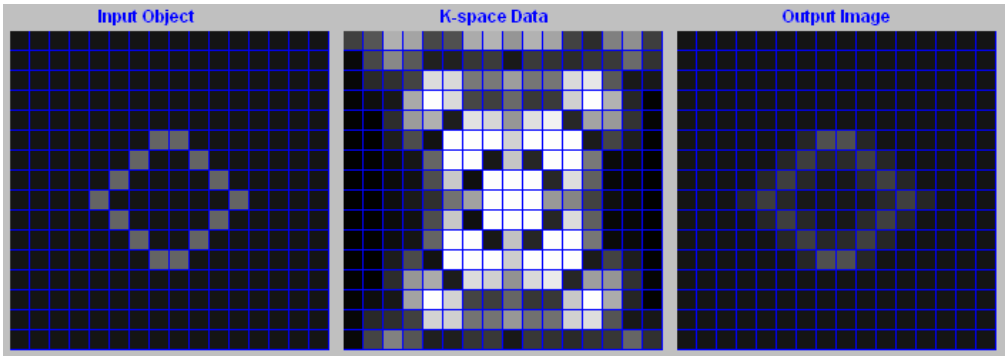

Figure 5.30 – Echo planar result for the fourth phantom, Hamming window applied: The output image is a blurred version of the input image. Pulse sequence parameters: {Echo Planar, TE = 0.50 msec, TR = 1000 msec, Flip Angle = 90}, Slice selection parameters: {Gz = 0.4 Tesla/m, Slice Center = 0.00 m, Slice Width = 0.01 m, Rf Duration = 21 μsec}. The Bo inhomogeneity value is set to 0.0 μTesla.

As seen in the figure above, the Hamming windowing operation caused resolution loss and blurring at the output image .

CHAPTER 6

CONCLUSION

6.1 Summary of the Thesis

In this thesis, a realistic and web accessible MRI simulator based on the solution of Bloch equation is proposed and implemented. The simulator generates images from virtually defined 3D objects using the pulse sequence and slice selection parameters provided by the user. The simulation is implemented in Java and the application is developed as a Java applet which is accessible through a web page. The simulation software is composed of two applications: "Spin Simulator" and "MRI Simulator". The "Spin Simulator" allows one to observe NMR phenomenon on 3D and 2D displays by applying RF pulse and relaxation to a virtual object and is developed as an educational and supportive tool for "MRI Simulator". Gradient echo and spin echo and echo planar pulse sequences are implemented. The "MRI Simulator" generates slice images by executing the pulse sequence with the selected parameters on the 3D virtual object. The applications are designed with an interactive and educative user interface.

6.2 Discussions and Future Work

The simulation kernel is based on the solution of Bloch equation. Bloch equation is solved by numerical methods during excitation. This enables to see the effect of an arbitrary envelope function on a spin system. If the analytic solution were used instead of numerical methods, the output signal transmitted by the virtual object

would be the summation of magnetization vectors only in the selected slice. The numerical solution is executed on the entire volume of the 3D object and contributions from other slices are also taken into account so that a more realistic signal is received form the object. During relaxation, the analytic solution of Bloch equation is used to calculate the magnetization values. Solving the Bloch equation by numerical methods during excitation and using the analytical solution during relaxation provides a realistic and efficient way of implementing the simulation kernel.

The 3D virtual object definition is structured in a rich content and is flexible to represent a real object by its all features. The structure allows one voxel to be composed of multiple magnetization vectors which may produce more realistic magnetization data at the cost of increased simulation time. The structure also allows more than one voxel to be placed at the same spatial position. This enables chemical shift affect to be observed during simulation.

In this thesis, two models for static magnetic field inhomogeneity are investigated. These are a random distributed noise model and a linear changing model. The effectiveness and problems of both models are observed. Including the static magnetic field inhomogeneity in the simulation, causes the individual magnetization vectors to have different precession frequencies during relaxation which enables the T_2^* affect to be simulated successfully. Since the magnetization vectors have position information and are distributed evenly on the entire volume of the object, the gradient fields change the precession frequencies of the individual magnetization vectors. This enables the T_2^{**} affect to be observed. Simulating T_2^* and T_2^{**} affects successfully is essential for successful implementation of gradient echo and spin echo pulse sequences. As shown in the results section, spin echoes and gradient can be generated as in the physical case. Since the 3D virtual object is a discrete representation of a real object, the T_2^* and T_2^{**} decay curves are not exponentials as in the real case. This is due the representation of millions of spins with a couple of thousand spins at most. If

higher number of magnetization vectors are used to represent the virtual object, more realistic T_2^* and T_2^{**} decays are obtained. Comparing both static magnetic field inhomogeneity models, the linear changing produces a smoother T_2^* decay with a less number of magnetization vectors but has the problem of being recoverable by gradient fields. On the other hand, the noise model is more realistic but more magnetization vectors should be used to obtain smooth T_2^* decay which increases computation time.

The simulation does not take into account the magnetic susceptibility variations among tissues which is one of the reasons behind T_2^* decay. In future, this affect can also be included in the simulation by changing the magnetic susceptibility of the spins on the borders of the tissues.

Developing the application in a platform independent and object oriented language like Java, shortened the development time and created a modular and expandable application. The simulation kernel is totally separable from the user interface of the application and is designed like a framework for MRI simulation applications. In MRI, there are many types of pulse sequences. In this thesis, the gradient echo, spin echo and echo planar pulse sequences are implemented. But it is very easy to implement any type of pulse sequence by use of the object oriented structure mentioned in section 4.1.

The simulation of a spin echo sequence for a 8x8x3 object takes 1 minute on an Intel Core Duo CPU at 1.73 GHZ and 2 GB RAM computer. The high simulation time is due to the small step size of the Runge-Kutta method used in the solution of Bloch equation. To decrease the simulation time, an adaptive step size algorithm can be embedded into the current implementation which can decrease the number of steps in a run. But if bigger objects (like 256x256x3) are used in the simulation, the simulations will still take long durations. To avoid this, parallel processing can be applied by distributing the kernel of the simulation to many nodes.

The simulator allows three types of pulse sequences to be selected. To give the user the flexibility to try any type of pulse sequence, the user interface can be designed to allow the user to create different pulse sequences. This can be done by defining excitation and relaxation events and putting them in a sequential order. The same idea can be applied to virtual object definitions. The simulation uses pre-defined virtual objects, in the future this can be replaced by a user interface that allows the user to set the proton density and spin parameters of 3D virtual object.

An MRI simulation can be extended to include the simulation of several actors in MR imaging process. As part of this study, the NMR phenomena and the imaging process of MR systems are simulated in a realistic way. The following items which are not simulated as part of this study, can also be included in an MRI simulation to produce more realistic simulations.

- RF magnetic field inhomogeneity
- Gradient fields with rising and falling edges
- Magnetic susceptibility variation on the borders of tissues
- Receiver coil inhomogeneity

As a conclusion, an easily accessible and helpful tool for MRI related people has been developed as a result of this study. It has the simulation capabilities of previous MRI simulators in an effective way and adds the advantage of being a web based and educative tool.

REFERENCES

[1] Zhi-Pei Liang, Paul C. Lauterbur, "Principles Of Magnetic Resonance Imaging / A Signal Processing Perspective" IEEE Press, New York

[2] The Basics of MRI, http://www.cis.rit.edu/htbooks/mri , visited on 31 July 2008

[3] D.A. Yoder, E. Changchien, C.B. Paschal, J.M. Fitzpatrick, "MRI simulator with static field inhomogeneity," SPIE Medical Imaging, 4684, 592–603, 2002.

[4] H. Benoit-Cattin, G. Collewet, B. Belaroussi, H. Saint-Jalmes, C. Odet, "The SIMRI Project: a versatile and interactive MRI Simulator," Journal of Magnetic Resonance, 173, 97-115, 2005.

[5] S. Kwan, A.C. Evans, G.B. Pike, An extensible MRI simulator for post-processing evaluation, in: International Conference on Visualization in Biomedical Computing, VBC□96, 1996, pp. 135–140.

[6] A.R. Brenner, J. Kursch, T.G. Noll, Distributed large-scale simulation of magnetic resonance imaging, Magnetic Resonance Materials in Biology, Physics, and Medicine 5 (1997) 129–138.

[7] G. Torheim, P.A. Rinck, R.A. Jones, J. Kvaerness, A simulator for teaching MR image contrast behavior, MAGMA 2 (1994) 515–522.

[8] T. Hacklaender, C. Schalla, A. Truemper, H. Mertens, J. Hiltner, B.M. Cramer, A virtual MR scanner for education, RSNA 2002, 88th Scientific Assembly and Annual Meeting, Radiology (2002) 757–225.