

VARIABLE RATE SELECTIVE EXCITATION
RF PULSE IN MRI

VARIABLE RATE SELECTIVE EXCITATION RF PULSE
IN MRI

Optimizing RF Pulses in MRI via Optimal Control

By

STEPHEN JAMES STOYAN, hB.Sc., B.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Science

McMaster University

©Copyright by Stephen James Stoyan, August 2004

MASTER OF SCIENCE (2004)
(Mathematics and Statistics)

McMaster University
Hamilton, Ontario

TITLE: Variable Rate Selective Excitation RF Pulse in MRI

AUTHOR: Stephen James Stoyan, hB.Sc., B.Sc.

SUPERVISORS: Dr. Christopher Kumar Anand and Dr. Tamás Terlaky

NUMBER OF PAGES: xvi, 121

Abstract

Magnetic Resonance Imaging (MRI) is an advanced tomographic technique that is able to produce high resolution cross-sectional images of an object or specimen by exploiting Radio Frequency (RF) pulses. A Variable Rate Selective Excitation (VERSE) pulse is a type of RF pulse that reduces the Specific Absorption Rate (SAR) of molecules in a specimen while preserving its duration and slice profile. SAR was designed to be minimized by the VERSE pulse as it leads to an increase in specimen temperature during MRI procedures.

The nonlinear VERSE model was sequentially transformed into an optimal control problem that was efficiently solved by Sparse Optimal Control Software (SOCS). The Magnetic Resonance (MR) signal produced by numerical simulations were then tested and analyzed by an MRI simulator. The VERSE model produced intriguing results and generated high-quality MR signals. The research and testing results produced by the VERSE pulse may influence further research in the area and have built an excellent foundation for more development of this RF pulse.

Acknowledgements

I would first like to thank my supervisors, Dr. Christopher Kumar Anand and Dr. Tamás Terlaky, for their guidance, inspiration and wisdom during the preparation of this thesis. I am grateful to Dr. Agnes Tourin, for her careful reading of my thesis and helpful discussions. Also, I appreciate the great aid and support from all the members of the Advanced Optimization Laboratory.

Finally, I am indebted to thank my family for their continuous support, encouragement and understanding. Special thanks go to my parents, my two brothers, and my grandfathers, Boris Andernov and Thomas Stoyan.

Contents

Abstract	iii
Acknowledgements	v
Notations	ix
Abbreviations	xi
List of Figures	xiii
Preface	1
1 Preliminaries	3
1.1 Tomography	3
1.2 MRI Basics	6
1.3 MRI Pulse Concepts	9
2 Variable Rate Selective Excitation (VERSE)	13
2.1 Background	13
2.2 The VERSE Model	17
2.3 Discretization	25
2.4 VERSE Penalty	27
2.5 Optimal Control Problem	29
3 Nonlinear Optimization	39
3.1 Unconstrained Optimization	39
3.1.1 Line Search Method	40
3.1.2 Trust Region Method	43
3.2 Equality-Constrained Optimization	47
3.3 Inequality-Constrained Optimization	50
3.3.1 Quadratic Optimization	51

Merit Function	52
Hessian Approximation	56
3.3.2 Sequential Quadratic Programming	57
The SQP Algorithm	59
Algorithm's Strategies	61
4 Implementation	65
4.1 SQP Implementation	65
4.2 Slice Assignment	68
4.3 Initial Solution	71
4.4 Sparse Optimal Control Software (SOCS)	76
5 Results	79
5.1 Initializations	79
5.2 Five Slice Results	80
5.3 Fifteen slice Results	81
5.4 Fifteen Slice Penalty Results	89
6 Simulation	97
6.1 Image Reconstruction	97
6.2 Imaging the Signal	98
6.2.1 1D Coverage and Data Collection	99
6.2.2 2D and 3D Coverage	101
6.3 VERSE Simulation	102
7 Conclusions and Future Work	109
Appendix: The MR Signal	115
Bibliography	119

List of Notations

- t : Time
- $\vec{\mu}(t)$: Magnetic moment vector
- $\vec{M}(t)$: Magnetization vector
- $\vec{M}_{\perp}(t)$: Transverse magnetization vector
- $M_x(t)$: Magnetization component in the x -axis direction
- $M_y(t)$: Magnetization component in the y -axis direction
- $M_z(t)$: Magnetization component in the z -axis direction
- M_0 : Initial magnetization value
- $\vec{B}(t)$: External magnetic field vector
- $\vec{B}_{\text{rf}}(t)$: Radio frequency magnetization vector
- $b_x(t)$: External magnetization component in the x -axis direction
- $b_y(t)$: External magnetization component in the y -axis direction
- $b_z(t)$: External magnetization component in the z -axis direction
- $\omega(t)$: Precessional frequency
- γ : Gyromagnetic constant
- τ_1 : Longitudinal magnetization parameter
- τ_2 : Transverse magnetization parameter
- $\vec{G}(t, s)$: Gradient vector
- $G(t)$: Gradient value
- G_{max} : Maximum gradient value
- s : Coordinate position of the magnetization vector
- S : Set of coordinate positions

- S_{in} : Set of coordinate positions in the slice
- S_{out} : Set of coordinate positions out of the slice
- n : Total number of coordinate positions
- N : Total number of time discretizations
- \underline{s} : Lower bound on coordinate positions in S_{in}
- \bar{s} : Upper bound on coordinate positions in S_{in}
- \underline{s}_l : Upper bound on coordinate positions lower than \underline{s} in S_{out}
- \bar{s}_u : Lower bound on coordinate positions greater than \bar{s} in S_{out}
- $W(t)$: Slew rate
- W_{max} : Maximum slew rate
- α : Angle of transverse tip
- ε_1 : S_{in} vector bound
- ε_2 : S_{out} vector bound
- $\Omega(t)$: State variables
- $\Phi(t)$: Control variables
- \mathbf{g} : Gradient matrix
- \mathbf{G} : Jacobian matrix
- \mathbf{H} : Hessian matrix
- \mathbb{R} : Set of Real numbers
- \mathbb{C} : Set of Complex numbers
- \mathbb{Z} : Set of Integers
- $\Upsilon(x, y, z)$: Total received signal
- $v(s)$: Signal received at coordinate position s
- k_x : K-Space x -axis signal position
- k_y : K-Space y -axis signal position
- k_z : K-Space z -axis signal position
- x : x -axis image position
- y : y -axis image position
- z : z -axis image position

List of Abbreviations

MR:	Magnetic Resonance
NMR:	Nuclear Magnetic Resonance
MRI:	Magnetic Resonance Imaging
VERSE:	Variable Rate Selective Excitation
RF:	Radio Frequency
SAR:	Specific Absorption Rate
NLO:	Nonlinear Optimization
NLP:	Nonlinear Programming
DFO:	Derivative Free Optimization
QO:	Quadratic Optimization
QP:	Quadratic Programming
SQP:	Sequential Quadratic Programming
SOCS:	Sparse Optimal Control Software
KKT:	Karush Kuhn Tucker
FT:	Fourier Transform

List of Figures

1.1	Electromagnetic spectrum.	5
1.2	The MR imaging process.	7
1.3	The effect of an RF pulse on an individual magnetic moment.	10
2.1	Magnetic moment vectors pointing in random directions.	14
2.2	Precession of nuclear spin about an external magnetic field in the z -axis direction, similar to the wobbling of a spinning top.	15
2.3	A generic NMR pulse imaging sequence.	18
2.4	Specimen separated into planes or slices about the z -axis.	20
2.5	MRI processor separating partitions of an object into the transverse plane by using different gradient strengths at each coordinate position s_1, s_2, \dots, s_n	25
2.6	Separating magnetization vectors into coordinate positions which are in the slice, S_{in} , and out, S_{out}	26
3.1	Depending on the values of $\tilde{\rho}$, the Trust Region method can take a Gradient step, $-\nabla J(\mathbf{x}^q)$ or Newton step, $-(\nabla^2 J(\mathbf{x}^q))^{-1}\nabla J(\mathbf{x}^q)$	46
4.1	The initial solution for magnetic moment vectors in S_{in} that have tipped into the transverse plane by an angle of α	73
4.2	The initial solution for magnetic moment vectors in S_{out}	74
4.3	The initial solution for the gradient function $G(t)$	74
4.4	The initial solution for the external magnetization $b_y(t)$	75
5.1	The separation of coordinate positions s_i into S_{out} and S_{in} for five magnetization vectors.	81
5.2	From left to right, magnetization vectors corresponding to coordinate positions s_1, s_2, s_3, s_4 and s_5	82
5.3	External magnetization components $b_x(t)$ and $b_y(t)$, shown respectively.	83

5.4	Gradient sequence pertaining to magnetization vectors plotted in Figure 5.2.	83
5.5	The separation of coordinate positions s_i into S_{out} and S_{in} for 15 magnetization vectors.	84
5.6	From left to right, magnetization vectors corresponding to coordinate positions s_1, s_2, s_3, s_4, s_5 and s_6	85
5.7	From left to right, magnetization vectors corresponding to coordinate positions s_7, s_8 and s_9	86
5.8	From left to right, magnetization vectors corresponding to coordinate positions $s_{10}, s_{11}, s_{12}, s_{13}, s_{14}$ and s_{15}	87
5.9	External magnetization components $b_x(t)$ and $b_y(t)$, shown respectively.	88
5.10	Gradient sequence pertaining to magnetization vectors plotted in Figures 5.6 – 5.8.	88
5.11	From left to right, magnetization vectors corresponding to coordinate positions s_1, s_2, s_3, s_4, s_5 and s_6	91
5.12	From left to right, magnetization vectors corresponding to coordinate positions s_7, s_8 and s_9	92
5.13	From left to right, magnetization vectors corresponding to coordinate positions $s_{10}, s_{11}, s_{12}, s_{13}, s_{14}$ and s_{15}	93
5.14	External magnetization components $b_x(t)$ and $b_y(t)$, shown respectively.	94
5.15	Gradient sequence pertaining to magnetization vectors plotted in Figures 5.11 – 5.13.	94
6.1	A schematic drawing of a general MR Imaging sequence.	98
6.2	A varying gradient field in combination with an RF pulse is applied to an object that produces a signal that can be imaged in 1D.	100
6.3	A varying gradient field and accompanying RF pulse producing a signal to be imaged in 1D.	100
6.4	The position of cerebrospinal fluid and gray matter to be imaged by our MRI simulation.	104
6.5	The signal produced by the VERSE pulse MRI simulation over two vertically aligned tissues.	105
6.6	The angular position of cerebrospinal fluid to be imaged by our second MRI simulation.	106
6.7	The signal produced by the VERSE pulse MRI simulation over the diagonal cerebrospinal fluid.	107

6.8 The signal produced when a generic RF pulse and gradient sequence is applied to the diagonal cerebrospinal fluid. . . . 108

Preface

The notion of applying mathematics to industrial, mechanical and other similar physical problems has existed for centuries. Previously, this was typically an area for engineers and physicists, however, recently mathematicians have been indulging in this field. Their knowledge and understanding of the intricate mathematical details behind these physical problems have allowed them to make many interesting discoveries and improvements in industry and science. Magnetic Resonance Imaging (MRI) is one of such revolutionary processes that produces internal images of specimens or objects without using any invasive diagnostic techniques. The impact of this imaging technique on the radiology community has been outstanding, primarily due to the systems ability to create high-quality images and uphold superior safety standards. In this thesis we present an optimized mathematical model that is designed to improve the signal generation stage of MRI. By attacking the problem using nonlinear optimization techniques we intend on upgrading the safety of the process while enhancing the image quality.

This thesis was written for individuals with backgrounds in applied mathematics; hence, the basis of MRI and its principles will be detailed as the intended audience will probably have a weak understanding of this field. For more information, there exist many books on the subject, one can consult any of the various references listed in the bibliography [7], [17], [18]. The mathematics behind the optimization model is covered in great detail. The model is primarily based on generating Magnetic Resonance (MR) signals through the use of Radio Frequency (RF) pulses. This is a very hot topic in MRI as there are many different types of RF pulses, each with specific characteristics. In fact, the characteristics of the RF pulse determine the contrast and resolution of the final image. The RF pulse we design is based on an idea from Conolly *et al.* [9] in 1987 that was never fully developed. We take this theory to the next level by modelling and implementing it in what is known as the Variable Rate Selective RF pulse.

Included in the thesis are seven chapters with three main topics, namely, the VERSE model, Nonlinear Optimization (NLO), and MR image reconstruction. In Chapter 2, the nonlinear VERSE optimization problem is formulated from an RF pulse idea. The NLO is then transformed into an optimal control problem that separates the models dimensions into state and control variables. The Sequential Quadratic Programming (SQP) optimization procedure for solving optimal control problems is detailed in Chapter 3. In addition, various methods of solving constrained and unconstrained optimization problems that progress to the SQP process are discussed. The implementation issues involved in computing the VERSE pulse are described in Chapter 4. The ideas behind the initial solution and the functionality of the optimal control software, SOCS, used for solving the problem is also mentioned. In Chapter 5, the computational results for the VERSE pulse are shown for three different test cases. The results for 5 slices, 15 slices and 15 penalized slices are graphically illustrated and documented. The results are then tested by an MRI simulation in Chapter 6, where they are analyzed and examined with respect to the MR signals they generate. Our results and MRI simulations clearly show that mathematical optimization can have an unprecedented effect on improving RF pulses. We hope that the material is covered at an appropriate level and is useful in influencing future developments in the field.

Chapter 1

Preliminaries

Magnetic Resonance Imaging (MRI) has been given much attention in the past decade as it is a relatively new discipline in the realm of applied sciences. Paul C. Lauterbur and Peter Mansfield were awarded the 2003 Nobel Prize in Physiology or Medicine for their discoveries in MRI that lead to its beginnings in 1973. They proposed to introduce a spatially varying magnetic field to an object, and showed that the different frequency components of the signal could be separated to give spatial information about the object. This key innovation of spatially encoding MR data enabled scientists and engineers to develop what we know as MRI today. In this chapter we will outline the basics of MRI and focus on radio frequency pulses, but first we will discuss how important this tomographic procedure is to the medical community.

1.1 Tomography

Tomography, or visualizing the interior of the human body without surgical intervention, has existed for a little less than half a century and started with the development of X-ray tomography [20]. Tomographic imaging modalities have progressed since then, a partial list includes, CAT (Computed Axial Tomography), PET (Positron Emission Tomography), SPECT (Single Pho-

ton Emission Computed Tomography), MRI, and various acoustic imaging systems such as Ultrasound. MRI, however, is the only tomographic imaging technique that produces images of internal physical and chemical characteristics of an object from externally measured Nuclear Magnetic Resonance (NMR) signals [18]. It is primarily based on the well known NMR phenomenon observed in bulk matter, independently described by Felix Bloch and Edward Purcell in 1946 [7]. Bloch continued extensive studies with the NMR of water, thereby laying the groundwork for later developments leading to MRI. He proposed that the nucleus of atoms behave like small magnets and described this nuclear magnetism in what is now known as the *Bloch equation*. The Bloch equation explains that since an atom's nucleus spins on an imaginary axis and has an electric charge, it possesses a microscopic magnetic field called a magnetic moment. It is through the physical properties of magnetic moments that we are able to extrapolate a signal using spatially varying magnetic fields and create an MR image.

The main thrust and reason MRI has been so well publicized is that it produces high resonance images without using radiation and thus, does not have the associated harmful effects. The lack of ionizing radiation has greatly influenced the medical community and is the reason MRI has presided any type of X-ray imaging. As shown in the electromagnetic spectrum in Figure 1.1, MRI uses Radio Frequency (RF) electromagnetic radiation and magnetic fields, which do not cause ionizing radiation and allow vital areas, like the head, to be imaged. However, some known side effects, such as patient heating caused by high levels of SAR (Specific Absorption Rate), do occur, but, they do not lead to the malignant diseases emitted by radiation. Another reason MRI has received so much attention is due to its *spatial* and *contrast* resolution. Spatial resolution refers to the ability of a device or process to identify small, dense objects such as metal fragments or micro-calcifications.

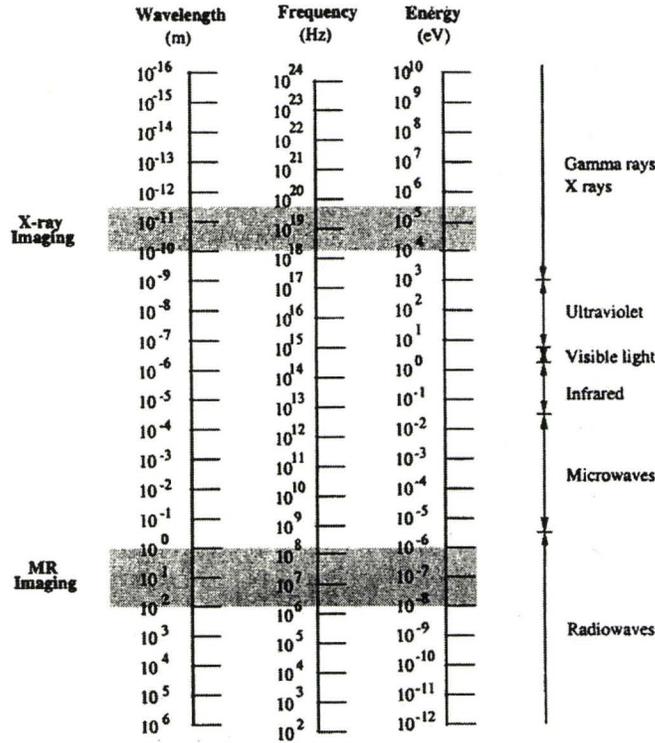


Figure 1.1: Electromagnetic spectrum.

Contrast resolution allows visualizations of low-density objects with similar soft tissue characteristics, for instance, liver-spleen or white matter-gray matter [7]. MRI has superior contrast resolution that surpasses leading tomographic instruments in that area, namely Ultrasound and CAT tomography, as well as, equal or better spatial resolution. Also, the resolution coefficient is not dependent of the strength of the rays, rather it is a function of several intrinsic properties of the tissue being imaged. The three most important properties include spin density, spin lattice relaxation time and spin-spin relaxation time, which will be discussed later. Finally, MRI has advantages over all other tomographic techniques in terms of the MR signals generated by the procedure. The MR signals used for image formation come directly from the object itself and are extremely rich in information content. In this sense,

MRI is a form of *emission* tomography similar to PET and SPECT, yet, it does not involve the injection of radioactive isotopes. Also, using the MR signals, MRI processors can construct two-dimensional sectional images in any orientation, three-dimensional volumetric images, and even four-dimensional images representing spatial-spectral distributions [18]. In addition, no mechanical adjustments to the imaging machinery are necessary in generating these images. Thus, although MRI has some side effects, the procedure is safe and more advanced than any other tomographic modality, making it a superior imaging technique.

1.2 MRI Basics

In constructing an MRI image, three key components are necessary: A main magnet, which creates a strong uniform static magnetic field, an RF coil, which is responsible for altering the uniform magnetic field and generating a signal, and finally, a computer processor, which produces an image from the data in the signal [7]. An oversimplification of the MRI procedure would be as follows: First, a specimen or object is positioned in a large main magnet, which creates an uniform magnetic field in one axial direction, known as B_0 . If we were able to look at the magnetic moments within the specimen or object, we would observe that they are all pointing in the same direction, B_0 . Next, an RF coil produces an “RF pulse,” which causes the magnetic moments to tip into an orthogonal direction of B_0 , called the transverse plane [17]. The RF pulse is only aimed at a specific portion of the object or specimen that the user intends to image. When the magnetic moments tip into the transverse plane they generate a signal that is picked up by receiver coils, also part of the RF coil. In addition, the RF pulse is accompanied by a gradient sequence that is used to spatially modulate the signals orientation [7]. Finally, the data information generated by the signal is formulated into a final image with the

assistance of a computer. This process is illustrated in Figure 1.2, where the first arrow represents the RF pulse generated by the RF coil and the accompanying gradient sequence. The second arrow represents the data processing of the signal into an image by a computer processor. Before going on, some

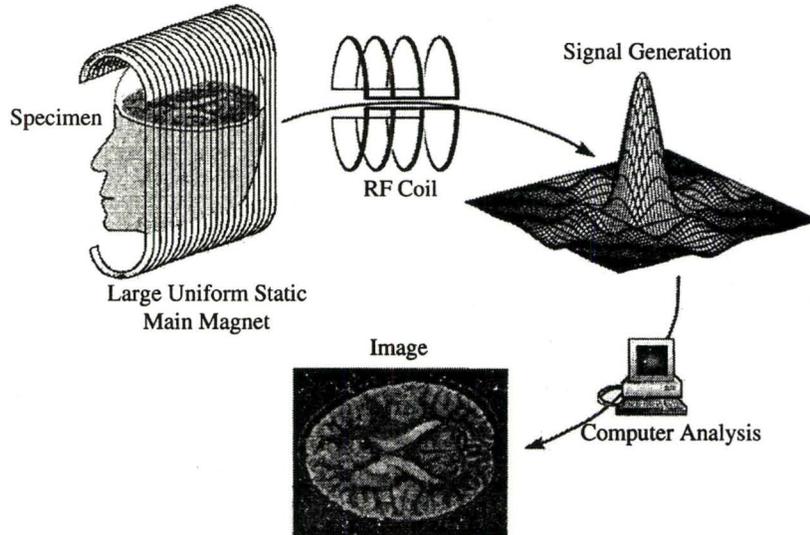


Figure 1.2: The MR imaging process.

important properties of the MRI hardware components should be discussed.

The Main Magnet

There are different types of main MRI magnets that can be utilized; however, generally super-conducting magnets are used as they produce high magnetic field strengths. Advantages of high field strengths include better signal-to-noise ratio and spectral resolution. The signal-to-noise ratio accounts for the amount of “useable” signal data information generated by the RF pulse. Spectral resolution describes the minimum frequency difference that can be detected in an MR spectrum, which relates to the resolution quality of the final image [17].

The RF System

An RF coil consists of two components, the transmitter coil and the receiver coil. The transmitter coil is capable of generating a rotating magnetic field, referred to as \vec{B}_{rf} , to excite the magnetic moments into the transverse plane. The receiver coil is responsible for signal detection, as it converts precessing magnetization into an electrical signal. A desirable feature of the RF coil is that it provides a uniform \vec{B}_{rf} field and high detection sensitivity without exceeding specific limits of SAR [18]. There are many different types of RF coils, each with their own specific size and shape depending on their application.

The Gradient System

The magnetic field gradient system consists of three orthogonal gradient coils that are integrated into the bore of the main magnet. Gradient coils are designed to produce time-varying magnetic fields of controlled spatial non-uniformity, whose formal definition will be described in Chapter 2. The gradient system is a critical component of MRI as it is essential for signal localization. By producing microscope differences in the strength of B_0 , the signal generated by the RF pulse has information on its spatial location with respect to the object or specimen being imaged. Important specifications for a gradient system include, the maximum gradient strength, and the rate at which a desired gradient strength can be obtained, known as the *slew rate* [18].

Signal Processing

The signal produced by the RF pulse is amplified, digitized, transformed, and then combined together with other signals to form a final image. The computations involved in processing the signal data are well-known image reconstruction problems, common to many tomographic imaging modalities

[11]. The signal processing stage of MRI has received much attention by software developers, however, the essential computations of each software design are the same, which will be described in Chapter 6.

1.3 MRI Pulse Concepts

We have alluded to the underlying ideas involved in MRI, which are based on the interactions of nuclear spin and an external magnetic field. Actually, imaging a specimen or object rests on the MR systems ability to manipulate the hydrogen nucleus, particularly the hydrogen proton. The spinning motion of the hydrogen proton, caused by a microscopic magnetic field mentioned earlier, is described as precession. Precession is the “magnetic fingerprint” specific to the environment an atom resides in, as well as, the strength of the external magnetic field, B_0 . The speed at which a proton spins, or precessional frequency, is defined in the Larmor equation, shown in Chapter 2. For example, given a general MRI external field strength, the hydrogen nucleus precessional frequency is 85 200 Hz, just below the FM range of radio broadcasting [17]. When a pulse is applied to a volume of hydrogen protons at precisely the same radio frequency, they become excited. Hence, the name of the pulse was adopted to RF (radio frequency) pulse, since its strength is equivalent to that of radio waves. Further, when the hydrogen protons become excited their magnetic moments tip away from the external field direction. The magnetic field produced by the aggregate proton spins yield a change in the flux of a nearby receiver coil [17]. In fact, this change in flux will be the greatest when the magnetic moments of the hydrogen atoms are directly perpendicular to the B_0 axis. This is known as a 90^0 pulse, and if the original magnitude of the magnetic moment vectors in the B_0 direction was M_0 , then the resulting transverse magnetization (i.e., magnetization in the x , y plane if B_0 is parallel to the z -axis) has a magnitude of M_0 . In addition,

when a magnetic moment is excited into the transverse plane, the tipping motion exerted by the RF pulse is actually a rotation about the B_0 axis, show in Figure 1.3. This is due to the properties of precession, and for this reason

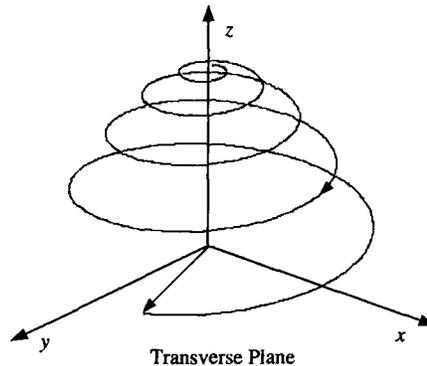


Figure 1.3: The effect of an RF pulse on an individual magnetic moment.

many of the equations underlying MRI are described in the rotating frame of reference. A simple childhood carousel can explain the difference between a rotating frame of reference and a laboratory frame of reference. If you tried to locate the position of a child while you were “on” the carousel, or in the rotating frame of reference, it would be much easier than if you were “off” the carousel, in the laboratory frame of reference.

After a number of RF pulses are given to a specimen or object, the signals they produce are mathematically combined to form the final MRI image. Within each RF pulse an accompanying gradient sequence is applied to the specimen or object being imaged. This enables the user, or computer processor, to identify what part of the specimen or object the pulse was derived from. For example, if a gradient was applied in one axial direction, the precessional frequency of the magnetic moment vectors on one side of the object would be slightly higher than the other, with regards to that axis. Hence,

depending on the precessional frequency of the signal, the user or computer processor would be able to identify what part of the object a signal comes from with respect to that axial direction. This can be expanded to account for gradients in three dimensions, as used with practical MRI machines.

Although the concept of producing MR signals may seem fairly simple, reliable RF pulses and gradient sequences have not yet been perfected. The improvement of RF pulse time, image resolution, signal quality, and SAR reduction, are just a few areas within MRI pulse designs that need further development. In the next chapter we will introduce an RF pulse and gradient sequence called the Variable Rate Selective Excitation pulse that is designed to reduce SAR and improve signal quality.

Chapter 2

Variable Rate Selective Excitation (VERSE)

In this chapter we describe the motivation and detail the intricate design involved in the Variable Rate Selective Excitation Radio Frequency pulse. The mathematical formulations governing MRI are introduced first so that the basis of the model and its underpinnings are appreciated.

2.1 Background

To understand the implications and effects of Variable Rate Selective Excitation (VERSE) in Magnetic Resonance Imaging (MRI) we should begin with a short review of basic chemistry. It is known that any biological specimen or physical object can be broken down into molecules that are composed of many atoms. Further, each atom contains orbiting electrons and a nucleus that has a finite radius, mass and net electric charge. More specifically, nuclei with odd atomic weights and/or odd atomic numbers possess an angular momentum referred to as spin [18]. Due to nuclear spin and electric charge a microscopic magnetic field is generated within each nuclei. An ensemble of nuclei, such as one present in an object or specimen, produces a “spin system,” illustrated

in Figure 2.1. The microscopic magnetic field of each atom is represented by

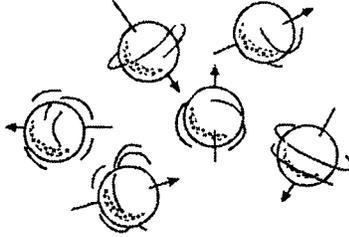


Figure 2.1: Magnetic moment vectors pointing in random directions.

a vector quantity $\vec{\mu}(t)$, and it is known as the nuclear magnetic dipole moment or magnetic moment. The aggregate magnetic moment of all nuclei in a given unit volume is described as their magnetization, an intrinsic property of atoms that enables Magnetic Resonance Imaging (MRI). Consider a sufficient volume (V) of protons or nuclei known as a voxel, the magnetization, $\vec{M}(t)$, is

$$\vec{M}(t) = \frac{1}{V} \sum_{\text{Protons in } V} \vec{\mu}_i(t), \quad (2.1.1)$$

where $\vec{\mu}_i(t)$ is the magnetic moment of proton i in V at time t . Like the needle of a compass, when an external magnetic field, $\vec{B}(t)$, is applied to a specimen or object the magnetic moment vectors align in the direction of the field. However, instead of mimicking the external field, the magnetic moment vectors behave like tiny spinning gyroscopes, a phenomenon known as precession, seen in Figure 2.2. Hence, the magnetic moment vectors precess in the direction of the external field and generate a net magnetization. The precessional frequency, $\omega(t)$, of a magnetization vector in the presence of an external magnetic field is represented by the fundamental Larmor relation

$$\omega(t) = \gamma \vec{B}(t),$$

where γ is the gyromagnetic constant. The precessional frequency is an essential part of applying any type of Radio Frequency (RF) pulse used in MRI,

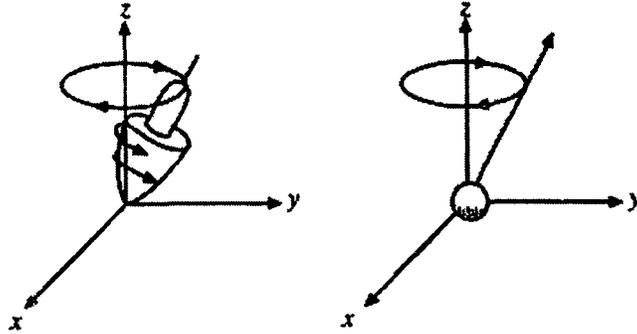


Figure 2.2: Precession of nuclear spin about an external magnetic field in the z -axis direction, similar to the wobbling of a spinning top.

as we will discuss later. Now, using the expression for torque on a magnetic moment due to an external magnetic field, we have

$$\frac{d\vec{\mu}(t)}{dt} = \gamma \vec{\mu}(t) \times \vec{B}(t).$$

Thus, a proton's magnetic moment with respect to time can be incorporated into (2.1.1) that results in,

$$\frac{d\vec{M}(t)}{dt} = \frac{1}{V} \sum_i \frac{d\vec{\mu}_i(t)}{dt} = \frac{\gamma}{V} \sum_i \vec{\mu}_i(t) \times \vec{B}(t)$$

and therefore,

$$\frac{d\vec{M}(t)}{dt} = \gamma \vec{M}(t) \times \vec{B}(t). \quad (2.1.2)$$

The relationship between proton interactions and an external magnetic field leads to additional terms in equation (2.1.2) that is described as the Bloch equation, an important stepping stone in the development of MRI. There are two types of proton interactions, spin-lattice interactions and spin-spin interactions. In spin-lattice interactions, a magnetic moment will tend to line up parallel to the external magnetic field, in its minimum energy state

[17]. As a result, the rate of change of longitudinal magnetization, magnetization in the z -axis direction, is proportional to the difference between initial magnetization and the z coordinate component of the magnetization vector. This proportionality relation becomes exact with the addition of an experimentally-determined parameter, τ_1 , which represents the inverse of the time scaled growth rate of longitudinal magnetization [17]. Thus, for the rate of change of longitudinal magnetization, $dM_z(t)/dt$, we have

$$\frac{dM_z(t)}{dt} = \frac{1}{\tau_1}(M_0 - M_z(t)),$$

where M_0 is the initial magnetization in the z -axis direction, $M_z(t)$ is the z coordinate component of the magnetization vector and τ_1 is the longitudinal magnetization parameter that has different values for various tissues. Also, the recovery of longitudinal magnetization is expressed by a relaxation rate parameter, R_1 , which is simply equivalent to the inverse of τ_1 [17]. Furthermore, this process is termed *Longitudinal Relaxation*, which is a consequence of spin-lattice proton interactions. Spin-spin interactions are slightly more intricate since spins experience local fields that are combinations of the applied field and the fields of their neighbours [17]. Since variations in the local fields produce different local precessional frequencies, individual spins tend to “fan out” and de-phase. This ultimately leads to a decay of the magnetization vector in the x - y plane, described as transverse magnetization, $\vec{M}_\perp(t)$. This process involves another experimentally-determined parameter, τ_2 , which is known as the transverse magnetization parameter and also has various values for different tissues. Thus, for the rate of change of transverse magnetization, $d\vec{M}_\perp(t)/dt$, we have

$$\frac{d\vec{M}_\perp(t)}{dt} = -\frac{1}{\tau_2}\vec{M}_\perp(t).$$

The decay or reduction rate of transverse magnetization is known as *Transverse Relaxation*. This process requires a second relaxation rate parameter,

R_2 , where $R_2 = 1/\tau_2$. Thus, by combining these two types of interactions and inducing an external magnetic field in the z -axis direction, the Bloch equation is as follows

$$\frac{d\vec{M}(t)}{dt} = \gamma\vec{M}(t) \times \vec{B}(t) + \frac{1}{\tau_1}(M_0 - M_z(t))\hat{z} - \frac{1}{\tau_2}\vec{M}_\perp(t),$$

where \hat{z} is the z -axis unit vector, M_0 is the initial magnetization in the \hat{z} direction and

$$\vec{M}(t) = \begin{bmatrix} M_x(t) \\ M_y(t) \\ M_z(t) \end{bmatrix}, \quad \vec{M}_\perp(t) = \begin{bmatrix} M_x(t) \\ M_y(t) \\ 0 \end{bmatrix}$$

are respectively the net and transverse magnetization vectors. Furthermore, we introduce the vector coordinates $b_x(t)$, $b_y(t)$ and $b_z(t)$ of the external magnetic field $\vec{B}(t)$, i.e.,

$$\vec{B}(t) = \begin{bmatrix} b_x(t) \\ b_y(t) \\ b_z(t) \end{bmatrix}.$$

2.2 The VERSE Model

When processing an image, a number of precise Radio Frequency (RF) pulses are applied in combination with synchronized gradients in different directions [18]. Gradients are designed to produce time-altering magnetic fields of linear-varying magnitude that ultimately allows our MRI processor to differentiate between specific sections of our specimen. An RF pulse at the Larmor frequency excites the magnetization vectors of a voxel of protons into the transverse (x, y) plane where an externally measurable signal is generated. This signal can then be amplified, digitized and Fourier Transformed (FT) into an

image.

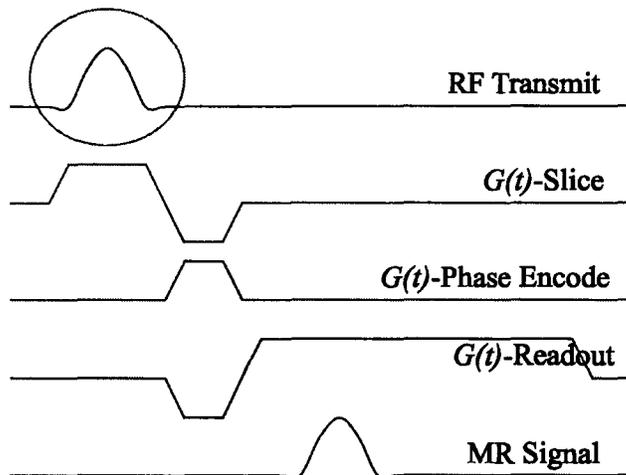


Figure 2.3: A generic NMR pulse imaging sequence.

Based on the fundamental Larmor relation and the ideas proposed by Conolly *et al.* [9], we developed a new variant of the Variable Rate Selective Excitation (VERSE) pulse. VERSE pulses are designed to perform a transverse excitation using only a fraction of the field strength in order to reduce patient heating caused by long, high energy pulses. The key innovation is to allow a “trade off” between time and amplitude. By lowering RF pulse amplitude the duration of the pulse may be extended [9]. As illustrated in Figure 2.3, RF pulses are generally very polarized (circled in the figure); our aim is to uniformly or more evenly distribute the signal. This flattened redistribution of the pulse causes a decrease in the Specific Absorption Rate (SAR) of our sample and hence, reduces the high signal amplitude found in other pulse sequences (i.e., spin echo) [9]. High levels of SAR constitute an increase in specimen temperature during MRI procedures. Thus, by uniformly distributing the pulse amplitude over the excitation interval, the SAR of a

selective RF pulse is decreased. Mathematically, this equates to minimizing the external magnetic field generated by the RF pulse ($\vec{B}_{\text{rf}}(t)$), and therefore our objective is

$$\min \text{SAR} = \int_0^T |\vec{B}_{\text{rf}}(t)|^2 dt = \int_0^T b_x^2(t) + b_y^2(t) dt,$$

where T is the time at the end of RF pulse and

$$\vec{B}_{\text{rf}}(t) = \begin{bmatrix} b_x(t) \\ b_y(t) \\ 0 \end{bmatrix}.$$

Earlier we mentioned that MRI is based on the interaction of nuclear spin with an external magnetic field; hence, $\vec{B}_{\text{rf}}(t)$ is simply the vertical and horizontal components of $\vec{B}(t)$. Also, if low pulse amplitudes are produced by the VERSE pulse, the duration T of the pulse can be increased.

Another part of MRI that we have not mentioned comes from the fact that since all our magnetization vectors are spinning, there exists a rotational frame of reference. However, if we set up our equations such that we are in the rotating frame of reference then we exclude the uniform magnetic field generated by the main super-conducting magnet, B_0 . Instead we are left with the magnetic field of our RF pulse, $\vec{B}_{\text{rf}}(t)$, and our gradient

$$\vec{G}(t, s) = \begin{bmatrix} 0 \\ 0 \\ sG(t) \end{bmatrix},$$

where $sG(t)$ is the gradient value at coordinate position s . Earlier we alluded to the function of gradients and their importance in producing time-altering magnetic fields. Hence, different parts of a specimen experience different gradient field strengths. Thus, by multiplying a constant gradient value by different coordinate positions s , we have potentially produced an equivalent

linear relationship to what is used in practice. Fundamentally, coordinate positions s have allowed us to split a specimen or object into “planes” or slices along the s direction, which for the purposes of this paper will be parallel to \hat{z} , as depicted in Figure 2.4. Here, s corresponds to a specific coordinate

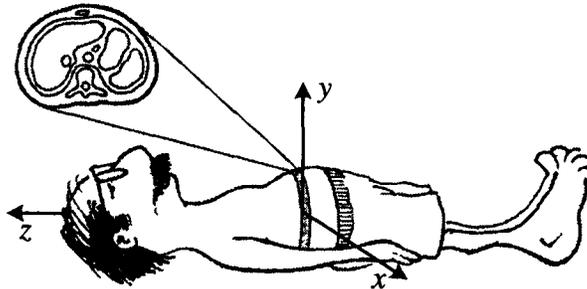


Figure 2.4: Specimen separated into planes or slices about the z -axis.

value depending on its respective position and further it has a precise and representative gradient strength. As mentioned, an RF pulse excites particular voxels of protons into the transverse (x, y) plane where a signal is generated that is eventually processed into an image. Thus, we will use s to distinguish between voxels that have been stimulated into the transverse plane by an RF pulse and those that have not. Coordinate positions, s , of voxels that are stimulated into the transverse plane will be recorded and referred to as being “in the slice.” Those voxels that are not tipped into the transverse plane will be referred to as being “outside of the slice,” whose respective coordinate positions, s , will also be noted. Since any specimen or object we intend to image will have a fixed length, given $s \in S$, we will restrict this semi-infinite constraint by choosing a finite set $S \subset \mathbb{R}$. S can then be further partitioned into the disjoint union of sets $S_{\text{in}} \cup S_{\text{out}}$, where S_{in} represents the coordinate positions in the slice and S_{out} represents the voxels that we do not want to tip into the transverse plane, those which are outside of the slice. For each coordi-

nate position, $s \in S$, we add constraints corresponding to the Bloch equation however, boundary constraints correspond to different conditions depending on the position of the slice, as we will discuss later. Fundamentally, voxels in S_{in} , ensure uniform magnetic tipping into the transverse plane, whereas the $s \in S_{\text{out}}$, certify external magnetization is preserved.

Thus, we now have $\vec{B}(t)$ with respect to coordinate positions s , whereby $b_x(t)$ and $b_y(t)$ are independent of s , hence

$$\begin{aligned} \vec{B}(t, s) &= \vec{B}_{\text{rf}}(t) + \vec{G}(t, s) \\ &= \begin{bmatrix} b_x(t) \\ b_y(t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ sG(t) \end{bmatrix} = \begin{bmatrix} b_x(t) \\ b_y(t) \\ sG(t) \end{bmatrix}. \end{aligned}$$

Also, since $\vec{B}(t, s)$ has divided the \hat{z} component of our external magnetization into coordinate components, the same notation must be introduced into our net magnetization, hence

$$\vec{M}(t, s) = \begin{bmatrix} M_x(t, s) \\ M_y(t, s) \\ M_z(t, s) \end{bmatrix},$$

where s denotes the magnetization vector at specific coordinate position. Also, since VERSE pulses typically have short sampling times we will assume there is no proton interactions or relaxation, thus, from the Bloch equation we are left with

$$\begin{aligned} \frac{d\vec{M}(t, s)}{dt} &= \gamma \vec{M}(t, s) \times \vec{B}(t, s) \\ &= \gamma \vec{M}(t, s) \times [b_x(t), b_y(t), sG(t)]^T. \end{aligned}$$

Hence, we have

$$\begin{aligned} \vec{M}(t, s) \times \vec{B}(t, s) &= \begin{vmatrix} i & j & k \\ b_x(t) & b_y(t) & sG(t) \\ M_x(t, s) & M_y(t, s) & M_z(t, s) \end{vmatrix} \\ &= \begin{bmatrix} 0 & -sG(t) & b_y(t) \\ sG(t) & 0 & -b_x(t) \\ -b_y(t) & b_x(t) & 0 \end{bmatrix} \begin{bmatrix} M_x(t, s) \\ M_y(t, s) \\ M_z(t, s) \end{bmatrix}, \end{aligned}$$

and finally

$$\frac{d\vec{M}(t, s)}{dt} = \gamma \begin{bmatrix} 0 & -sG(t) & b_y(t) \\ sG(t) & 0 & -b_x(t) \\ -b_y(t) & b_x(t) & 0 \end{bmatrix} \begin{bmatrix} M_x(t, s) \\ M_y(t, s) \\ M_z(t, s) \end{bmatrix}. \quad (2.2.1)$$

When stimulating a specific segment of a specimen by an RF pulse, some of the magnetization vectors are fully tipped into the transverse plane, partially tipped, and those lying outside the slice are minimally affected. The magnetization vectors that are only partially tipped into the transverse plane are described as having off-resonance and tend to disrupt pulse sequences and distort the final MRI image [17]. In anticipation of removing such inhomogeneities we introduce the angle α , at which net magnetization moves from the \hat{z} direction to the transverse plane. By convention, α will be the greatest at the end of our RF pulse, at time T , and since we are in the rotating frame we can remove the y -axis from our equations. Thus, we can eliminate off-resonance s coordinates by bounding voxels affected by the pulse

$$\left\| \begin{bmatrix} M_0 \sin(\alpha) \\ 0 \\ M_0 \cos(\alpha) \end{bmatrix} - \begin{bmatrix} M_x(T, s) \\ M_y(T, s) \\ M_z(T, s) \end{bmatrix} \right\| \leq \varepsilon_1,$$

and those not affected by the pulse, with $\alpha = 0$, hence

$$\left\| \begin{bmatrix} 0 \\ 0 \\ M_0 \end{bmatrix} - \begin{bmatrix} M_x(T, s) \\ M_y(T, s) \\ M_z(T, s) \end{bmatrix} \right\| \leq \varepsilon_2,$$

where $\varepsilon_1, \varepsilon_2 \geq 0$. Therefore, by comparing these two bounds we can determine the s coordinates from which we would like the signal to be generated and exclude off-resonance.

Another factor we must integrate into our pulse is slew rate, $W(t)$, also called gradient-echo rise time. This identifies how fast a magnetic gradient field can be ramped to different gradient field strengths [9]. As a result, higher slew rates enable shorter measurement times since, the signal generated by the RF pulse to be imaged is dependent on it. Thus, the slew rate and gradient field strength together determine an upper bound on the speed and ultimately minimum time needed to perform the pulse. Thus, there must be a bound on these two entities in our constraints,

$$W(t) = \begin{cases} |G(t)| \leq G_{\max}, \\ \left| \frac{dG(t)}{dt} \right| \leq W_{\max}. \end{cases}$$

Finally, we have the semi-infinite nonlinear optimization problem

$$\min \text{SAR} = \int_0^T b_x^2(t) + b_y^2(t) dt, \quad (2.2.2)$$

subject to,

$$\frac{d\vec{M}(t, s)}{dt} - \gamma \begin{bmatrix} 0 & -sG(t) & b_y(t) \\ sG(t) & 0 & -b_x(t) \\ -b_y(t) & b_x(t) & 0 \end{bmatrix} \begin{bmatrix} M_x(t, s) \\ M_y(t, s) \\ M_z(t, s) \end{bmatrix} = 0, \quad (2.2.3S)$$

$$\left\| \begin{bmatrix} M_0 \sin(\alpha) \\ 0 \\ M_0 \cos(\alpha) \end{bmatrix} - \begin{bmatrix} M_x(T, s) \\ M_y(T, s) \\ M_z(T, s) \end{bmatrix} \right\| \leq \varepsilon_1, \quad (2.2.4S_{in})$$

$$\left\| \begin{bmatrix} 0 \\ 0 \\ M_0 \end{bmatrix} - \begin{bmatrix} M_x(T, s) \\ M_y(T, s) \\ M_z(T, s) \end{bmatrix} \right\| \leq \varepsilon_2, \quad (2.2.4S_{out})$$

$$|G(t)| \leq G_{\max}, \quad (2.2.5)$$

$$\left| \frac{dG(t)}{dt} \right| \leq W_{\max}, \quad (2.2.6)$$

$$M_x(0, s) = 0, \quad (2.2.7S)$$

$$M_y(0, s) = 0, \quad (2.2.8S)$$

$$M_z(0, s) = M_0, \quad (2.2.9S)$$

where equations (2.2.2) – (2.2.9S) apply $\forall s \in S, t \in [0, T]$. Expanding the first constraint (2.2.3S) produces the following equations,

$$\frac{dM_x(t, s)}{dt} = \gamma[-sG(t)M_y(t, s) + b_y(t)M_z(t, s)], \quad (2.2.10S)$$

$$\frac{dM_y(t, s)}{dt} = \gamma[sG(t)M_x(t, s) - b_x(t)M_z(t, s)], \quad (2.2.11S)$$

$$\frac{dM_z(t, s)}{dt} = \gamma[-b_y(t)M_x(t, s) + b_x(t)M_y(t, s)]. \quad (2.2.12S)$$

Thus, depending on our bound for the pulse, we will construct two sets of constraints, one for the voxels $S_{in} \in \mathbb{R}$ that will be stimulated by the RF pulse and one for those that will not, $S_{out} \in \mathbb{R}$. Which indices are affected will be determined by constraints (2.2.4S_{in}) and (2.2.4S_{out}). Thus, if we are given the voxels, (2.2.4S_{in}) our pulse affects, then we can apply equations (2.2.10S), (2.2.11S) and (2.2.12S) respectively. The same can be done for the other set of voxels minimally affected by the RF pulse, S_{out} .

2.3 Discretization

By separating our specimen into coordinate positions we have ultimately created two dimensional segments that are similar to records in a record box, whereby $s \in S$ represents the transverse plane at a particular position. Now we will discretize S into coordinate positions s_1, s_2, \dots, s_n , where n is the total number of slices. Figure 2.5 represents what an MRI processor would interpret for a given object under a particular gradient where we have incorporated coordinate positions. Previously we defined S_{in} as the coordinate

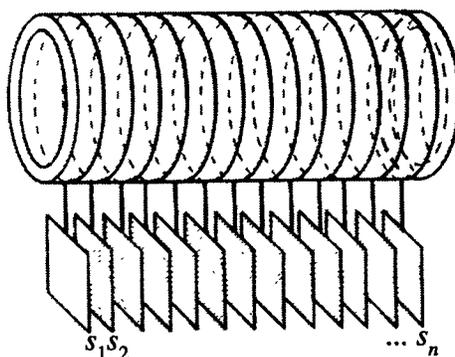


Figure 2.5: MRI processor separating partitions of an object into the transverse plane by using different gradient strengths at each coordinate position s_1, s_2, \dots, s_n .

positions whose voxels have been tipped into the transverse plane by an RF pulse. Now S_{in} will consist of a finite band of particular coordinate positions whose magnetization vectors have been excited into the transverse plane, hence, $S_{\text{in}} = s_k, \dots, s_{k+\delta}$, where $1 < k \leq k + \delta < n$, $\delta \geq 0$ and $k, \delta \in \mathbb{Z}$. Subsequently S_{out} , which was defined as positions that were not stimulated in the transverse plane, will consist of all coordinate positions not in S_{in} , thus, $S_{\text{out}} = s_1, \dots, s_{k-1}, s_{(k+\delta)+1}, \dots, s_n$. Figure 2.6 represents how $s_i \in S$ for $i = 1, \dots, n$ would separate magnetization vectors into coordinate positions

that have been tipped into the transverse plane, and those that have not. One

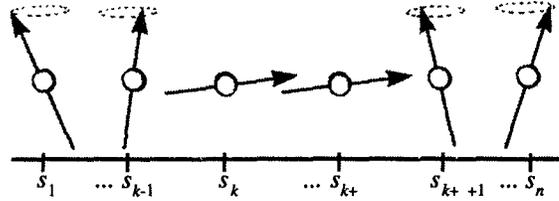


Figure 2.6: Separating magnetization vectors into coordinate positions which are in the slice, S_{in} , and out, S_{out} .

should also note that we have only discretized with respect to coordinate positions $s_i \in S$, not time t . Furthermore, we will define the coordinate position in S_{in} where RF pulse stimulation begins as \underline{s} , and similarly, the position where it stops as \bar{s} . Thus, we have $\underline{s} = s_k$ and $\bar{s} = s_{k+\delta}$, and we can now state the coordinate positions in the slice as $S_{\text{in}} = [\underline{s}, \bar{s}]$. The first position where RF stimulation is a minimum, closest to \underline{s} , but in S_{out} and towards the direction of s_1 , will be defined as \underline{s}_l . As well, the same will be done for the position closest to \bar{s} , which is in S_{out} and towards the direction of s_n , defined as \bar{s}_u . Consequently, $\underline{s}_l = s_{k-1}$ and $\bar{s}_u = s_{(k+\delta)+1}$, and therefore the coordinate positions outside the slice can be represented as $S_{\text{out}} = [s_1, \underline{s}_l] \cup [\bar{s}_u, s_n]$. As depicted in Figure 2.6, S_{in} is located between the two subintervals of S_{out} , where $s_i \in S_{\text{in}}$ is centered around 0, leaving S_{out} subintervals, $[s_1, \underline{s}_l] < 0$ and $[\bar{s}_u, s_n] > 0$. As well, $[s_1, \underline{s}_l]$ and $[\bar{s}_u, s_n]$ are symmetric with respect to each other, hence, the length of these subintervals are equivalent, $s_{k-1} - s_1 = s_n - s_{(k+\delta)+1}$. Furthermore, the difference between respective coordinate positions within each

interval are equal to one another such that,

$$\begin{aligned}
 s_2 - s_1 &= s_n - s_{n-1} \\
 s_3 - s_2 &= s_{n-1} - s_{n-2} \\
 &\vdots \\
 &\vdots \\
 s_{k-1} - s_{k-2} &= s_{(k+\delta)+2} - s_{(k+\delta)+1}.
 \end{aligned}
 \tag{2.3.1}$$

Also note that the discretization points, s_i , within any interval $[s_1, \underline{s}_l]$, $[\underline{s}, \bar{s}]$ and $[\bar{s}_u, s_n]$ do not necessarily have to be uniformly distributed and thus, more coordinate positions could be positioned closer to the boundaries of S_{in} and S_{out} . The distance between coordinate positions $(\underline{s}_l, \underline{s})$ and (\bar{s}, \bar{s}_u) will be much larger in comparison to other increments of s_i . This is typically the area where voxels that have off-resonance characteristics are located. As mentioned earlier, magnetization vectors having off-resonance tend to disrupt pulse sequences and distort the MRI image. For this reason we will define the tolerance gaps of finite length where off-resonance prominently resides, between $(\underline{s}_l, \underline{s})$ and (\bar{s}, \bar{s}_u) , as S_0 . Hence, S can now be partitioned into $S_{\text{in}} \cup S_{\text{out}} \cup S_0$ where a general sequence of the intervals would be $S_{\text{out}}, S_0, S_{\text{in}}, S_0, S_{\text{out}}$.

2.4 VERSE Penalty

An important component of the model now becomes evident, the nonlinear optimization problem defined in (2.2.2) – (2.2.9S) may be infeasible or difficult to solve when the number n of $s_i \in S$ becomes large and the slices are close together. In particular, constraints (2.2.4 S_{in}) and (2.2.4 S_{out}) potentially pose a threat to the feasibility of the problem as the number of variables increases. A penalty for the violation of these constraints can be imposed such that an optimal solution is located for problems with large numbers of variables and close s_i coordinate positions. The basic idea in penalty methods is to essentially eliminate particular constraints and add a penalty term to the ob-

jective function that prescribes high cost to infeasible points [1]. The penalty parameter determines the severity of the penalty and as a consequence, the extent to which the resulting unconstrained problem approximates the original constrained one. Thus, returning to the semi-infinite nonlinear optimization problem formulated in Section 2, we introduce penalty variables ξ_1 and ξ_2 to constraints (2.2.4S_{in}) – (2.2.4S_{out}), and the optimization problem becomes,

$$\min \text{SAR} = \int_0^T b_x^2(t) + b_y^2(t) dt + \xi_1 \zeta_1 + \xi_2 \zeta_2, \quad (2.4.1)$$

subject to,

$$\frac{d\vec{M}(t, s_i)}{dt} - \gamma \begin{bmatrix} 0 & -s_i G(t) & b_y(t) \\ s_i G(t) & 0 & -b_x(t) \\ -b_y(t) & b_x(t) & 0 \end{bmatrix} \begin{bmatrix} M_x(t, s_i) \\ M_y(t, s_i) \\ M_z(t, s_i) \end{bmatrix} = 0, \quad (2.4.2S)$$

$$\left\| \begin{bmatrix} M_0 \sin(\alpha) \\ 0 \\ M_0 \cos(\alpha) \end{bmatrix} - \begin{bmatrix} M_x(T, s_i) \\ M_y(T, s_i) \\ M_z(T, s_i) \end{bmatrix} \right\| \leq \varepsilon_1 + \xi_1, \quad (2.4.3S_{\text{in}})$$

$$\left\| \begin{bmatrix} 0 \\ 0 \\ M_0 \end{bmatrix} - \begin{bmatrix} M_x(T, s_i) \\ M_y(T, s_i) \\ M_z(T, s_i) \end{bmatrix} \right\| \leq \varepsilon_2 + \xi_2, \quad (2.4.3S_{\text{out}})$$

$$|G(t)| \leq G_{\max}, \quad (2.4.4)$$

$$\left| \frac{dG(t)}{dt} \right| \leq W_{\max}, \quad (2.4.5)$$

$$M_x(0, s_i) = 0, \quad (2.4.6S)$$

$$M_y(0, s_i) = 0, \quad (2.4.7S)$$

$$M_z(0, s_i) = M_0, \quad (2.4.8S)$$

where $\zeta_1, \zeta_2 \in \mathbb{R}$ are scalar penalty parameters and equations (2.4.1) – (2.4.8S) apply $\forall s \in S, t \in [0, T]$. One should note that the larger the value of ζ_1 and/or ζ_2 , the less violated constraints (2.4.3S_{in}) and/or (2.4.3S_{out}) become.

In addition, as it is written, the penalty variables are applied to each $s_i \in S$ for constraints (2.4.3 S_{in}) and (2.4.3 S_{out}). However, depending on computational results, it may be appropriate to only penalize coordinate positions in the neighbourhood of the bounds $[\underline{s}_l, \underline{s}]$ and $[\bar{s}, \bar{s}_u]$. This would enhance the constraints on the optimization problem and only allow violations to occur at the most vulnerable points of the problem. Adding penalty variables and parameters to our optimization problem is an option that may not be necessary and is dependent on the number n of coordinate positions applied to the model and how close we would like $s_i \in S$ to be to one another. Hence, for the remainder of this paper we will omit writing out the penalty variables and parameters, however, the reader should note that they can easily be incorporated into the formulation.

2.5 Optimal Control Problem

The VERSE pulse formulation is a Nonlinear Optimization (NLO) problem that requires the objective function (2.2.2) to be minimized without violating the set of constraints (2.2.3 S) – (2.2.9 S). An NLO problem can be extended to an infinite number of variables where it can then be treated as an optimal control problem. Hence, an optimal control problem is an infinite-dimensional extension of an NLO problem, however, practical methods for solving optimal control problems require iterations with a finite set of variables and constraints [4]. Typically, optimal control problems are formulated as a collection of independent, state and control variables. By definition, state variables act collectively as the trajectory of the system, whereas, control variables determine the course of the process [8]. For the VERSE pulse problem the independent variable is time t , while the state and control variables are defined within the dynamics of the system. Thus, for a problem with n slices, the state variable

is the $3n + 1$ dimensional vector

$$\Omega(t) = [M_x(t, s_1), M_y(t, s_1), M_z(t, s_1), \dots, M_x(t, s_n), M_y(t, s_n), M_z(t, s_n), G(t)]^T,$$

where $\Omega(t) \in \mathbb{R}^{3n+1}$. Similarly, the three dimensional control vector is

$$\Phi(t) = [b_x(t), b_y(t), W(t)]^T$$

with $\Phi(t) \in \mathbb{R}^3$. Subsequently, for any VERSE pulse problem we solve, the total number of state and control variables are $3n + 4$. Our system is governed by differential equations (2.2.10S), (2.2.11S), (2.2.12S) and slew rate, where for $i = 1, \dots, n$ we have

$$\begin{aligned} \frac{dM_x(t, s_i)}{dt} &= \gamma[-s_i G(t) M_y(t, s_i) + b_y(t) M_z(t, s_i)], \\ \frac{dM_y(t, s_i)}{dt} &= \gamma[s_i G(t) M_x(t, s_i) - b_x(t) M_z(t, s_i)], \\ \frac{dM_z(t, s_i)}{dt} &= \gamma[-b_y(t) M_x(t, s_i) + b_x(t) M_y(t, s_i)], \\ \frac{dG(t)}{dt} &= W(t). \end{aligned}$$

This can then be represented as a function of state and control variables, namely

$$f(\Omega(t), \Phi(t)) = \begin{bmatrix} \frac{dM_x(t, s_1)}{dt} \\ \frac{dM_y(t, s_1)}{dt} \\ \frac{dM_z(t, s_1)}{dt} \\ \vdots \\ \frac{dM_x(t, s_n)}{dt} \\ \frac{dM_y(t, s_n)}{dt} \\ \frac{dM_z(t, s_n)}{dt} \\ \frac{dG(t)}{dt} \end{bmatrix}, \quad (2.5.1)$$

where $f(\Omega(t), \Phi(t))$ is a $3n + 1$ dimensional vector. In addition, the solution must also satisfy path constraints $G(t)$ and $W(t)$. For our problem bounds can be imposed on the state variable,

$$-G_{\max} \leq G(t) \leq G_{\max} \quad (2.5.2)$$

and the control variable,

$$-W_{\max} \leq W(t) \leq W_{\max}, \quad (2.5.3)$$

which pertains to constraints (2.2.5) and (2.2.6), respectively. Therefore, we will define our path constraints by the vector

$$\Psi(\Omega(t), \Phi(t)) = \begin{bmatrix} G(t) \\ W(t) \end{bmatrix}, \quad (2.5.4)$$

which satisfies

$$\Psi_L \leq \Psi(\Omega(t), \Phi(t)) \leq \Psi_U, \quad (2.5.5)$$

where

$$-\Psi_L = \Psi_U = \begin{bmatrix} G_{\max} \\ W_{\max} \end{bmatrix}.$$

In anticipation of finding an optimal solution, boundary conditions define the values of particular state variables at the start and end time of our evaluation. This allows the value of the dynamic variables at the beginning and end of our time interval to be pre-defined [4]. Thus, the initial conditions at the start of the time interval, $t = 0$, are

$$M_x(0, s_i) = 0, \quad (2.5.6)$$

$$M_y(0, s_i) = 0, \quad (2.5.7)$$

$$M_z(0, s_i) = M_0, \quad (2.5.8)$$

again for $i = 1, \dots, n$. Hence, the values from (2.5.6)-(2.5.8) are entered into $\Omega(0)$ at the beginning of our evaluation. Terminal conditions that must

be satisfied at the end of the time interval are different for magnetization vectors in S_{in} , then for those in S_{out} . As depicted in constraints (2.2.4 S_{in}) and (2.2.4 S_{out}), at the end of our time interval, $t = T$, the terminal condition for the voxels $s_i \in S_{\text{in}}$ are,

$$-\varepsilon_1 \leq \begin{bmatrix} M_0 \sin(\alpha) \\ 0 \\ M_0 \cos(\alpha) \end{bmatrix} - \begin{bmatrix} M_x(T, s_i) \\ M_y(T, s_i) \\ M_z(T, s_i) \end{bmatrix} \leq \varepsilon_1. \quad (2.5.9)$$

Whereas, for voxels $s_i \in S_{\text{out}}$, we have the following terminal condition,

$$-\varepsilon_2 \leq \begin{bmatrix} 0 \\ 0 \\ M_0 \end{bmatrix} - \begin{bmatrix} M_x(T, s_i) \\ M_y(T, s_i) \\ M_z(T, s_i) \end{bmatrix} \leq \varepsilon_2. \quad (2.5.10)$$

Subsequently, the values for (2.5.9) and (2.5.10) are entered into $\Omega(T)$ at the end of the evaluation. Thus, the boundary conditions for the VERSE pulse problem will be expressed by

$$\psi_L \leq \psi(\Omega(t), \Phi(t)) \leq \psi_U, \quad (2.5.11)$$

where ψ_L and ψ_U contain the respective initial and terminal condition values found in (2.5.6)-(2.5.10). Penalty variables ξ_1 and ξ_2 would be incorporated into (2.5.9) and (2.5.10), respectively, if the problem required penalty terms. Also note that equality constraints can be imposed by using inequality ones by simply setting upper and lower bounds equal to one another, i.e., $\psi_L = \psi_U$. Finally, our objective function to be minimized will be expressed as

$$\int_0^T w(\Phi(t)) dt = \int_0^T b_x^2(t) + b_y^2(t) dt, \quad (2.5.12)$$

where $w(\Phi(t))$ is known as the quadrature function, which is commonly found in optimal control literature [5]. If penalty was part of our problem then $\xi_1 \zeta_1$ and $\xi_2 \zeta_2$ would be added to the quadrature function in (2.5.12). Collectively,

we refer to the functions evaluated during the time interval as

$$F(t) = \begin{bmatrix} f(\Omega(t), \Phi(t)) \\ \Psi(\Omega(t), \Phi(t)) \\ w(\Phi(t)) \end{bmatrix}, \quad (2.5.13)$$

the vector of continuous functions, however, boundary conditions evaluated at specific points are referred to as point functions [5]. Therefore, the solution to the optimal control problem requires

$$J(t) = \int_0^T w(\Phi(t)) dt \quad (2.5.14)$$

to be minimized. Notice that the objective function includes contributions evaluated at point functions and over the quadrature function [4].

Once the explicit details of our optimal control problem have been established, it is then possible to discretize with respect to time. Thus, to solve the VERSE pulse problem, we take N discretization points on the time interval $[0, T]$, including the end points, and hence

$$0 = t_1 < t_2 < \dots < T = t_N.$$

The discretized time intervals have the step size,

$$h_\ell = \lambda_\ell(t_N - t_1) = \lambda_\ell t_N,$$

where $\ell = 1, \dots, N - 1$, $0 < \lambda_\ell < 1$ and $\sum \lambda_\ell = 1$, is chosen such that the discretization points are located at fixed fractions of the total time duration [5]. We will define \hat{t} to be composed of all the discretization points, hence

$$\hat{t} = [t_1, t_2, \dots, t_N]^T$$

and thus, the NLO variables can then be expressed as a function of our state and control variables

$$\mathbf{x}(\Omega(\cdot), \Phi(\cdot), \hat{t}) = [\Omega(t_1), \Phi(t_1), \Omega(t_2), \Phi(t_2), \dots, \Omega(t_N), \Phi(t_N)]^T,$$

where $\mathbf{x}(\Omega(\cdot), \Phi(\cdot), \hat{t}) \in \mathbb{R}^{2N}$. However, one should note that $\mathbf{x}(\Omega(\cdot), \Phi(\cdot), \hat{t})$ is composed of $2N$ sub-vectors that when expanded have the following form:

$$\mathbf{x}(\Omega(\cdot), \Phi(\cdot), \hat{t}) = \begin{bmatrix} \Omega(t_1) \\ \Phi(t_1) \\ \vdots \\ \Omega(t_N) \\ \Phi(t_N) \end{bmatrix} = \begin{bmatrix} M_x(t_1, s_1) \\ M_y(t_1, s_1) \\ M_z(t_1, s_1) \\ \vdots \\ M_x(t_1, s_n) \\ M_y(t_1, s_n) \\ M_z(t_1, s_n) \\ G(t_1) \\ b_x(t_1) \\ b_y(t_1) \\ W(t_1) \\ \vdots \\ M_x(t_N, s_1) \\ M_y(t_N, s_1) \\ M_z(t_N, s_1) \\ \vdots \\ M_x(t_N, s_n) \\ M_y(t_N, s_n) \\ M_z(t_N, s_n) \\ G(t_N) \\ b_x(t_N) \\ b_y(t_N) \\ W(t_N) \end{bmatrix},$$

which has the dimension $N(3n + 4)$. For simplicity, we will let $\Omega_j \equiv \Omega(t_j)$,

thus

$$\Omega_j = [M_x(t_j, s_1), M_y(t_j, s_1), M_z(t_j, s_1), \dots, M_x(t_j, s_n), M_y(t_j, s_n), M_z(t_j, s_n), G(t_j)]^T$$

and similarly,

$$\Phi_j \equiv \Phi(t_j) = [b_x(t_j), b_y(t_j), W(t_j)]^T,$$

for $j = 1, \dots, N$. Finally, we will set $\mathbf{x} \equiv \mathbf{x}(\Omega(\cdot), \Phi(\cdot), \hat{t})$ and therefore, \mathbf{x} now becomes

$$\mathbf{x} = [\Omega_1, \Phi_1, \Omega_2, \Phi_2, \dots, \Omega_N, \Phi_N]^T. \quad (2.5.15)$$

Also, the function from (2.5.1), which presently involves the discretized time intervals, t_j , has the simplified notation

$$f_j \equiv f(\Omega_j, \Phi_j) \equiv f(\Omega(t_j), \Phi(t_j)). \quad (2.5.16)$$

Using this notation, the ODEs defined in f_j are then approximated by setting finite differences equal to zero, hence

$$\begin{aligned} 0 &= \Omega_{\ell+1} - \Omega_\ell - \frac{h_\ell}{2}[f_{\ell+1} + f_\ell] \\ &= \Omega_{\ell+1} - \Omega_\ell - \frac{1}{2}\lambda_\ell[t_N f_{\ell+1}] - \frac{1}{2}\lambda_\ell[t_N f_\ell], \end{aligned} \quad (2.5.17)$$

which will be a component of our NLO constraints [2]. In anticipation of writing each equation from $\ell = 1, \dots, N - 1$ in a simplified matrix form, the nonlinear relationships are isolated in the vector

$$\mathbf{p}(\mathbf{x}) = \begin{bmatrix} t_N f_1 \\ t_N f_2 \\ \vdots \\ t_N f_N \end{bmatrix}, \quad (2.5.18)$$

(2.5.14) can now be expressed in terms of \mathbf{x} . Finally, the NLO problem can be stated as follows:

$$\begin{aligned} \min \quad & J(\mathbf{x}), \\ \text{s.t.} \quad & c_L \leq \begin{bmatrix} \mathbf{Ax} + \mathbf{Bp}(\mathbf{x}) \\ \mathbf{x} \end{bmatrix} \leq c_U, \\ & \psi_L \leq \mathbf{x} \leq \psi_U. \end{aligned} \tag{2.5.24}$$

A number of different NLO algorithms can be employed to solve (2.5.24), the VERSE optimal control problem, which we will discuss in the next chapter. However, regardless of the NLO method, this would find the optimal solution to the objective function (2.2.2) while satisfying constraints (2.2.3S) – (2.2.9S).

Chapter 3

Nonlinear Optimization

Among the various Nonlinear Optimization methods, Sequential Quadratic Programming is a powerful optimization technique utilized by many competitive software systems. In this chapter we will begin with a background of certain unconstrained optimization methods leading to the development of Sequential Quadratic Optimization, implicated within many optimization software packages.

3.1 Unconstrained Optimization

Before attempting to solve the nonlinear, constrained, VERSE optimization problem, we will begin with an overview of unconstrained optimization. Consider the following minimization problem,

$$\min J(\mathbf{x}),$$

where $\mathbf{x} \in \mathbb{R}^N$ and $J : \mathbb{R}^N \rightarrow \mathbb{R}$. Many algorithms have been developed to solve unconstrained optimization problems that can be separated into two general categories, direct search methods or derivative-based methods. Direct search methods are composed of algorithms that are only based on function value comparison. Typically, these methods are costly and are used for prob-

lems in which the function, J , is possibly neither smooth nor continuous. Frequently used direct search methods include the pattern search Nelder-Mead simplex algorithm, the Hooke-Jeeves cyclic coordinate search and other Derivative Free Optimization (DFO) algorithms. Derivative-based methods, on the other hand, are used for problems in which J is smooth and thus, the derivatives are easy to calculate. For the VERSE optimization problem, a derivative-based method will be utilized. First we will begin by outlining derivative-based methods in unconstrained optimization. Two classical approaches to such an optimization problem are *Line Search* based methods and *Trust Region* methods.

3.1.1 Line Search Method

First, we will describe the general outline of Line Search Based Algorithms, then we will discuss its steps in more detail. A line search based algorithm can be outlined as follows:

Input: $\varepsilon > 0$, the accuracy parameter and \mathbf{x}^0 , a given feasible starting point.

- Step 1 Initialization:** Set $\mathbf{x} = \mathbf{x}^0$ and $q = 0$.
- Step 2 Compute Search Direction:** Determine a non-zero vector σ^q representing a descending feasible search direction from \mathbf{x}^q ;
If no such direction exists, **Stop**, \mathbf{x}^q is a local optimal solution.
- Step 3 Compute Line search:** Find a positive step length μ^q , such that $\mu^q = \min_{\mu} J(\mathbf{x}^q + \mu\sigma^q)$.
- Step 4 Update:** Let $\mathbf{x}^{q+1} = \mathbf{x}^q + \mu^q\sigma^q$ and $q = q + 1$.
- Step 5 Test for stopping criteria:** If the stopping criteria is satisfied
Stop;
Else return to **Step 2**.

An explanation of the important steps in the algorithm will now be given. For more information on stopping criteria, convergence and algorithms with regards to specific search directions, the reader can refer to [1], [12] or [13].

Search Direction

Depending on the search direction method, a first-order and/or second-order partial derivative of the function J is evaluated during successive iterations to determine the search direction. Therefore, J has to be at least once continuously differentiable and for higher order methods like the Newton method, it is necessary that J be twice continuously differentiable. Most models are based on a Taylor series approximation and for our purposes we will exploit two well known search directions, the *Gradient Direction* and the *Newton Direction*.

A. Gradient Direction

The first-order Taylor expansion of the function J around \mathbf{x}^q gives

$$\tilde{J}(\mathbf{x}^q + \sigma^q) = J(\mathbf{x}^q) + (\sigma^q)^T \nabla J(\mathbf{x}^q). \quad (3.1.1)$$

The Gradient method, which is also known as the *Steepest Descent*, utilizes the search direction σ^q , such that $\tilde{J}(\mathbf{x}^q + \sigma^q)$ is minimized, while the length of σ^q is normalized to $\|\sigma^q\| = \|\nabla J(\mathbf{x}^q)\|$. Therefore, we have

$$\min_{\|\sigma^q\| = \|\nabla J(\mathbf{x}^q)\|} (\sigma^q)^T \nabla J(\mathbf{x}^q)$$

and hence, the search direction becomes [1]

$$\sigma^q = -\nabla J(\mathbf{x}^q). \quad (3.1.2)$$

B. Newton Direction

The second-order Taylor expansion of J around \mathbf{x}^q gives

$$\tilde{J}(\mathbf{x}^q + \sigma^q) = J(\mathbf{x}^q) + (\sigma^q)^T \nabla J(\mathbf{x}^q) + \frac{1}{2}(\sigma^q)^T \nabla^2 J(\mathbf{x}^q) \sigma^q. \quad (3.1.3)$$

The Newton method also searches for a direction σ^q that minimizes $\tilde{J}(x^q + \sigma^q)$. By taking the first derivative of the right hand side with respect to σ^q and setting the left hand side of (3.1.3) equal to zero, we have

$$\begin{aligned} 0 &= \nabla J(x^q) + \nabla^2 J(x^q)\sigma^q \\ -\nabla^2 J(x^q)\sigma^q &= \nabla J(x^q). \end{aligned}$$

Thus, the search direction becomes [1]

$$\sigma^q = -(\nabla^2 J(x^q))^{-1}\nabla J(x^q). \quad (3.1.4)$$

It is obvious to see that in order to have a solution, the Hessian, $\nabla^2 J(x^q)$, must be non-singular. In addition, the Hessian must also be positive definite, that will ensure we have a descent direction.

Line Search

After computing a search direction σ^q for a given iteration, a line search is then used to decide how far to move along this search direction. A line search is a subroutine in our algorithm that chooses a step size such that the new iterate has a better value with respect to the objective. The iteration is given by

$$x^{q+1} = x^q + \mu^q \sigma^q,$$

where the positive scalar value, μ^q , is known as the step length. Although it is not always possible, the ideal choice for μ^q is the minimizer of

$$m(\mu) = J(x^q + \mu\sigma^q), \quad (3.1.5)$$

where $m(\cdot)$ is a univariate function [12]. Also note that usually the optimal μ is positive, otherwise we would be at an optimal solution.

A univariate function is *unimodal* in $[\mu^{\min}, \mu^{\max}]$ if there exists a unique $\mu^* \in [\mu^{\min}, \mu^{\max}]$ such that for any $\mu_1, \mu_2 \in [\mu^{\min}, \mu^{\max}]$ where $\mu_1 < \mu_2$, we have that:

$$\begin{aligned} &\text{if } \mu_2 < \mu^*, \text{ then } J(\mu_1) > J(\mu_2) \text{ (slope down);} \\ &\text{and if } \mu_1 > \mu^*, \text{ then } J(\mu_2) > J(\mu_1) \text{ (slope up).} \end{aligned}$$

This implies that the function is non-increasing on the interval $[\mu^{\min}, \mu^*]$ and non-decreasing on the interval $[\mu^*, \mu^{\max}]$ [12]. The Golden Section search and Quadratic Interpolation are two examples of many methods that find the minimum, μ^* , of a univariate function, granted it is unimodal. These methods, however, are expensive since they usually require many function evaluations to find μ^* , and also, functions are rarely unimodal. Thus, more practical strategies include *inexact* line searches. Such searches identify a step length that achieves adequate reduction of the function, J , at minimal cost. The Goldstein-Armijo principle is a well known rule that defines an acceptable step length range that is often used in *inexact* line searches [12]. When computing a step length μ^q of $J(x^q + \mu^q \sigma^q)$, the new point should decrease J proportional to the tangent line. Thus, we use the following bound,

$$0 < -\mu^q \varphi_1 J'(x^q) \sigma^q \leq J(x^q) - J(x^{q+1}) \leq -\mu^q \varphi_2 J'(x^q) \sigma^q \quad (3.1.6)$$

where $0 < \varphi_1 \leq \varphi_2 < 1$, $\mu^q > 0$ and $J'(x^q) \sigma^q < 0$. The upper and lower bounds in the above Goldstein-Armijo principle ensure μ^q is a good choice by specifying a sufficient decrease in the objective function while exploiting the maximal allowed step length [19]. Generally, φ_1 is quite small, however, typical values of φ_2 depend on the search direction that is used.

3.1.2 Trust Region Method

The Trust Region method is similar to Newton's method, however, the model is minimized without the line search, but under the restriction of some Trust

Region radius. Also, the algorithm does not require that the Hessian is invertible nor positive definite, instead these properties are enforced with the addition of another matrix. As in Newton's method, the approximation is derived from a second order Taylor expansion that we will define as $h(\mathbf{x}^q + \sigma^q)$, and hence

$$\tilde{J}(\mathbf{x}^q + \sigma^q) = J(\mathbf{x}^q) + \nabla J(\mathbf{x}^q)^T \sigma^q + \frac{1}{2}(\sigma^q)^T \nabla^2 J(\mathbf{x}^q) \sigma^q. \quad (3.1.7)$$

Instead of the line search computed in Section 3.1.1, when the search direction, σ^q , is determined, the Trust Region method restricts the step size by,

$$\|\sigma^q\| \leq \Delta_q, \quad (3.1.8)$$

where $\Delta_q > 0$ and is known as the Trust Region radius [1]. This enables the step size to be finitely bounded, hence, the steps taken from \mathbf{x}^q to \mathbf{x}^{q+1} ,

$$\mathbf{x}^{q+1} = \mathbf{x}^q + \sigma^q,$$

have a maximal length of Δ_q . To deal with the length of Δ_q between each iteration, its value is adjusted based on the relation between the approximation

$$J(\mathbf{x}^q) + \nabla J(\mathbf{x}^q)^T \sigma^q + \frac{1}{2}(\sigma^q)^T \nabla^2 J(\mathbf{x}^q) \sigma^q$$

and the objective function $J(\mathbf{x}^q + \sigma^q)$ values. If the relation is "strong", then the model can be "trusted" and further, Δ_q can be increased. However, if the relation is "weak", then Δ_q is decreased or it remains unchanged depending on the level of "weakness." Exactly how this entity is determined will be discussed when describing the algorithm.

Thus, at each iteration, q , the minimizer of the approximation

$$J(\mathbf{x}^q) + \nabla J(\mathbf{x}^q)^T \sigma^q + \frac{1}{2}(\sigma^q)^T \nabla^2 J(\mathbf{x}^q) \sigma^q$$

must be established over the trust region. There are many approaches to find such minimizers, however, for the purposes of this paper, it will be sufficient to describe one of the strategies found in [12]. Thus, the problem becomes the minimization of

$$\nabla J(\mathbf{x}^q)^T \sigma^q + \frac{1}{2}(\sigma^q)^T (\nabla^2 J(\mathbf{x}^q) + \tilde{\rho} \mathbf{I}) \sigma^q, \quad (3.1.9)$$

where \mathbf{I} is the identity matrix and $\tilde{\rho} \in \mathbb{R}$ can be considered to be the the Lagrange multiplier for constraint (3.1.8) whose purpose is to ensure that $(\nabla^2 J(\mathbf{x}^q) + \tilde{\rho} \mathbf{I})$ is positive definite. Also, since $\tilde{\rho}$ is the Lagrange multiplier for $\|\sigma^q\| \leq \Delta_q$, we have

$$\tilde{\rho}(\Delta_q - \|\sigma^q\|) = 0,$$

and later, one can verify that the larger Δ_q becomes, the smaller $\tilde{\rho}$ is, and visa versa. The search direction, σ^q , of the Trust Region method is derived after solving the equation,

$$(\nabla^2 J(\mathbf{x}^q) + \tilde{\rho} \mathbf{I}) \sigma^q = -\nabla J(\mathbf{x}^q). \quad (3.1.10)$$

One should note that when we solve for the search direction,

$$\sigma^q = -(\nabla^2 J(\mathbf{x}^q) + \tilde{\rho} \mathbf{I})^{-1} \nabla J(\mathbf{x}^q), \quad (3.1.11)$$

it becomes apparent that as $\tilde{\rho} \rightarrow \infty$, then we approach some multiple of the Gradient step, however, if $\tilde{\rho} = 0$, then a Newton step is taken, as illustrated in Figure 3.1. This process is controlled by comparing the predicted decrease in the approximation, $J(\mathbf{x}^q) - h(\mathbf{x}^q + \sigma^q)$, and the actual decrease in the objective function, $J(\mathbf{x}^q) - J(\mathbf{x}^q + \sigma^q)$. Hence, the ratio

$$v = \frac{J(\mathbf{x}^q) - J(\mathbf{x}^q + \sigma^q)}{J(\mathbf{x}^q) - h(\mathbf{x}^q + \sigma^q)} = \frac{\text{Actual Decrease}}{\text{Predicted Decrease}},$$

is calculated at each iteration and if v is large enough, then the trust region is expanded in the next iteration. However, if v is sufficiently small, then the

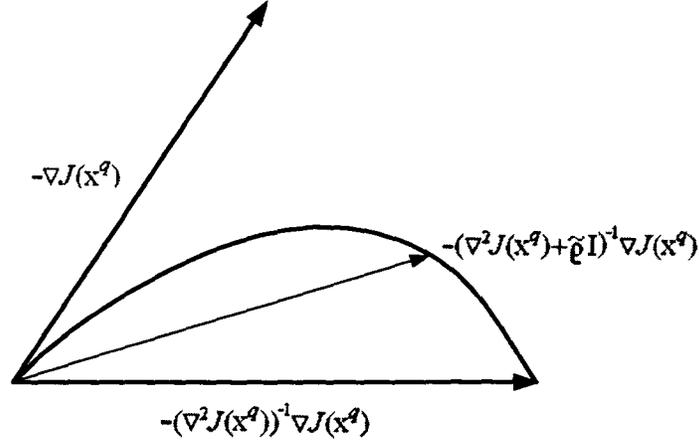


Figure 3.1: Depending on the values of $\tilde{\rho}$, the Trust Region method can take a Gradient step, $-\nabla J(\mathbf{x}^q)$ or Newton step, $-(\nabla^2 J(\mathbf{x}^q))^{-1} \nabla J(\mathbf{x}^q)$.

trust region is reduced; otherwise, the trust region radius remains the same.

Trust-Region Algorithm

The Trust-Region algorithm can be controlled by using the Trust-Region radius, Δ_q , or using the Lagrange multiplier, $\tilde{\rho}$. Using a $\tilde{\rho}$ update, the Trust-Region method would be as follows:

Input: The starting point \mathbf{x}^0 , Lagrange multiplier $\tilde{\rho} > 0$, and the constants,

$$0 < \eta_1 < \eta_2 < 1 \text{ and } 0 < \gamma_1 < 1 < \gamma_2.$$

For $q = 0, 1, \dots$

If \mathbf{x}^q satisfies the stopping criteria, **Stop**.

Else, solve for σ^q

$$\min h(\mathbf{x}^q + \sigma^q) = J(\mathbf{x}^q) + \nabla J(\mathbf{x}^q)^T \sigma^q + \frac{1}{2} (\sigma^q)^T (\nabla^2 J(\mathbf{x}^q) + \tilde{\rho} \mathbf{I}) \sigma^q$$

and compute,

$$v_q = \frac{J(\mathbf{x}^q) - J(\mathbf{x}^q + \sigma^q)}{J(\mathbf{x}^q) - h(\mathbf{x}^q + \sigma^q)}.$$

Then update $\tilde{\rho}$,

If $v_q < \eta_1$ then, $\mathbf{x}^{q+1} = \mathbf{x}^k$ and $\tilde{\rho} = \gamma_2 \tilde{\rho}$ (step failed);

If $\eta_1 \leq v_q \leq \eta_2$ then, $\mathbf{x}^{q+1} = \mathbf{x}^q + \sigma^q$ and $\tilde{\rho} = \tilde{\rho}$ (step as predicted);

If $v_q > \eta_2$ then, $\mathbf{x}^{q+1} = \mathbf{x}^q + \sigma^q$ and $\tilde{\rho} = \gamma_1 \tilde{\rho}$ (step is very good).

The parameters, γ_1 and γ_2 , decide the amount to increase or decrease the trust region. As mentioned, an increase in $\tilde{\rho}$ leads to a decrease in Δ_q . Typical starting values for the variables you set are: $\eta_1 = \frac{1}{4}$, $\eta_2 = \frac{3}{4}$, $\gamma_1 = \frac{1}{2}$, and $\gamma_2 = 2$ [12].

3.2 Equality-Constrained Optimization

The preceding section has addressed unconstrained optimization problems, in this section, we complicate matters slightly by introducing constraints to our discussion. In particular, let $J : \mathbb{R}^N \rightarrow \mathbb{R}$, $c_i : \mathbb{R}^N \rightarrow \mathbb{R}$ for $i = 1, \dots, m \leq N$, and consider how to find the N dimensional vector $\mathbf{x}^T = (x_1, \dots, x_N)$ to minimize

$$J(\mathbf{x}) \tag{3.2.1}$$

subject to the $m \leq N$ constraints

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} c_1(\mathbf{x}) \\ \vdots \\ c_m(\mathbf{x}) \end{bmatrix} = 0. \tag{3.2.2}$$

The classical approach is to define the *Lagrangian*

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\rho}) = J(\mathbf{x}) - \boldsymbol{\rho}^T \mathbf{c}(\mathbf{x}) = J(\mathbf{x}) - \sum_{i=1}^m \rho_i c_i(\mathbf{x}), \tag{3.2.3}$$

where $\boldsymbol{\rho}^T = (\rho_1, \dots, \rho_m)$ is the *Lagrange multiplier*. For the point \mathbf{x}^* to be an optimum, the derivatives of the Lagrangian with respect to both \mathbf{x} and $\boldsymbol{\rho}$, must be zero, hence

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\rho}^*) = 0 \tag{3.2.4}$$

and

$$\nabla_{\rho} \mathcal{L}(\mathbf{x}^*, \rho^*) = 0. \quad (3.2.5)$$

Here, the gradient of \mathcal{L} with respect to \mathbf{x} is

$$\nabla_{\mathbf{x}} \mathcal{L} = \nabla_{\mathbf{x}} J(\mathbf{x}) - \sum_{i=1}^m \rho_i \nabla_{\mathbf{x}} c_i(\mathbf{x}). \quad (3.2.6)$$

If we let $\mathbf{g}(\mathbf{x}) = \nabla_{\mathbf{x}} J(\mathbf{x})$ and define the *Jacobian* of the constraints by

$$\mathbf{G}(\mathbf{x}) \equiv \frac{\partial \mathbf{c}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial c_1}{\partial x_1} & \frac{\partial c_1}{\partial x_2} & \cdots & \frac{\partial c_1}{\partial x_N} \\ \frac{\partial c_2}{\partial x_1} & \frac{\partial c_2}{\partial x_2} & \cdots & \frac{\partial c_2}{\partial x_N} \\ \vdots & & \ddots & \vdots \\ \frac{\partial c_m}{\partial x_1} & \frac{\partial c_m}{\partial x_2} & \cdots & \frac{\partial c_m}{\partial x_N} \end{bmatrix}, \quad (3.2.7)$$

then (3.2.6) can be simplified to

$$\nabla_{\mathbf{x}} \mathcal{L} = \mathbf{g}(\mathbf{x}) - (\mathbf{G}(\mathbf{x}))^T \boldsymbol{\rho}. \quad (3.2.8)$$

The gradient of \mathcal{L} with respect to $\boldsymbol{\rho}$ is

$$\nabla_{\rho} \mathcal{L} = -\mathbf{c}(\mathbf{x}). \quad (3.2.9)$$

Conditions (3.2.4) and (3.2.5), however, do not distinguish whether or not a point is a minimum or maximum, therefore, we require conditions on the curvature of the function. Let us define the second order derivative of the Lagrangian as

$$\mathbf{L} = \nabla_{\mathbf{xx}}^2 \mathcal{L} = \nabla_{\mathbf{xx}}^2 J(\mathbf{x}) - \sum_{i=1}^m \rho_i \nabla_{\mathbf{xx}}^2 c_i(\mathbf{x}). \quad (3.2.10)$$

Then a sufficient condition for defining a minimum is that

$$\mathbf{y}^T \mathbf{L} \mathbf{y} > 0 \quad (3.2.11)$$

for any vector \mathbf{y} in the constraint tangent space. Hence, if $\mathbf{G}(\mathbf{x})\mathbf{y} = 0$, then the vector \mathbf{y} is tangent to the constraints at \mathbf{x} and further, is in the constraint

tangent space.

Let us now apply Newton's method to find the values of (\mathbf{x}, ϱ) such that the necessary conditions (3.2.4) and (3.2.5) are satisfied. First we will simplify our notation by letting $\mathbf{g} \equiv \mathbf{g}(\mathbf{x})$ and $\mathbf{G} \equiv \mathbf{G}(\mathbf{x})$. As well, we will define $\mathbf{H} \equiv \mathbf{H}(\mathbf{x})$ as the *Hessian* of the constraints and σ as a search direction for a step $\bar{\mathbf{x}} = \mathbf{x} + \sigma$. Taking a Taylor series expansion analogous to the one found in (3.1.3), the quadratic objective is the minimization of

$$\mathbf{g}^T \sigma + \frac{1}{2} \sigma^T \mathbf{H} \sigma \quad (3.2.12)$$

subject to the constraints

$$\mathbf{G} \sigma = -\mathbf{c}(\mathbf{x}). \quad (3.2.13)$$

Thus, after constructing the Lagrangian, as in (3.2.3), and making the substitution $\sigma = \bar{\mathbf{x}} - \mathbf{x}$, the functions in (3.2.4) and (3.2.5) become

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \varrho^*) = 0 = \mathbf{g} - \mathbf{G}^T \varrho + \mathbf{L}(\bar{\mathbf{x}} - \mathbf{x}) - \mathbf{G}^T (\bar{\varrho} - \varrho), \quad (3.2.14)$$

$$\nabla_{\varrho} \mathcal{L}(\mathbf{x}^*, \varrho^*) = 0 = -\mathbf{c}(\mathbf{x}) - \mathbf{G}(\bar{\mathbf{x}} - \mathbf{x}), \quad (3.2.15)$$

where $\bar{\varrho}$ is the vector of the Lagrange multipliers at the new point and the Hessian is approximated by the second order derivative of the Lagrangian [4]. Equation (3.2.14) can be simplified to

$$0 = \mathbf{g} - \mathbf{G}^T \bar{\varrho} + \mathbf{L}(\bar{\mathbf{x}} - \mathbf{x}). \quad (3.2.16)$$

Here, (3.2.15) and (3.2.16) lead to a set of equations analogous to (3.2.13) called the Karush-Kuhn-Tucker (KKT) system, namely

$$\begin{bmatrix} \mathbf{L} & \mathbf{G}^T \\ \mathbf{G} & 0 \end{bmatrix} \begin{bmatrix} -\sigma \\ \bar{\varrho} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{c}(\mathbf{x}) \end{bmatrix}. \quad (3.2.17)$$

It is important to note that the quadratic approximation is made to the Lagrangian (3.2.3), and it is not simply the objective function $J(\mathbf{x})$.

3.3 Inequality-Constrained Optimization

A NonLinear Optimization (NLO) problem can be stated as finding the vector \mathbf{x} that minimizes the *objective function*

$$J(\mathbf{x}), \tag{3.3.1}$$

subject to the *constraints*

$$c_L \leq \mathbf{c}(\mathbf{x}) \leq c_U, \tag{3.3.2}$$

and the simple *bounds*

$$\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U. \tag{3.3.3}$$

Applications of NLO usually involve a large number of variables and constraints, as in the VERSE pulse problem. For a solution, \mathbf{x}^* , to satisfy the Karush-Kuhn-Tucker (KKT) conditions, the following conditions must be satisfied.

KKT Conditions:

- i. \mathbf{x}^* is feasible, hence (3.3.2) and (3.3.3) are satisfied;
- ii. There exists Lagrange multipliers $\varrho^T = (\varrho_1, \dots, \varrho_m)$ and $\lambda^T = (\lambda_1, \dots, \lambda_m)$ such that

$$\mathbf{g} = \mathbf{G}^T \varrho + \lambda, \tag{3.3.4}$$

where $\mathbf{g} = \nabla_{\mathbf{x}} J(\mathbf{x})$ is the gradient vector and \mathbf{G} is the Jacobian matrix (3.2.7);

- iii. The Lagrange multipliers

$$\begin{aligned} \varrho_i, \lambda_i &\geq 0 & \forall i \in \mathcal{A}(c_L, \mathbf{x}_L), \\ \varrho_i, \lambda_i &\leq 0 & \forall i \in \mathcal{A}(c_U, \mathbf{x}_U), \\ \varrho_i, \lambda_i &= 0 & \forall i \notin \mathcal{A}(\mathbf{c}(\mathbf{x}), \mathbf{x}), \end{aligned}$$

where $\mathcal{A}(\cdot)$ is the set of active constraints or variables [1].

Sequential Quadratic Programming (SQP) is one of the most powerful method of solving NLO problems. It is especially efficient in finding the solution to discretized optimal control problems. The algorithm finds a feasible objective value by solving a sequence of quadratic subproblems. The fundamental premise of the approach is to approximate the nonlinear constraint functions by a linear model, and the objective function by a quadratic model. First we will discuss the Quadratic Optimization (QO) subproblem, followed by a definition of the merit function, and then show how it is applied within the SQP algorithm.

3.3.1 Quadratic Optimization

A primary element of the SQP algorithm is our ability to solve quadratic subproblems efficiently. Solutions of the QO subproblem are used to define new estimates for the variables according to the formula

$$\bar{\mathbf{x}} = \mathbf{x} + \mu\boldsymbol{\sigma}, \tag{3.3.5}$$

where the vector $\boldsymbol{\sigma}$ is the search direction and the scalar μ determines the step length. The search direction $\boldsymbol{\sigma}$ is found by minimizing the quadratic objective function

$$\mathbf{g}^T\boldsymbol{\sigma} + \frac{1}{2}\boldsymbol{\sigma}^T\mathbf{H}\boldsymbol{\sigma} \tag{3.3.6}$$

subject to the linear constraints

$$\tilde{\mathbf{c}}_L \leq \begin{bmatrix} \mathbf{G}\boldsymbol{\sigma} \\ \boldsymbol{\sigma} \end{bmatrix} \leq \tilde{\mathbf{c}}_U, \tag{3.3.7}$$

where \mathbf{H} is a positive definite approximation of the constraint Hessian matrix, which will be described later; for more details on can also consult [10].

In addition, the constraints involve the Jacobian matrix \mathbf{G} and the search direction, which are bounded by

$$\tilde{\mathbf{c}}_L = \begin{bmatrix} c_L - \mathbf{c}(\mathbf{x}) \\ \mathbf{x}_L - \mathbf{x} \end{bmatrix}, \quad \text{and} \quad \tilde{\mathbf{c}}_U = \begin{bmatrix} c_U - \mathbf{c}(\mathbf{x}) \\ \mathbf{x}_U - \mathbf{x} \end{bmatrix}. \quad (3.3.8)$$

We will now describe the *merit function* and Hessian approximation techniques that are used to solve the quadratic optimization problem. In the next chapter we will explain how the vector \mathbf{g} , and matrices, \mathbf{H} and \mathbf{G} , are configured with respect to our VERSE pulse problem.

Merit Function

When a quadratic program is used to approximate a constrained nonlinear problem, it is necessary to adjust the step length, μ , using a merit function. The merit function combines the constraints and the objective function such that a relatively large step length is taken that produces sufficient reduction between iterates. The merit function is defined as

$$M(\mathbf{x}, \varrho, \lambda, \mathbf{u}, \mathbf{v}) = J(\mathbf{x}) - \varrho^T(\mathbf{c} - \mathbf{u}) - \lambda^T(\mathbf{x} - \mathbf{v}) + \frac{1}{2}(\mathbf{c} - \mathbf{u})^T \mathbf{Q}(\mathbf{c} - \mathbf{u}) + \frac{1}{2}(\mathbf{x} - \mathbf{v})^T \mathbf{R}(\mathbf{x} - \mathbf{v}), \quad (3.3.9)$$

where $\mathbf{c} \equiv \mathbf{c}(\mathbf{x})$ for simplicity. In addition, the diagonal penalty matrices, \mathbf{Q} and \mathbf{R} , have diagonal elements denoted by $\mathbf{Q}_{ii} = q_i$ and $\mathbf{R}_{ii} = r_i$. The target values, \mathbf{u} and \mathbf{v} , for the merit function are defined at the beginning of the step, such that we have

$$\mathbf{u}_i := \begin{cases} c_{Li} & \text{if } c_{Li} > c_i - \varrho_i/q_i, \\ c_i - \varrho_i/q_i & \text{if } c_{Li} \leq c_i - \varrho_i/q_i \leq c_{Ui}, \\ c_{Ui} & \text{if } c_i - \varrho_i/q_i > c_{Ui}, \end{cases} \quad (3.3.10)$$

and

$$\mathbf{v}_i := \begin{cases} \mathbf{x}_{Li} & \text{if } \mathbf{x}_{Li} > \mathbf{x}_i - \lambda_i/r_i, \\ \mathbf{x}_i - \lambda_i/r_i & \text{if } \mathbf{x}_{Li} \leq \mathbf{x}_i - \lambda_i/r_i \leq \mathbf{x}_{Ui}, \\ \mathbf{x}_{Ui} & \text{if } \mathbf{x}_i - \lambda_i/r_i > \mathbf{x}_{Ui}, \end{cases} \quad (3.3.11)$$

where $i = 1, \dots, m$ [4]. In each step of the SQP algorithm, first (3.3.6) – (3.3.8) are solved to get the search direction, σ . Then the predicted constraint variables, $\hat{\mathbf{u}}$, are derived, where

$$\hat{\mathbf{u}} = \mathbf{G}\sigma + \mathbf{c}. \quad (3.3.12)$$

Using this expression (3.3.12) we define the constraint vector displacement by

$$\Delta\mathbf{u} \equiv \hat{\mathbf{u}} - \mathbf{u} = \mathbf{G}\sigma + (\mathbf{c} - \mathbf{u}). \quad (3.3.13)$$

A similar technique defines the search direction for the \mathbf{v} variables

$$\Delta\mathbf{v} \equiv \hat{\mathbf{v}} - \mathbf{v} = \sigma + (\mathbf{x} - \mathbf{v}). \quad (3.3.14)$$

An estimate of the Lagrange multiplier is necessary since, in general, equation (3.3.4) is not always satisfied. There are a number of different methods available to find Lagrange multiplier estimates, $\hat{\rho}$ and $\hat{\lambda}$, one of which, from [15], involves optimizing the following least-squares problem:

$$(\hat{\rho}, \hat{\lambda}) = \arg \min_{\rho, \lambda} \|\mathbf{G}^T \rho + \lambda - \mathbf{g}\|^2.$$

It is then possible to define the displacements for the multipliers as,

$$\Delta\rho = \hat{\rho} - \rho, \quad (3.3.15)$$

and

$$\Delta\lambda = \hat{\lambda} - \lambda. \quad (3.3.16)$$

Thus, the search direction is given by

$$[\sigma^T, \Delta\rho^T, \Delta\lambda^T, \Delta\mathbf{u}^T, \Delta\mathbf{v}^T]^T.$$

It is then necessary to update the penalty parameters \mathbf{Q} and \mathbf{R} to ensure the search direction is decreasing. In [4, 14], it is shown that the convergence of the method assumes that the penalty parameters are chosen such that

$$M'_0 \leq -\frac{1}{2}\sigma^T \mathbf{H}\sigma, \quad (3.3.17)$$

where M'_0 denotes the directional derivative of the merit function (3.3.9) with respect to the step length μ evaluated at $\mu = 0$. To satisfy inequality (3.3.17), it is necessary to utilize the vector Ξ , whose elements have the following characteristics:

$$\Xi_j = \begin{cases} q_j - \xi_0 & \text{if } 1 < j \leq m, \\ r_{j-m} - \xi_0 & \text{if } m < j \leq (m + N), \end{cases} \quad (3.3.18)$$

where $\xi_0 > 0$ is a strictly positive constant known as a “threshold.” Since (3.3.17) provides a condition for the $(m + N)$ penalty parameters, we make the choice unique by minimizing the norm $\|\Xi\|_2$. This yields

$$\Xi = \mathbf{a}(\mathbf{a}^T \mathbf{a})^{-1} \zeta, \quad (3.3.19)$$

where

$$\mathbf{a}_j = \begin{cases} (c_j - \mathbf{u}_j)^2 & \text{if } 1 < j \leq m, \\ (\mathbf{x}_{j-m} - \mathbf{v}_{j-m})^2 & \text{if } m < j \leq (m + N), \end{cases} \quad (3.3.20)$$

and

$$\begin{aligned} \zeta = & -\frac{1}{2} \sigma^T \mathbf{H} \sigma + \hat{\rho}^T \Delta \mathbf{u} + \hat{\lambda}^T \Delta \mathbf{v} - 2(\Delta \rho)^T (\mathbf{c} - \mathbf{u}) \\ & - 2(\Delta \lambda)^T (\mathbf{x} - \mathbf{v}) - \xi_0 (\mathbf{c} - \mathbf{u})^T (\mathbf{c} - \mathbf{u}) - \xi_0 (\mathbf{x} - \mathbf{v})^T (\mathbf{x} - \mathbf{v}). \end{aligned} \quad (3.3.21)$$

Typically, the threshold parameter ξ_0 , is set to machine precision and in essence, the penalty parameters are chosen to be as small as possible with the descent condition (3.3.17).

Using the Goldstein-Armijo principle in (3.1.6), a line search $\bar{\mu}$ that minimizes the merit function (3.3.9) is determined. Then a new point is derived according to

$$\begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\rho} \\ \bar{\lambda} \\ \bar{\mathbf{u}} \\ \bar{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \rho \\ \lambda \\ \mathbf{u} \\ \mathbf{v} \end{bmatrix} + \bar{\mu} \begin{bmatrix} \sigma \\ \Delta \rho \\ \Delta \lambda \\ \Delta \mathbf{u} \\ \Delta \mathbf{v} \end{bmatrix}. \quad (3.3.22)$$

Thus, the SQP algorithm to solve the NLO given by (3.3.1) – (3.3.3) can be summarized in the following algorithm:

Obtain \mathbf{u} and \mathbf{v} from (3.3.10) – (3.3.11), and fix ρ , λ .

- Step 1** Solve the QO subproblem (3.3.6) – (3.3.7) to obtain σ .
- Step 2** Determine $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ and find $\Delta\mathbf{u}, \Delta\mathbf{v}$ using (3.3.13) – (3.3.14).
- Step 3** Update the displacements $\Delta\rho, \Delta\lambda$ in (3.3.15) – (3.3.16) using $\hat{\rho}, \hat{\lambda}$.
- Step 4** Update \mathbf{Q} and \mathbf{R} to ensure the search direction $[\sigma^T, \Delta\rho^T, \Delta\lambda^T, \Delta\mathbf{u}^T, \Delta\mathbf{v}^T]^T$ is decreasing.
- Step 5** Compute the Goldstein-Armijo line search to minimize the merit function (3.3.9) that give the step length, $\bar{\mu}$.
- Step 6** Compute the new point $[\bar{\mathbf{x}}, \bar{\rho}, \bar{\lambda}, \bar{\mathbf{u}}, \bar{\mathbf{v}}]^T$ using (3.3.22).
- Step 7** If the stopping criteria is satisfied **Stop**;
 Else return to **Step 1**.

A stopping criteria would also include that if no descent search direction is found in **Step 4**, then a local optimal solution is found.

The merit function is also used to update a fundamental parameter in developing a positive definite approximation of the Hessian, \mathbf{H} in (3.3.6), which will be described later. We will outline the two quantities involved in such a process and their purpose will become apparent when describing the algorithm. For simplicity, we will set $M(\mathbf{x}, \rho, \lambda, \mathbf{u}, \mathbf{v}) \equiv M$, and for an iteration, the algorithms “actual reduction” is represented as

$$\nu_1 = M - \bar{M}, \tag{3.3.23}$$

where M is the current value of the merit function and \bar{M} is the value after one step of the algorithm. The algorithms “predicted reduction” of the step is

$$\nu_2 = M - \widehat{M} = -M'_0 - \frac{1}{2}\sigma^T \mathbf{H} \sigma, \tag{3.3.24}$$

where \widehat{M} is the predicted value of the merit function made during the minimization process, described in the SQP algorithm on page 59. For different merit functions one can consult [21].

Hessian Approximation

A positive definite Hessian ensures that the optimal solution of the QO problem is unique and allows \mathbf{Q} and \mathbf{R} to satisfy the descent condition (3.3.17). First we construct the second order derivative of the Lagrangian,

$$\mathbf{L} = \nabla_{\mathbf{xx}}^2 J(\mathbf{x}) - \sum_i \varrho_i \nabla_{\mathbf{xx}}^2 c_i(\mathbf{x}). \quad (3.3.25)$$

However, the problem is that \mathbf{L} is generally not positive definite, consequently a modified matrix \mathbf{H} is used, where

$$\mathbf{H} = \mathbf{L} + \iota(|\widehat{\mathbf{H}}_L| + 1)\mathbf{I}. \quad (3.3.26)$$

Here, ι is the Levenberg parameter that is chosen such that $0 \leq \iota \leq 1$ and is normalized using the Gerschgorin bound for the most negative eigenvalue of \mathbf{L} , where

$$\widehat{\mathbf{H}}_L = \min_{1 \leq i \leq N} \left\{ h_{ii} - \sum_{i \neq j}^N |h_{ij}| \right\} \quad (3.3.27)$$

and h_{ij} is the nonzero elements of \mathbf{L} , see [3]. The proper choice of the Levenberg parameter, ι , may greatly effect the performance of the SQP algorithm. For instance, quadratic convergence can be obtained if $\iota = 0$. However, if $\iota = 1$, in order to guarantee a positive definite Hessian, a gradient direction is used and subsequently, the algorithm converges linearly [13]. Thus, a strategy similar to that used in the Trust-Region method is employed to choose the Lavenberg parameter between successive iterations. By utilizing parameters ν_1 and ν_2 , such a strategy would maintain a positive definite Hessian while attempting to have strong convergence. In addition, the positive definiteness

of the projected Hessian is inferred by the inertia of the related KKT matrix, the abbreviated 2×2 matrix in (3.2.17). However, in order to describe the inertia of the KKT matrix and how it can be utilized, equations (3.3.6) – (3.3.7) must be reformulated.

3.3.2 Sequential Quadratic Programming

The QO formulations can now be incorporated in an SQP (Sequential Quadratic Programming) framework. First it is necessary to state the QO subproblem in the following matrix form:

Compute $\tilde{\sigma}$ to minimize

$$\tilde{\mathbf{g}}^T \tilde{\sigma} + \frac{1}{2} \tilde{\sigma}^T \tilde{\mathbf{H}} \tilde{\sigma}, \quad (3.3.28)$$

subject to the constraints

$$\tilde{\mathbf{G}} \tilde{\sigma} = \tilde{\mathbf{c}}, \quad (3.3.29)$$

and simple bounds

$$\tilde{\sigma}_L \leq \tilde{\sigma} \leq \tilde{\sigma}_U. \quad (3.3.30)$$

Since this formulation involves only simple bounds (3.3.30) and equality constraints (3.3.29), the tilde notation was introduced to denote the transformation of the original variables in (3.3.6) – (3.3.7) to (3.3.28) – (3.3.30). This is accomplished by including slack variables in (3.3.29) and bound vectors in (3.3.30). The search direction can be computed by solving the KKT system similar to (3.2.17), which in this case is

$$\begin{bmatrix} \tilde{\mathbf{H}} & \tilde{\mathbf{G}}^T \\ \tilde{\mathbf{G}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} -\tilde{\sigma} \\ \tilde{\varrho} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{g}} \\ \mathbf{0} \end{bmatrix}, \quad (3.3.31)$$

where we assume that the current iterate is feasible or starts at a feasible point, i.e., $\tilde{\mathbf{c}} = 0$. Thus, the KKT matrix in (3.3.31) is defined as

$$\mathbf{K} = \begin{bmatrix} \tilde{\mathbf{H}} & \tilde{\mathbf{G}}^T \\ \tilde{\mathbf{G}} & 0 \end{bmatrix}, \quad (3.3.32)$$

and now we can show how the Levenberg parameter ι is adjusted from iteration to iteration by using \mathbf{K} . Inertia is defined as the number of positive, negative, and zero eigenvalues of a matrix [6]. The inertia of the KKT matrix can be used to infer the positive definiteness of the Hessian matrix, shown in [16]. The inertia of \mathbf{K} is easily computed as a byproduct of the symmetric indefinite factorization by counting the number of positive and negative elements in the diagonal matrix. Using results from [16], the Hessian will be positive definite if the inertia of \mathbf{K} is

$$In(\mathbf{K}) = (N, m, 0), \quad (3.3.33)$$

where N is the number of rows in $\tilde{\mathbf{H}}$ and m is the number of rows in $\tilde{\mathbf{G}}$. Basically the philosophy is to reduce the Levenberg parameter when the predicted reduction in the merit function agrees with the actual reduction, and increase the parameter when the agreement is poor. The process is accelerated by making the change in ι proportional to the rate of change in the gradient of the Lagrangian. To be more precise, we compute ν_1 and ν_2 at iteration q from (3.3.23) – (3.3.24). Then find the rate of change in the norm of the gradient of the Lagrangian

$$\nu_3 = \frac{\|\vartheta^q\|_\infty}{\|\vartheta^{q-1}\|_\infty}, \quad (3.3.34)$$

where the error in the gradient of the Lagrangian is

$$\vartheta = \mathbf{g} - \mathbf{G}^T \varrho - \lambda. \quad (3.3.35)$$

Then, if $\nu_1 \leq 0.25\nu_2$, the actual behavior is much worse than predicted, so the step will be towards the gradient by setting $\iota^{q+1} = \min(2\iota^q, 1)$. On the

other hand, if $\nu_1 \geq 0.75\nu_2$, then the actual behavior is sufficiently close to predicted one, so the search direction will change towards a Newton direction by setting $\iota^{q+1} = \iota^q \min(0.5, \nu_3)$. It is important to note that this strategy, similar to the one employed by the NLO solver used for our VERSE problem, does not necessarily ensure that the Hessian is positive definite but makes an intelligent prediction. In fact, it may be necessary to increase ι^{q+1} whenever the inertia of the KKT matrix is incorrect, as will be done in the algorithm of the next section.

The SQP Algorithm

We can now summarize the steps in the algorithm. Thus, for any iteration q , at the point \mathbf{x} , the minimization proceeds as follows:

Step 1 *Gradient Evaluation:*

- (a) Evaluate the error in the gradient of the Lagrangian from (3.3.35);
- (b) Terminate if the KKT conditions are satisfied;
- (c) Compute \mathbf{L} from (3.3.25);
- (d) If this is the first iteration go to **Step 2**; otherwise
 - i. Compute the rate of change in the norm of the gradient of the Lagrangian from (3.3.34);
 - ii. Complete the *Levenberg modification* using (3.3.23) – (3.3.24) and (3.3.34):
 - If $\nu_1 \leq 0.25\nu_2$, then $\iota^{q+1} = \min(2\iota^q, 1)$;
 - If $\nu_1 \geq 0.75\nu_2$, then $\iota^{q+1} = \iota^q \min(0.5, \nu_3)$.

Step 2 *Search Direction:* Construct the optimization search direction;

- (a) Compute \mathbf{H} from (3.3.26);
- (b) Compute σ by solving the QO subproblem (3.3.6) – (3.3.7);
- (c) *Inertia Control:* if the inertia of \mathbf{K} is incorrect and
 - i. If $\iota < 1$, then increase ι and return to **Step 2(a)**;

- ii. If $\iota = 1$ and $\mathbf{H} \neq \mathbf{I}$, then set $\iota = 0$, $\mathbf{H} = \mathbf{I}$ and return to **Step 2(a)**;
- iii. If $\mathbf{H} = \mathbf{I}$, then the QO constraints are locally inconsistent, terminate the algorithm;
- (d) Compute $\Delta \mathbf{u}$ and $\Delta \mathbf{v}$ from (3.3.13) and (3.3.14);
- (e) Compute $\Delta \rho$ and $\Delta \lambda$ from (3.3.15) and (3.3.16);
- (f) Compute penalty parameters to satisfy (3.3.17);
- (g) Initialize $\bar{\mu} = 1$.

Step 3 Prediction:

- (a) Compute the predicted point for the variables, multipliers, and slacks from (3.3.22);
- (b) Evaluate the constraints at the predicted point, $\bar{\mathbf{c}} = \mathbf{c}(\bar{\mathbf{x}})$.

Step 4 Line Search: Evaluate the merit function $M(\bar{\mathbf{x}}, \bar{\rho}, \bar{\lambda}, \bar{\mathbf{u}}, \bar{\mathbf{v}}) = \bar{M}$ and

- (a) If the merit function value \bar{M} is “sufficiently” less than M , then $\bar{\mathbf{x}}$ is an improved point, terminate the line search and go to **Step 5**;
- (b) Else, decrease the step length, $\bar{\mu}$, to reduce M and return to **Step 3**.

Step 5 Update: Update all quantities, set $q = q + 1$;

- (a) Compute the actual reduction from (3.3.23);
- (b) Compute the predicted reduction form (3.3.24);
- (c) Return to **Step 1**.

Note that the algorithm consists of an outer loop, **Steps 1 – 5**, which potentially finds the optimal solution, and an inner loop, **Steps 3 – 4**, which approves the sufficient reduction of the the merit function.

The steps outlined describe the fundamental elements of the SQP optimization process, however, several points deserve additional clarification. We

address some of them and for an even more detailed explanation the reader should consult [4]. First note that in **Step 3(a)** the algorithm requires a line search in the direction defined by (3.3.22) with the step length $\bar{\mu}$ adjusted to reduce the merit function. Adjusting the value of the step length $\bar{\mu}$, as required in **Step 4(b)**, is accomplished by using a line search procedure that constructs a quadratic or cubic model of the merit function. The reduction is ensured to be “sufficient” by using the Goldstein-Armijo principle.

In addition, in order to evaluate \mathbf{L} from (3.3.25), an estimate of the Lagrangian multipliers is needed. The values obtained by solving the QO problem with $\mathbf{H} = \mathbf{I}$ are used for the first iteration, and thereafter, the values $\bar{\rho}$ from (3.3.22) are used. Note that, at the very first iteration, two QO sub-problems are solved, the first is to compute the first order multiplier estimates and the second is to compute the step. Furthermore, for the first iteration, the multipliers search directions are $\Delta\rho = 0$ and $\Delta\lambda = 0$, so that the multipliers will be initialized to the QO estimates $\bar{\rho} = \rho = \hat{\rho}$ and $\bar{\lambda} = \lambda = \hat{\lambda}$. Also, the multipliers are reset in a similar fashion if the QO constraints are locally inconsistent, in **Step 2(c)iii**. Thus, the Lavenberg parameter, ι , in (3.3.26) and the penalty parameters, q_i and r_i , in (3.3.9) are initialized to zero. Consequently, the merit function is initially simply the Lagrangian.

Algorithm’s Strategies

The basic algorithm described above has been implemented in FORTRAN as a part of the SOCS library and is documented in [5]. Sparse Optimal Control Software or SOCS, is the NLO package utilized for solving the VERSE optimization problem. In the software, the preceding approach is referred to as strategy M, since the iterates follow a path from the initial point to the solution. However, in practice it may be desirable and/or more efficient to first locate a feasible point. Consequently, the software provides three differ-

ent algorithm strategies, namely:

- M** Minimize. Starting with x^0 , solve a sequence of quadratic programs until a solution \mathbf{x}^* is found.
- FM** Find a Feasible point and then Minimize. Starting with x^0 , solve a sequence of quadratic programs to locate a feasible point, \mathbf{x}^f , then starting from \mathbf{x}^f , solve a sequence of quadratic programs until a solution \mathbf{x}^* is found.
- FME** Find a Feasible point and then Minimize subject to Equalities. Starting with x^0 , solve a sequence of quadratic programs to locate a feasible point, \mathbf{x}^f , then starting from \mathbf{x}^f , solve a sequence of quadratic programs while maintaining feasible equalities until a solution \mathbf{x}^* is found.

Additional details on the FM and FME strategy can be found in [3]. The software employs the FM strategy as a default since computational experience suggests that it is more robust and efficient than the other two strategies.

In addition, as with many NLO solvers, a number of things can go wrong that will prevent the software from finding an optimal solution. To highlight a few problems one might encounter, the software may not find an optimum because:

1. The linear constraints (3.3.7) are inconsistent (i.e., have no solution);
2. The Jacobian matrix \mathbf{G} is rank deficient;
3. The linear constraints (3.3.7) are redundant or extraneous, which can correspond to Lagrange multipliers that are zero at the solution;
4. The quadratic objective (3.3.6) is unbounded in the null space of the active constraints.

Unfortunately, because the QO is a subproblem within the overall NLO, it is not always obvious how to determine the cause of the difficulty. In particular, the QO subproblem may have problems locally simply because the quadratic/linear model does not approximate the nonlinear behavior accurately. On the other hand, the QO subproblem may also have difficulties because the original NLO problem is inherently ill-posed. Regardless of the cause of the difficulties in the QO subproblem, the overall algorithm behavior can be significantly impacted. Thus, much thought must be put into the model and how the NLO is constructed before implementation.

Chapter 4

Implementation

The implementation issues surrounding the VERSE model and algorithms involved in the SQP computation can now be addressed. The implementation was based on the formulations and equations that were detailed in Chapter 2.

4.1 SQP Implementation

Sparse Optimal Control Software (SOCS) from The Boeing Company was used to solve the nonlinear VERSE pulse problem. By utilizing an SQP algorithm, SOCS is currently one of the most competitive NLO solvers in the world. It was developed and is currently used at Boeing to tackle many nonlinear optimization problems. A Trapezoidal method similar to the one outlined in Section 4 of Chapter 2 was implemented in our optimization algorithm using SOCS.

From Chapter 3, we are left to explain how the vector \mathbf{g} , and matrices \mathbf{H} and \mathbf{G} , are configured with respect to our VERSE NLO model. The reader should refer to Section 4 of Chapter 2 and notice that from (2.5.15) our state and control variables were isolated in the vector

$$\mathbf{x} = [\Omega_1, \Phi_1, \Omega_2, \Phi_2, \dots, \Omega_N, \Phi_N]^T,$$

where $\Omega_j \equiv \Omega(t_j)$ and $\Phi_j \equiv \Phi(t_j)$ for $j = 1, \dots, N$ discretized time intervals. In order to define the gradient, it is necessary to create a subvector that separates \mathbf{x} by its discretized time intervals, hence

$$\mathbf{x}_j = [\Omega_j, \Phi_j]^T.$$

Thus, for $j = 1, \dots, N$ and $h = 1, \dots, N$, the gradient \mathbf{g} becomes

$$\mathbf{g} = \frac{\partial f_j}{\partial \mathbf{x}_h} = \begin{bmatrix} \nabla_{\mathbf{x}_1} f_1 & \nabla_{\mathbf{x}_2} f_1 & \dots & \nabla_{\mathbf{x}_N} f_1 \\ \nabla_{\mathbf{x}_1} f_2 & \nabla_{\mathbf{x}_2} f_2 & & \\ \vdots & & \ddots & \\ \nabla_{\mathbf{x}_1} f_N & \dots & & \nabla_{\mathbf{x}_N} f_N \end{bmatrix}, \quad (4.1.1)$$

where f_j is the simplified notation from (2.5.16) [4]. Using the Trapezoidal approximation defined in (2.5.17), the Jacobian matrix for the resulting NLO problem is defined by

$$\mathbf{G} \equiv \frac{\partial \mathbf{c}_j}{\partial \mathbf{x}_h}, \quad (4.1.2)$$

where \mathbf{c} is the abbreviated form of the constraints denoted in (3.3.2) of Chapter 3 [4]. Furthermore, we will make the appropriate constraint substitution from (2.5.22) and therefore, we are left with

$$\mathbf{G} \equiv \frac{\partial \mathbf{c}_j}{\partial \mathbf{x}_h} = \mathbf{A} + \mathbf{B} \frac{\partial \mathbf{p}_j}{\partial \mathbf{x}_h}, \quad (4.1.3)$$

where matrices \mathbf{A} , \mathbf{B} and vector $\mathbf{p} = \mathbf{p}(\mathbf{x})$ correspond to (2.5.20), (2.5.21) and (2.5.18), respectively. One can observe from (2.5.18) that

$$\frac{\partial \mathbf{p}_j}{\partial \mathbf{x}_h} = t_N \mathbf{g}, \quad (4.1.4)$$

and thus the Jacobian becomes

$$\mathbf{G} = \mathbf{A} + t_N \mathbf{B} \mathbf{g}. \quad (4.1.5)$$

Finally, the Hessian, \mathbf{H} , is constructed by what was outlined in (3.3.25) – (3.3.27) of Chapter 3. To start we derive an approximation of the Hessian, which is actually the second order derivative of the Lagrangian

$$\mathbf{L} = \nabla_{\mathbf{xx}}^2 J(\mathbf{x}) - \sum_{j=1}^N \varrho_j \nabla_{\mathbf{xx}}^2 f_j, \quad (4.1.6)$$

where $\nabla_{\mathbf{xx}}^2 J(\mathbf{x}) = \frac{\partial^2 J(\mathbf{x})}{\partial \mathbf{x}_j \partial \mathbf{x}_h}$, $\nabla_{\mathbf{xx}}^2 f_j = \frac{\partial^2 f_j}{\partial \mathbf{x}_j \partial \mathbf{x}_h}$ and ϱ_j is the Lagrange multiplier for $j = 1, \dots, N$. Then, by computing **Step 1** and **Step 2** within the SQP algorithm described on pages 59 – 60, the Levenberg parameter in equations (3.3.26) – (3.3.27) is updated and the Hessian is constructed. As detailed, the Hessian is first approximated by $\mathbf{H} \approx \mathbf{L}$, then in the subsequent iterations of the algorithm it is precisely calculated.

An important aspect of this construction now becomes evident; notice that \mathbf{g} , and consequently \mathbf{G} and \mathbf{H} , involve the partial derivatives of f with respect to the state and control variables, all evaluated at some time discretization. In particular, we have the following nonzero structure

$$\frac{\partial f_j}{\partial \mathbf{x}_h} = \left(\frac{\partial f_j}{\partial \Omega_h}, \frac{\partial f_j}{\partial \Phi_h} \right). \quad (4.1.7)$$

The nonzero pattern defined by (4.1.7) appears repeatedly in \mathbf{g} and \mathbf{G} at different time increments and is problem dependent since it is defined by the functional form of the state equations, see (2.5.1) of Chapter 2. There are a number of techniques imposed for specifying the nonzero structure in (4.1.7), however, the approach implemented in SOCS involves numerically constructing the matrix template using random perturbations about random nominal points. For additional information on alternative methods for designing matrix templates one can consult [4].

4.2 Slice Assignment

In the VERSE model constructed in Chapter 2, S was discretized into coordinate positions s_1, s_2, \dots, s_n and partitioned into the sets S_{in} and S_{out} . Furthermore, the coordinate positions in S_{in} were bounded by $[\underline{s}, \bar{s}]$, and similarly, S_{out} was composed of coordinate positions in $[s_1, \underline{s}_l]$ and $[\bar{s}_u, s_n]$. To investigate how coordinate values were assigned to magnetization vectors the reader should re-familiarize themselves with the variables formalized in Section 3. More specifically, $\underline{s}_l = s_{k-1}$, $\underline{s} = s_k$, $\bar{s} = s_{k+\delta}$ and $\bar{s}_u = s_{k+\delta+1}$, for $1 < k \leq k + \delta < n$ and $\delta \geq 0$. Thus, for an application with n slices, each $s_i \in S$ was given a scalar value defined by

$$s_i = \begin{cases} \underline{\beta} + \rho_1(i) & i \leq k - 1, \\ \beta + \rho_2(i) & k \leq i \leq k + \delta, \\ \bar{\beta} + \rho_3(i) & i \geq (k + \delta) + 1, \end{cases} \quad (4.2.1)$$

where $\underline{\beta}, \beta, \bar{\beta} \in \mathbb{R}$. In order to include the off-resonance characteristics found between $(\underline{s}_l, \underline{s})$ and (\bar{s}, \bar{s}_u) , the formula in (4.2.1) is designed such that $\underline{\beta} + \rho_1(k - 1) < \beta \leq \beta + \rho_2(k + \delta) < \bar{\beta}$. Also, $\rho_1(i), \rho_2(i), \rho_3(i)$ are strictly monotonically increasing functions that can uniformly or randomly disperse increments of s_i . As stated in Section 3 of Chapter 2, the subinterval $[\underline{s}, \bar{s}]$ is intended to be centered around 0, and hence, β is chosen such that $\beta + \rho_2(i)$ has the same features for $k \leq i \leq k + \delta$. Also, the values, $\underline{\beta} < 0$ and $\bar{\beta} > 0$, are assigned such that the positions $\underline{\beta} + \rho_1(i)$ for $i \leq k - 1$ and $\bar{\beta} + \rho_3(i)$ for $i \geq (k + \delta) + 1$ are symmetric with respect to each other, as shown in (2.3.1) of Chapter 2. Therefore, using this construction, $\beta + \rho_2(i)$ will contain the values for the magnetization vectors in S_{in} , whereas $\underline{\beta} + \rho_1(i)$ and $\bar{\beta} + \rho_3(i)$ will control the $s_i \in S_{\text{out}}$ values. The initial positions, $\underline{\beta}, \beta, \bar{\beta}$, for this piecewise step function will be chosen depending on how many slices, n , we have and how far we would like to disperse our RF pulse. For example, generally we

would assign values such that $\underline{\beta} \approx s_1$, $\beta \approx s_k$ and $\overline{\beta} \approx s_{(k+\delta)+1}$. Also notice, we can set the distance between $\underline{\beta} + \rho_1(k-1) < \beta$ and $\beta + \rho_2(k+\delta) < \overline{\beta}$ as large as we like, thus, potentially controlling the negative imaging effects described in Section 2, which are experienced by off-resonance magnetization vectors found in these positions.

To implement the function derived in (4.2.1), the values for k , δ , $\underline{\beta}$, β and $\overline{\beta}$ were initialized and the functions ρ_1 , ρ_2 and ρ_3 were defined. The implementation for assigning values to various coordinate positions was as follows:

```

For  $i = 1, \dots, n$ 
  If  $i < k$ , then
     $s_i = \underline{\beta} + \rho_1(i)$ ;
  Else if  $i \geq k$  and  $i \leq k + \delta$ , then
     $s_i = \beta + \rho_2(i)$ ;
  Else
     $s_i = \overline{\beta} + \rho_3(i)$ ;
  End
End

```

Using the values assigned to each coordinate position, s_i , the magnetization vectors were then separated into their respective sets S_{in} and S_{out} , which would later be applied to constraints (2.2.4 S_{in}) and (2.2.4 S_{out}). The algorithm for partitioning the positions s_i into their appropriate sets was:

```

For  $i = 1, \dots, n$ 
  If  $s_i \geq \beta + \rho_2(k)$  and  $s_i \leq \beta + \rho_2(k + \delta)$ , then
     $S(i) = 1$ ;
  Else
     $S(i) = 0$ ;
  End
End

```

The value $S(i) = 1$ corresponds to the coordinate positions in S_{in} , and otherwise, $S(i) = 0$ was the value assigned to $s_i \in S_{\text{out}}$.

After our slices are separated into the sets S_{in} and S_{out} with appropriate values, they are ready to be evaluated within constraints (2.2.3S) – (2.2.9S). Thus, the dynamic variables from constraint (2.2.3S) of Chapter 2, where time has been discretized, was implemented in the following manner:

```

For  $j = 1, \dots, N$ 
   $\hat{n} = 0$ ;
  For  $i = 1, \dots, n$ 
     $\hat{n} = \hat{n} + 1$ ;
     $f(t_j, \hat{n}) = \gamma(-s_i G(t_j) M_y(t_j, s_i) + b_y(t_j) M_z(t_j, s_i));$ 
     $\hat{n} = \hat{n} + 1$ ;
     $f(t_j, \hat{n}) = \gamma(s_i G(t_j) M_x(t_j, s_i) - b_x(t_j) M_z(t_j, s_i));$ 
     $\hat{n} = \hat{n} + 1$ ;
     $f(t_j, \hat{n}) = \gamma(-b_y(t_j) M_x(t_j, s_i) + b_x(t_j) M_y(t_j, s_i));$ 
  End
End

```

The counter variable \hat{n} was introduced because SOCS requires the array $f(t_j, \hat{n})$ to be 1 dimensional, hence, \hat{n} was increased between each computation to account for this prerequisite. The array, $f(t_j, \hat{n})$, was then inserted as an equality constraint into a generic subroutine within SOCS called ODERHS.

Constraints (2.2.4S_{in}) and (2.2.4S_{out}) describe where the magnetization vectors should be at the end of our time interval, $T = t_N$. Here, we have different constraints for variables in S_{in} and S_{out} , hence, the algorithm was as follows:

```

For  $i = 1, \dots, n$ 
  If  $S(i) = 1$ , then
     $\psi(T, i) = [M_0 \sin(\alpha) - M_x(T, s_i)]^2 + [M_y(T, s_i)]^2$ 
     $\quad + [M_0 \cos(\alpha) - M_z(T, s_i)]^2;$ 
     $\psi(T, i) = \sqrt{\psi(T, i)};$ 
  Else
     $\psi(T, i) = [M_x(T, s_i)]^2 + [M_y(T, s_i)]^2 + [M_0 - M_z(T, s_i)]^2;$ 
     $\psi(T, i) = \sqrt{\psi(T, i)};$ 
  End
End

```

Using another subroutine defined within SOCS, ODEPTF, we bound $\psi(T, i)$ by ε_1 if $S(i) = 1$, or ε_2 if $S(i) = 0$. Finally, at $t_1 = 0$, the values of $M_x(0, s_i)$, $M_y(0, s_i)$ and $M_z(0, s_i)$ are easily initialized for $i = 1, \dots, n$ in an input routine, and in the next section we will show how a guess subroutine of the initial solution is efficiently used to estimate $M_x(t_j, s_i)$, $M_y(t_j, s_i)$ and $M_z(t_j, s_i)$ for $t_j \in (t_1, t_N)$.

4.3 Initial Solution

A software's efficiency and robustness in solving a nonlinear problem can be improved by the addition of an intelligent initial guess to the solution of the problem. As mentioned in Chapter 3, even finding a feasible starting point can be difficult with NLO problems, re-emphasizing the importance of our initial solution implemented in SOCS as a guess subroutine. In the VERSE problem, we understand how the magnetization vectors

$$[M_x(t_j, s_1), M_y(t_j, s_1), M_z(t_j, s_1)]^T, \dots, [M_x(t_j, s_n), M_y(t_j, s_n), M_z(t_j, s_n)]^T$$

physically behave in vivo. Also, a generic RF pulse design can be utilized to hypothesize what the values of $G(t_j)$, $b_x(t_j)$ and $b_y(t_j)$ could be. Thus, for these variables we supply a subroutine that defines the initial guess of the

solution to our optimal control problem. Essentially this subroutine evaluates an initial guess for the time dependent function \mathbf{x} . We begin by detailing how the algorithm was coded for the n magnetic moment vectors and follow with the gradient and external magnetization components.

The input for values of the n magnetic moment vectors, $M_x(t_j, s_i)$, $M_y(t_j, s_i)$ and $M_z(t_j, s_i)$, were different depending on whether $s_i \in S_{\text{in}}$ or $s_i \in S_{\text{out}}$. For the vectors that were in S_{in} , our initial guess subroutine was required to tip $[M_x(t_j, S_{\text{in}}), M_y(t_j, S_{\text{in}}), M_z(t_j, S_{\text{in}})]^T$ into the transverse plane by an angle of α . However, if $s_i \in S_{\text{out}}$, then these vectors were required to be in the direction of the static external magnetic field, B_0 , which as mentioned earlier is parallel to the z -axis. Therefore, the algorithm for the initial guess of the vectors $s_i \in S$, over the discretized time interval t_1, \dots, t_N , was as follows:

```

For  $j = 1, \dots, N$ 
  For  $i = 1, \dots, n$ 
    If  $S(i) = 1$ , then
       $M_x(t_j, s_i) = M_0 \sin(\alpha \frac{t_j}{t_N});$ 
       $M_y(t_j, s_i) = 0;$ 
       $M_z(t_j, s_i) = M_0 \cos(\alpha \frac{t_j}{t_N});$ 
    Else
       $M_x(t_j, s_i) = 0;$ 
       $M_y(t_j, s_i) = 0;$ 
       $M_z(t_j, s_i) = M_0;$ 
    End
  End
End

```

where M_0 is the initial magnetization in the z direction. Using this implementation, the magnetic moment vectors in S_{in} would tip into the transverse plane by an angle of α at the end of the time duration t_N , as shown in Figure 4.1. Magnetic moment vectors in S_{out} , however, would align in the z -axis direction with a height of M_0 , which is illustrated in Figure 4.2. To produce

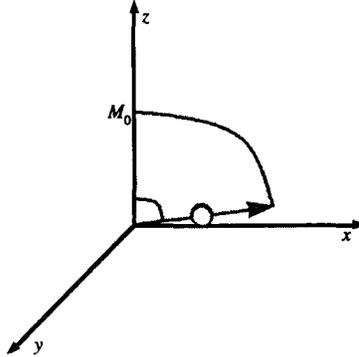


Figure 4.1: The initial solution for magnetic moment vectors in S_{in} that have tipped into the transverse plane by an angle of α .

the high quality final images discussed in Chapter 2, this is exactly what we would like to observe in terms of magnetic moment vectors.

With regards to gradient and external magnetization, a generic RF pulse sequence similar to the one shown in Figure 2.3 of Chapter 2 was used to infer how our initial solution for these variables were modelled. In doing so, Figure 4.3 illustrates how our gradient function, $G(t)$, behaved. There are a few important characteristics of the gradient function worth mentioning. The most significant is that the two areas, highlighted by diagonal lines, are equivalent. Specifically, the area between points g_1 to g_3 is equal to that of g_5 to g_8 , where g_3 is the midpoint of g_2 and g_4 . Another important element of the gradient function is that the slope of the line from g_1 to g_2 , g_7 to g_8 , and the negative slope of g_4 to g_6 , are all equal. The gradient function also requires that the absolute value of the slope of these lines be less than or equal to the maximum slew rate, W_{max} . Finally, the absolute value of the height of the lines, g_2 to g_4 and g_6 to g_7 , are required to be less than or equal to the maximum gradient, G_{max} . To implement such a function, a simple program in Maple, symbolic

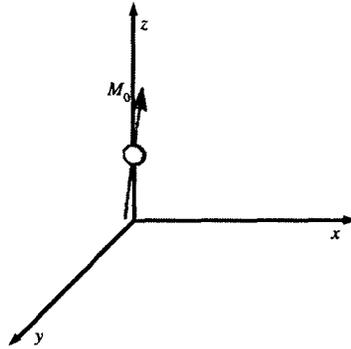


Figure 4.2: The initial solution for magnetic moment vectors in S_{out} .

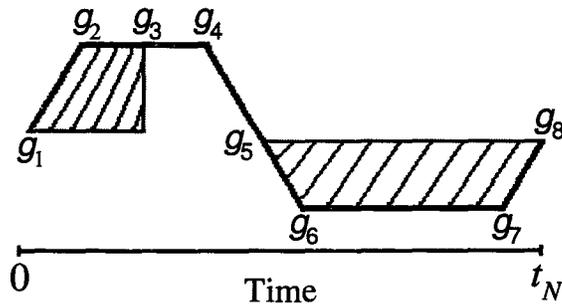


Figure 4.3: The initial solution for the gradient function $G(t)$.

mathematics software produced by Maplesoft, was created that would list the possible values for g_1 to g_8 that satisfy the above criteria. One can easily deduce from Figure 4.3 that the values for g_1, \dots, g_8 will correspond to a specific time discretizations, t_j , within the interval t_1 to t_N . Hence, when the time values of g_1, \dots, g_8 have been determined, including the slope from g_1 to g_2 , and the value of both lines, g_2 to g_4 and g_6 to g_7 , the gradient function can then be implemented. Thus, given the value of the slope from g_1 to g_2 , which will be noted as $m_{1,2}$, the value of the line from g_2 to g_4 , noted as $m_{2,4}$, and g_6 to g_7 , noted as $m_{6,7}$, the algorithm for our gradient function was:

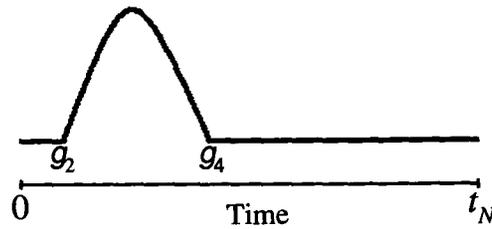


Figure 4.4: The initial solution for the external magnetization $b_y(t)$.

```

For  $j = 1, \dots, N$ 
  If  $j < g_2$ , then
     $G(t_j) = (m_{1,2})j$ ;
  Else if  $g_2 \leq j < g_4$ , then
     $G(t_j) = m_{2,4}$ ;
  Else if  $g_4 \leq j < g_6$ , then
     $G(t_j) = -(m_{1,2})j + (m_{1,2})g_4 + m_{2,4}$ ;
  Else if  $g_6 \leq j < g_7$ , then
     $G(t_j) = m_{6,7}$ ;
  Else
     $G(t_j) = (m_{1,2})j - (m_{1,2})g_7 + m_{6,7}$ ;
  End
End

```

For the values of the external magnetization variables a standard RF pulse is used. Generally, $b_x(t)$ remains constant and is usually zero, however, $b_y(t)$ behaves similar to Figure 4.4. In the illustration, the value of g_2 and g_4 correspond to those which are determined in the gradient function. Thus, the implementation for $b_y(t)$ was as follows:

```

For  $j = 1, \dots, N$ 
  If  $j < g_2$ , then
     $b_y(t_j) = 0;$ 
  Else if  $g_2 \leq j \leq g_4$ , then
     $b_y(t_j) = \sin \left( (j - g_2) \left( \frac{\pi}{g_4 - g_2} \right) \right);$ 
  Else
     $b_y(t_j) = 0;$ 
  End
End

```

Using this model for our guess function, we have created an intelligent approximation of how the variables defined in the vector \mathbf{x} should behave.

4.4 Sparse Optimal Control Software (SOCS)

By applying the subroutines supplied by SOCS that were designed to solve optimal control problems, the solution to our VERSE pulse problem was effectively computed. An outline of the important subroutines and functions that were used in finding the solution to our optimal control problem will be discussed, however, for a description of the defaults or built in functions that SOCS performs, one can refer to [5].

HDSOCS

The subroutine HDSOCS is a powerful optimal control routine provided by SOCS that was called to determine the $3n + 4$ dimensional control and state vectors to minimize

$$J(\mathbf{x}) = \sum_{j=0}^N \left(\int_0^{t_j} w(\Phi(t_j)) dt_j \right), \quad (4.4.1)$$

as shown in Chapter 2. HDSOCS was the central subroutine in the VERSE pulse program, all other routines were eventually passed to HDSOCS in finding the optimal solution.

ODEINP

An important subroutine that must be present in HDSOCS is one that sequentially defines the variables and parameters involved in the optimal control problem. The generic name for this routine, which can be found in the SOCS manual [5], is ODEINP. This subroutine declares the VERSE pulse variables, the number of time discretizations, the number of continuous and discrete user defined functions, the transcription method used to solve the problem, and other such parameters important to locating the optimal solution. As mentioned, to solve the VERSE pulse problem we utilized a Trapezoidal transcription method, which proved to provide the best results when compared to the other methods supplied by SOCS. Also, within this routine the user is required to assign certain values to particular functions defined within the software that ensures the problem is minimized.

ODERHS

HDSOCS also requires a subroutine known as ODERHS that supplies the quadrature function, $w(\Phi(t_j))$, and the dynamic variables implemented in the array $f(t_j, \hat{n})$, shown in Section 2. This subroutine was carefully implemented as it was called many times by SOCS during computation.

ODEPTF

The last important subroutine is ODEPTF, which is responsible for the terminal constraints outlined in the algorithm at the end of Section 2. This subroutine sets the appropriate terminal conditions for vectors in S_{in} and S_{out} to be relayed to HDSOCS.

As with many NLO programs, a subroutine that initializes the data and one that provides the initial solution defined in the preceding section, was

also included in the implementation. The input subroutine that initializes the values for the variables γ , G_{\max} , W_{\max} , M_0 , α , T , ε_1 and ε_2 , also contained the algorithm that assigns values for s_i and separates them into S_{in} and S_{out} . Finally, with regards to the overall functionality of SOCS, although it is one of the most competitive NLO solvers, it is very difficult to use. For example, defining the state and control variables in ODEINP have to be precisely ordered and counted. As well, to set up the quadrature objective, values are given to specific functions in SOCS that depend on how the model is formulated. Hence, careful planing on how to arrange the algorithms in your program is critical. For more detail on other routines and declarations necessary to the functionality of SOCS one can consult the SOCS manual [5].

Chapter 5

Results

In this chapter we present the VERSE pulse computational results derived by SOCS. All numerical experiments were performed on an IBM RS/6000 44P Model 270 Workstation.

5.1 Initializations

The VERSE pulse was precisely designed to improve RF pulse sequences by minimizing SAR and enhancing resolution in MRI. The complex mathematical requirements of the VERSE model may be difficult to satisfy, even simple NLO problems with large numbers of variables can be challenging to solve and threatens many software packages. Thus, when attempting to minimize the objective function in (2.2.2) under the constraints, (2.2.3*S*) – (2.2.9*S*), the number of variables implemented was especially important. Preliminary results were found by implementing the VERSE model using five coordinate positions. This kept the variable count to a minimum of 19, $(3n + 4)$, excluding the independent time variable, t . The number of variables was systematically increased to 49, until software limitations on memory became a factor. Nonetheless, this was a remarkably larger number of variables than anticipated, as it accounted for 15 slices. After experimenting and consulting

the literature, realistic MRI values for the constants were used during each computational simulation. Namely, $\gamma = 42.58 \text{ Hz/mT}$, $G_{\max} = 0.02 \text{ mT/mm}$ and $W_{\max} = 0.2 \text{ mT/mm/ms}$, where Hz is Hertz, mm is millimeters, ms is milliseconds, and mT is millitelsa, the units used to describe the strength of magnetization. The magnetization vectors in S_{in} were fully tipped into the transverse plane, hence, $\alpha = \frac{\pi}{2}$. The magnitude of the initial magnetization vector for each coordinate position had an initial magnetization value of $M_0 = 1$ spin density unit. Finally, we choose $\varepsilon_1, \varepsilon_2 \leq 0.1$, and as the number of variables increased for the problem, the larger the value of ε_1 and ε_2 had to be in order to find a feasible solution.

5.2 Five Slice Results

For the results of the five slice problem, penalty variables and parameters did not need to be introduced, as well, stricter bounds on ε_1 and ε_2 could be imposed. Given that there were only five slices, the middle magnetization vector was tipped into the transverse plane and the others remained in S_{out} . Hence, coordinate position s_3 was in S_{in} and positions s_1, s_2, s_4 and s_5 resided in S_{out} , as shown in Figure 5.1. The exact values for the coordinate positions were as follows:

$$\begin{array}{ccccc} -21 & -20 & 0 & 20 & 21 \\ s_1 & s_2 & s_3 & s_4 & s_5 \end{array},$$

which were in mm. The results for the five coordinate simulation are illustrated in Figures 5.2 – 5.4. Specifically, information on the magnetic vector projection are shown in the graphs found in Figure 5.2. The resulting RF pulse procedure, represented by external magnetization components $b_x(t)$ and $b_y(t)$, and gradient sequence, $G(t)$, is shown in Figures 5.3 – 5.4.

One can observe the precession evident in the graphs of magnetization

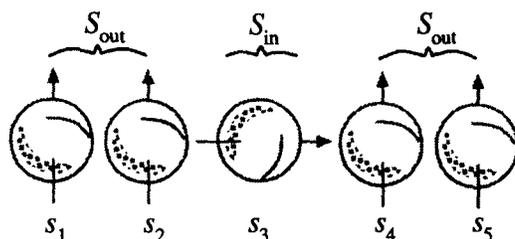


Figure 5.1: The separation of coordinate positions s_i into S_{out} and S_{in} for five magnetization vectors.

vectors s_1, s_2, s_4 and s_5 , those voxels that are in S_{out} . These voxels initially precess with a wide radius, but eventually reduce the size of their orbit. The magnetization vector in S_{in} , namely s_3 , tips into the transverse (x, y) plane very smoothly, without any cusps or peaks. The gradient sequence starts off negative and then ends up positive. It is not a smooth curve since it is composed of many local hills and valleys. Also, the gradient seems to be the opposite of what is used in practical MRI sequences, shown in Figure 4.3, which later proves to be a proficient sequence as we will investigate in the next chapter. The external magnetization components, $b_x(t)$ and $b_y(t)$, are constant and linear, precisely what we optimized for in the objective function. The value of $b_x(t)$ is approximately zero, while $b_y(t)$ has a constant value of 0.0028.

5.3 Fifteen slice Results

The results of the 15 slice problem were more challenging to solve, especially as the distance from \underline{s} to \bar{s} increased. Since there were 15 slices, the three middle magnetization vectors were tipped into the transverse plane to ensure that the symmetric structure of the problem was maintained. Thus, coordinate positions s_7, s_8 and s_9 were in S_{in} , while s_1, s_2, \dots, s_6 and $s_{10}, s_{11}, \dots, s_{15}$

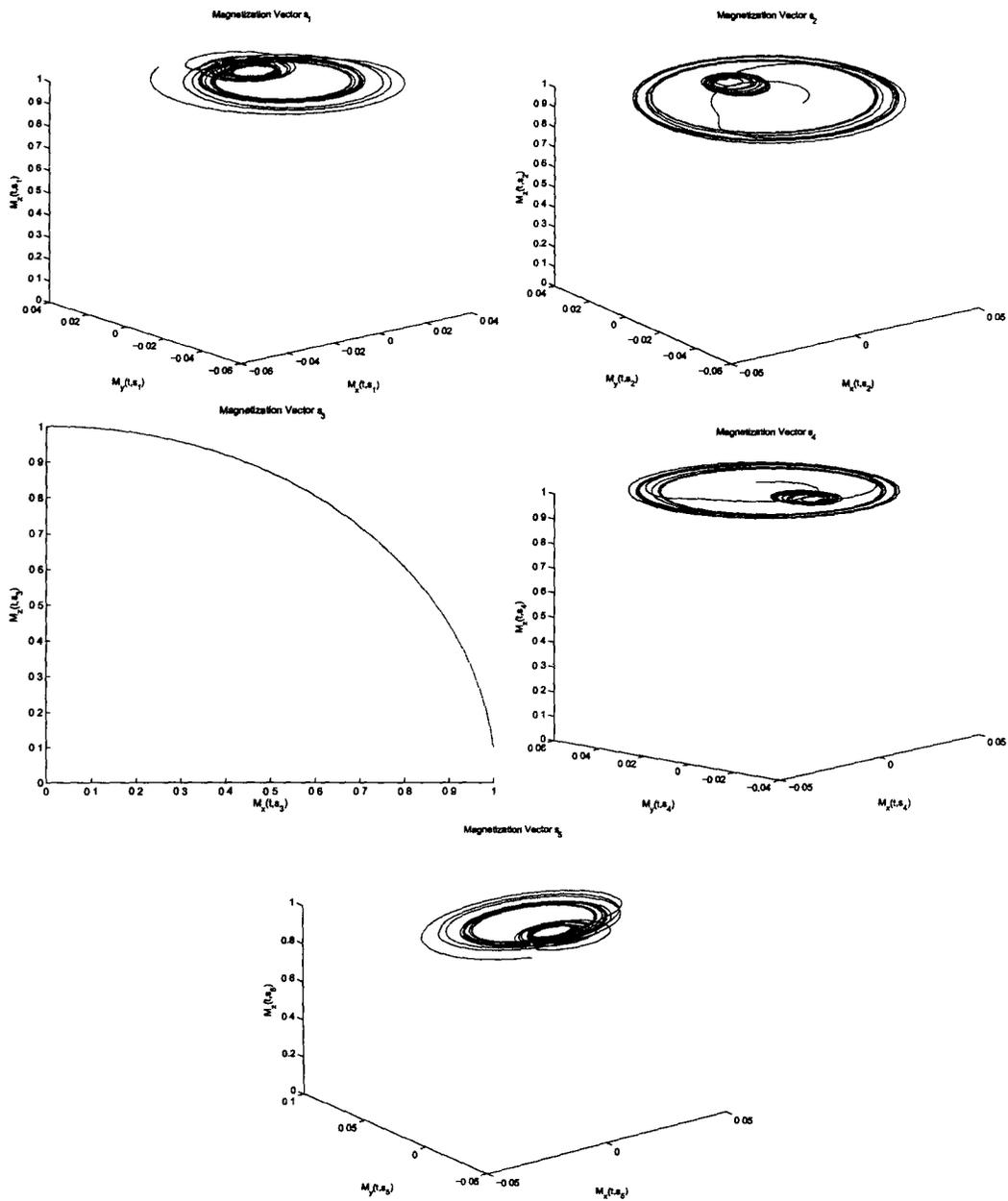


Figure 5.2: From left to right, magnetization vectors corresponding to coordinate positions s_1, s_2, s_3, s_4 and s_5 .

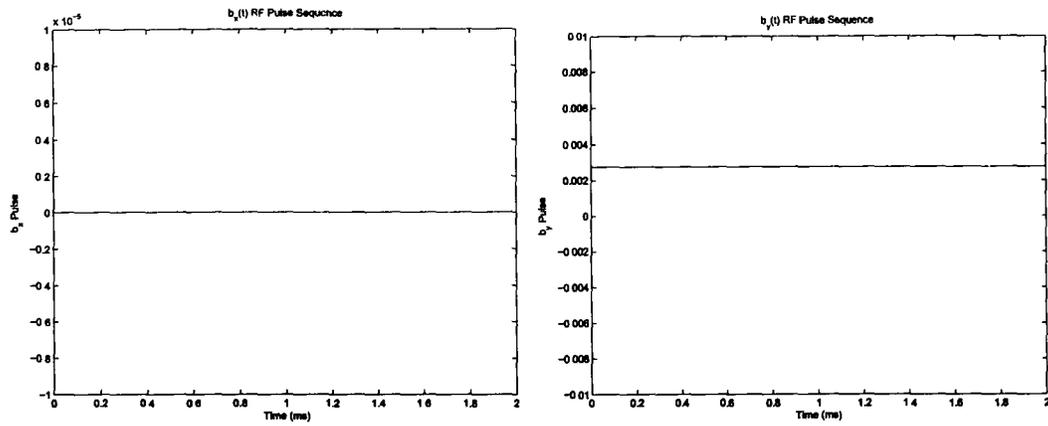


Figure 5.3: External magnetization components $b_x(t)$ and $b_y(t)$, shown respectively.

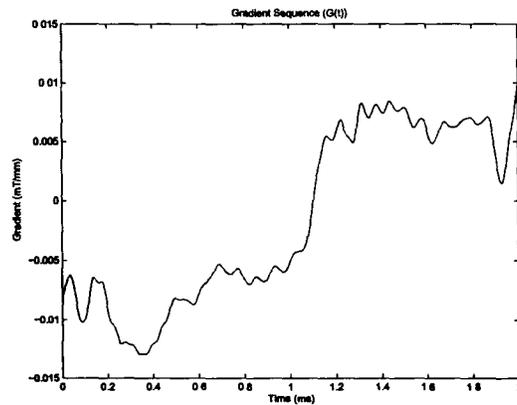


Figure 5.4: Gradient sequence pertaining to magnetization vectors plotted in Figure 5.2.

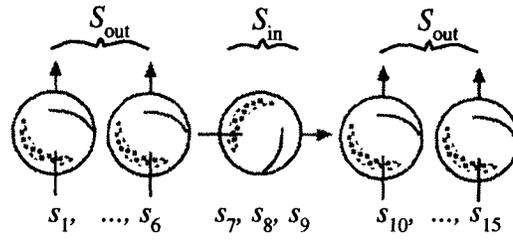


Figure 5.5: The separation of coordinate positions s_i into S_{out} and S_{in} for 15 magnetization vectors.

remained in S_{out} . The arrangement of the coordinate positions is essentially the same as the five slice problem, however, with an increased number of variables incorporated between each slice, as shown in Figure 5.5. The exact values for the coordinate positions were as follows:

- 30	- 28	- 26	- 24	- 22	- 20	- 0.2	0	0.2	20	22	24	26	28	30
s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}

which again were in mm. The results for the 15 slice coordinate simulation is illustrated in Figures 5.6 – 5.10. Information on the magnetic vector projection is shown in the graphs found in Figures 5.6 – 5.8. Specifically, Figures 5.6 and 5.8 correspond to magnetization vectors in S_{out} , and Figure 5.7 refers to the coordinate positions in S_{in} . The resulting RF pulse procedure, represented by external magnetization components $b_x(t)$ and $b_y(t)$, and gradient sequence, $G(t)$, is shown in Figures 5.9 – 5.10.

Again, the precession of the magnetization vectors in S_{out} is evident, this is shown in the graphs of Figures 5.6 and 5.8. The initial point is close to the voxels precession range and at most it takes one full rotation for them to orbit uniformly. The magnetization vectors in Figure 5.7, those s_i that belong to S_{in} , smoothly tip into the transverse plane, again without any cusps or peaks.

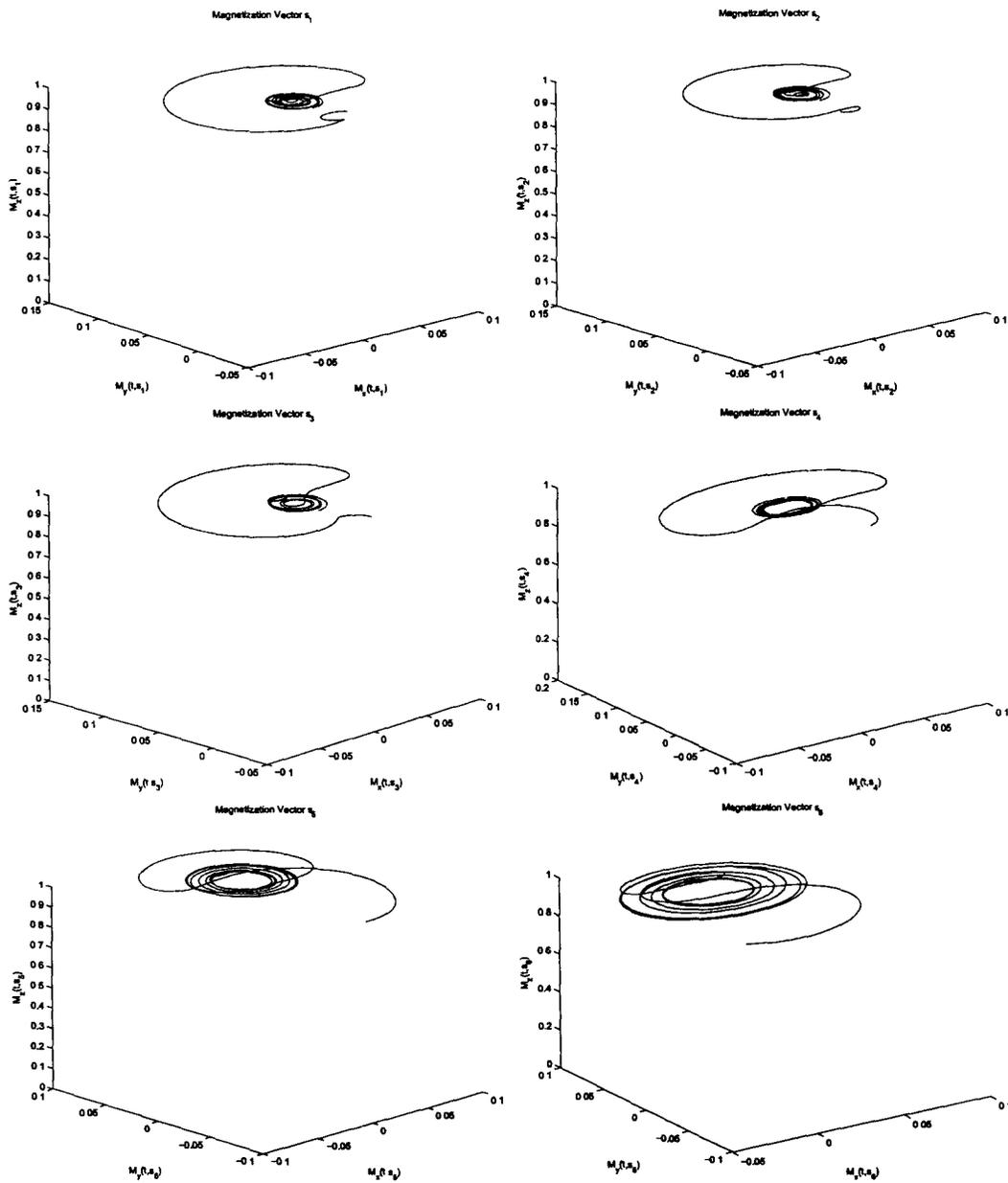


Figure 5.6: From left to right, magnetization vectors corresponding to coordinate positions s_1, s_2, s_3, s_4, s_5 and s_6 .

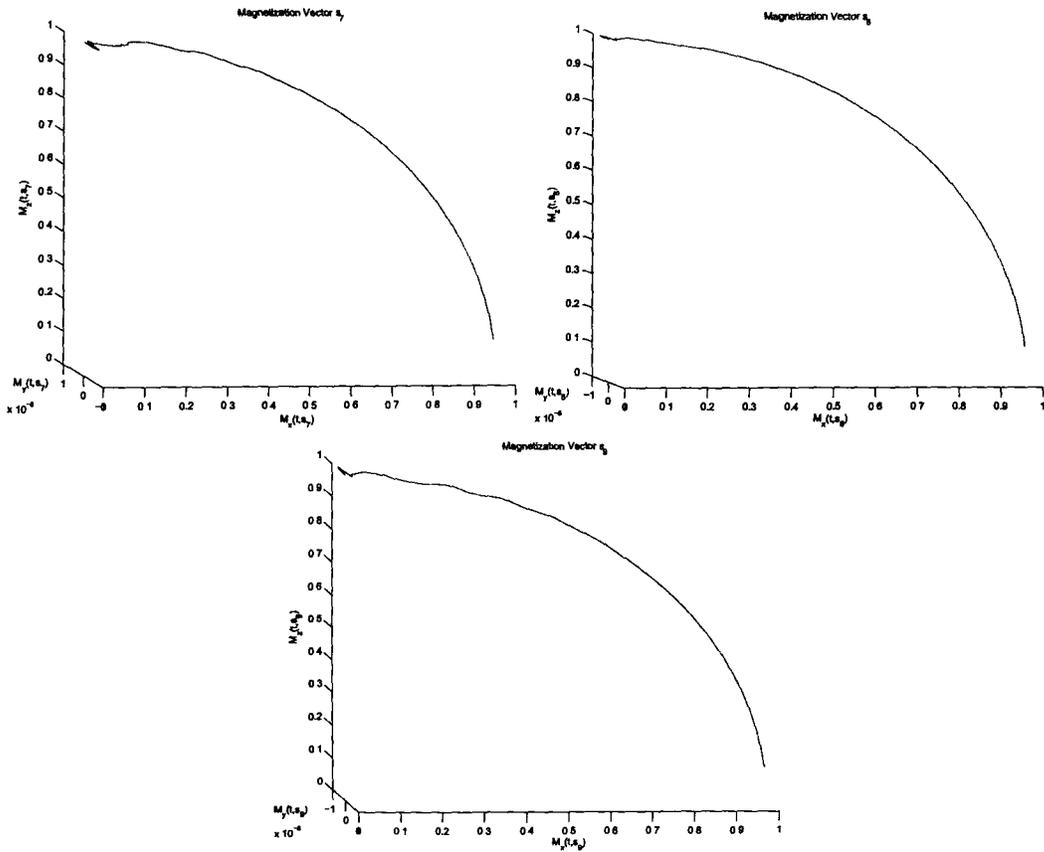


Figure 5.7: From left to right, magnetization vectors corresponding to coordinate positions s_7 , s_8 and s_9 .

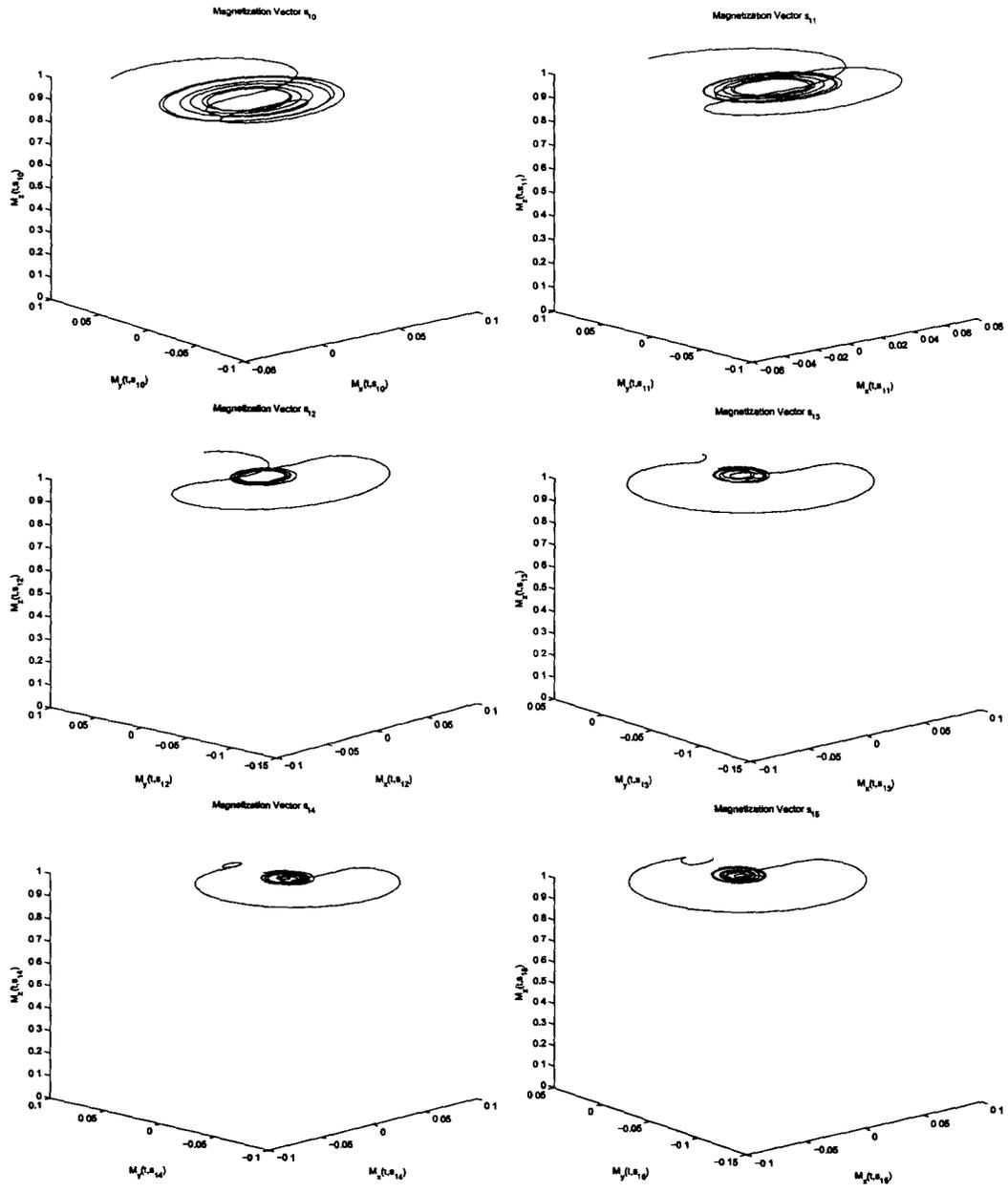


Figure 5.8: From left to right, magnetization vectors corresponding to coordinate positions s_{10} , s_{11} , s_{12} , s_{13} , s_{14} and s_{15} .

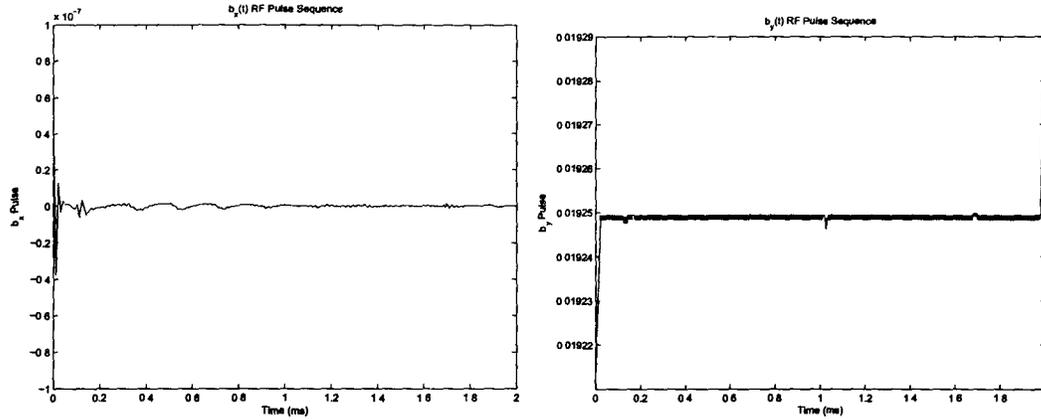


Figure 5.9: External magnetization components $b_x(t)$ and $b_y(t)$, shown respectively.

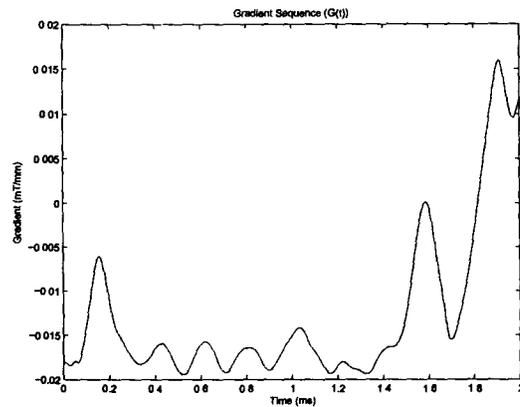


Figure 5.10: Gradient sequence pertaining to magnetization vectors plotted in Figures 5.6 – 5.8.

There are small differences between s_7 , s_8 and s_9 , however, s_7 and s_9 have more similarities and do not tip into the transverse plane as smoothly as s_8 . The gradient sequence is similar to that which was found in the five slice results, however, there are larger peaks and there is less of them. Also, the gradient is negative for the majority of the time duration except for the end when it steeply rises. Finally, the external magnetization components, $b_x(t)$ and $b_y(t)$, are again constant and linear, although, the value of $b_y(t)$ has increased to 0.01925. Notice that the vertical axis of $b_x(t)$ was decreased for illustrative purposes, it is still equal to zero if the same number of significant digits was used as in the five slice results.

5.4 Fifteen Slice Penalty Results

To increase the distance between the coordinate positions that were tipped into the transverse plane and allow a smooth transition between magnetization vectors in S_{in} and S_{out} , penalty variables and parameters were introduced. Initially, penalty variables were only integrated into the constraints corresponding to coordinate positions that were close to the border of S_{in} and S_{out} , as described in Section 4 of Chapter 2. However, when the penalty variables ξ_1 and ξ_2 were only added to constraints pertaining to s_i in a neighbourhood of $(\underline{s}_l, \underline{s})$ and (\bar{s}, \bar{s}_u) , no feasible solution was found. In fact, in order to increase the distance between s_7 and s_9 penalty variables had to be incorporated to each s_i vector in constraints (2.4.3 S_{in}) and (2.4.3 S_{out}). The remaining variables, constants, and constraints were consistent with what was used in the other results. The exact values for the coordinate positions were as follows:

-30	-28	-26	-24	-22	-20	-2	0	2	20	22	24	26	28	30
s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
ξ_2	ξ_2	ξ_2	ξ_2	ξ_2	ξ_2	ξ_1	ξ_1	ξ_1	ξ_2	ξ_2	ξ_2	ξ_2	ξ_2	ξ_2

where the positions that were penalized have their respective penalty variables listed below them. Notice that, with the addition of penalty variables and parameters the distance from s_7 to s_9 increased to 4 mm, compared to the 0.4 mm difference in the 15 slice results on page 84. Also, in the implementation of the penalized optimization problem from (2.4.1) – (2.4.8S), the value of the penalty parameters could not exceed, $\zeta_1 = 100$ and $\zeta_2 = 100$. The results for the penalized 15 coordinate simulation is illustrated in Figures 5.11 – 5.15. Information on the magnetic vector projection is shown in the graphs found in Figures 5.11 – 5.13. Specifically, Figures 5.11 and 5.13 correspond to magnetization vectors in S_{out} , and 5.12 refers to the coordinate positions in S_{in} .

The resulting RF pulse procedure, represented by external magnetization components $b_x(t)$ and $b_y(t)$, and gradient sequence, $G(t)$, is shown in Figures 5.14 – 5.15.

The precession of the magnetization vectors in S_{out} , Figures 5.11 and 5.13, have a much larger radius than that of the 15 slice problem. In fact, these magnetization vectors have at most three successive orbits in the entire time duration. The magnetization vectors in Figure 5.12, those s_i that belong to S_{in} , smoothly tip into the transverse plane and there is a greater similarity between s_7 and s_9 than in the preceding results. However, due to the penalty variables these vectors do not tip down as far into the transverse plane, to approximately a value of 0.2. Also, the y -axis is larger than the 15 slice problem, this is because the $M_y(t, \cdot)$ component of these magnetization vectors is increasing as they are descending into the transverse plane. The gradient sequence is more linear than either of the last results. It contains two large peaks, the first is negative and it starts about one quarter into the time period. The second peak is positive and it starts approximately three quarters into the time period. Also, the gradient sequence has three linear segments. One that is zero at the start of the sequence and the other two

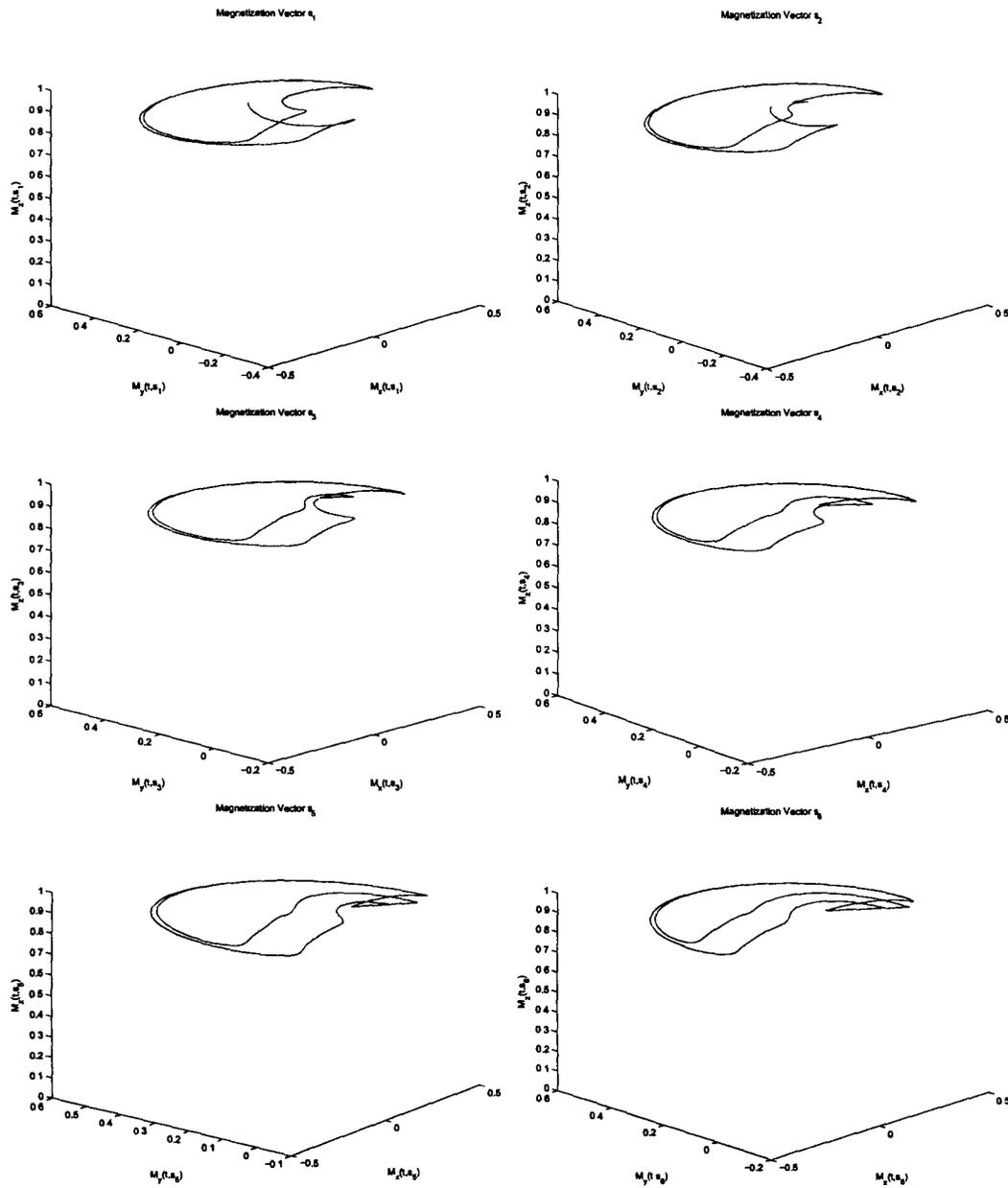


Figure 5.11: From left to right, magnetization vectors corresponding to coordinate positions s_1 , s_2 , s_3 , s_4 , s_5 and s_6 .

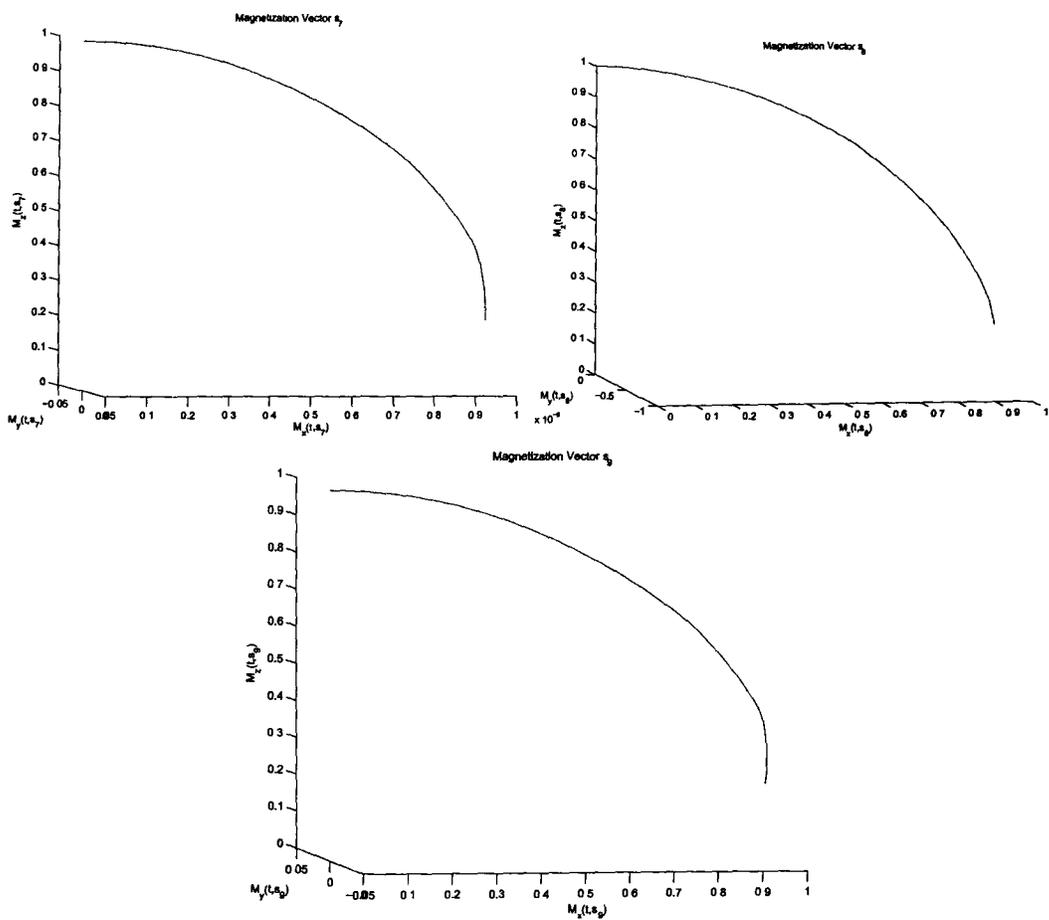


Figure 5.12: From left to right, magnetization vectors corresponding to coordinate positions s_7 , s_8 and s_9 .

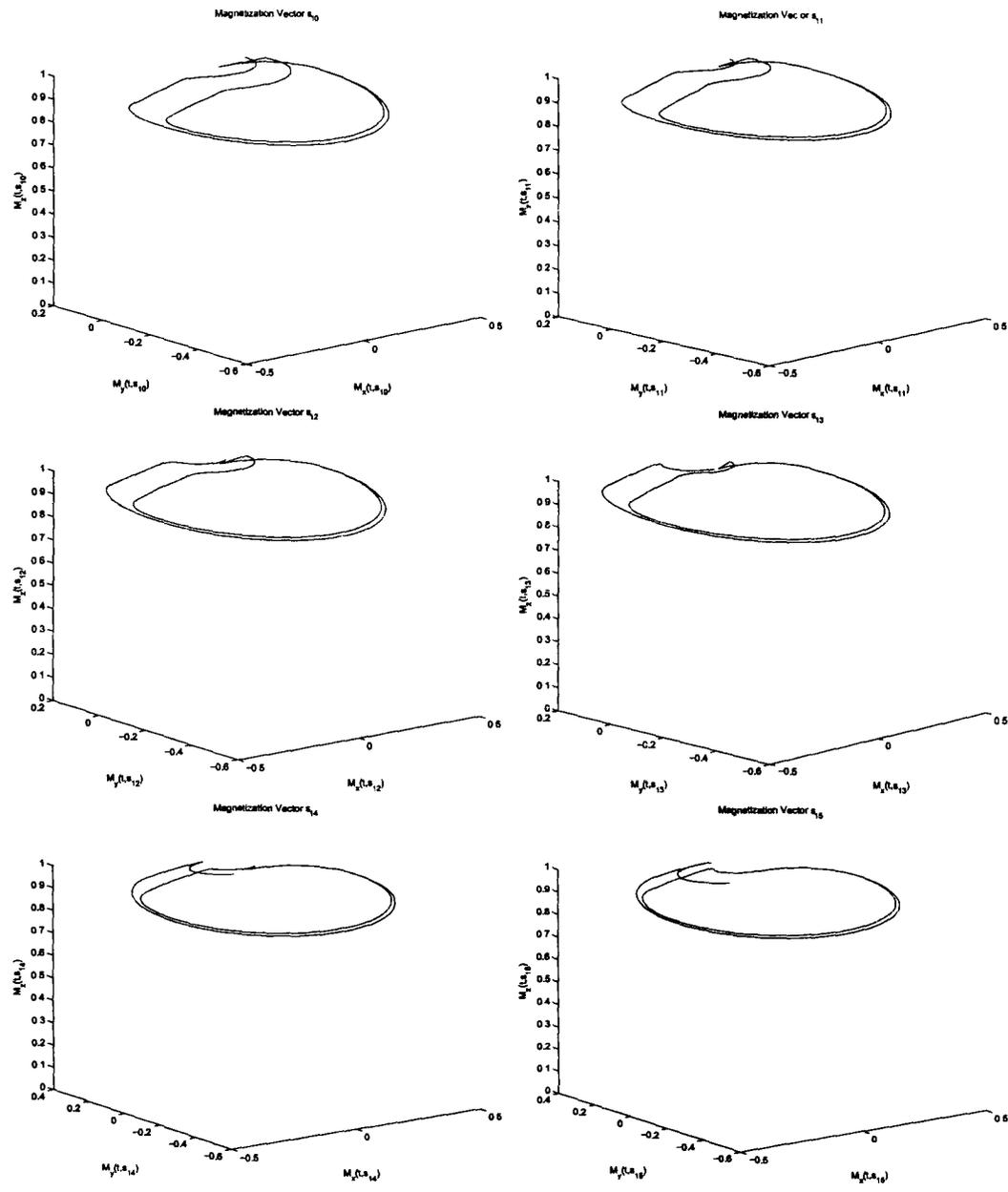


Figure 5.13: From left to right, magnetization vectors corresponding to coordinate positions s_{10} , s_{11} , s_{12} , s_{13} , s_{14} and s_{15} .

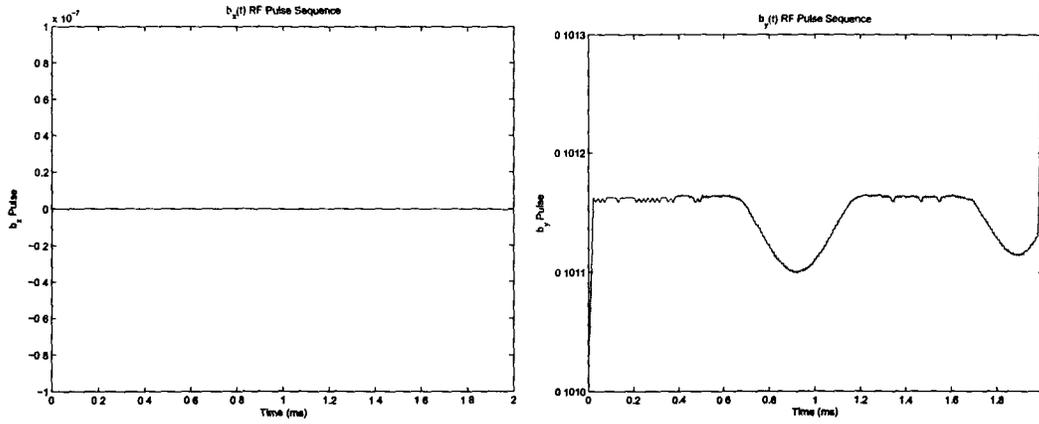


Figure 5.14: External magnetization components $b_x(t)$ and $b_y(t)$, shown respectively.

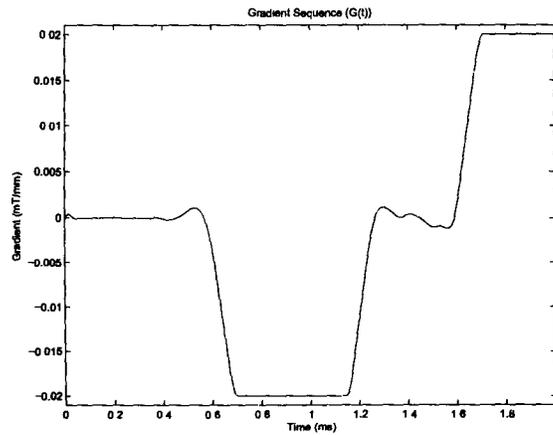


Figure 5.15: Gradient sequence pertaining to magnetization vectors plotted in Figures 5.11 – 5.13.

occur within the peaks, each having a value of exactly $\pm G_{\max}$. For the external magnetization components, $b_x(t)$ is again constant and has a value of zero. However, $b_y(t)$ is not as linear as the previous results and has increased to a value of approximately 0.10116. Nevertheless, this is still less than the amplitude for a conventional pulse, such as the one illustrated in Figure 4.4, which has a typical $b_y(t)$ value of approximately 0.7500. In fact, if we look at the value of the objective function in (2.2.2), namely

$$\int_0^T b_x^2(t) + b_y^2(t) dt ,$$

the 15 slice penalty results have an objective value of 0.1874, whereas the generic RF pulse produced a value of 0.5923. In addition, the penalty results had the largest objective for the VERSE pulse, the 15 slice results gave an objective value of 0.0385 and the 5 slice results were the lowest with a value of 0.0055.

As mentioned in the preceding chapter, many different initial guess solutions were attempted and alternative constants were tested, however, we have reported the best results derived using SOCS. In many cases SOCS could not find an optimal solution due to software limitations on memory, which became a great factor as the number of variables increased.

Chapter 6

Simulation

A background on the image reconstruction process involved in MRI is now described such that the signal generated by the VERSE pulse can be interpreted and analyzed. Using the numerical results computed in Chapter 5, an MRI simulation was designed to replicate practical procedures.

6.1 Image Reconstruction

In Magnetic Resonance Imaging, the signal produced by the RF pulse is mathematically amplified, digitized, transformed, and then combined together with other signals to form a final image. There are several techniques that can be used to produce a final image, however, the core of the systematic procedure is the same for all methods. As mentioned, the signal or the raw data of measurements is directly related to the distribution of transverse magnetization in the object or specimen. An RF coil is used to generate the RF pulse and detect the Magnetic Resonance (MR) signal at the end of the pulse. Thus, when an RF pulse, in conjunction with a gradient sequence is applied to an object or specimen, the signal is collected by the RF coil and then the data is relayed to an RF amplifier. The MR signal is considerably weaker than the input RF pulse, hence, an amplification of the MR signal that does not

distort the information is necessary. The signal is then digitized by an analog-to-digital converter before it is Fourier Transformed (FT) [11]. The FT of the signal is then stored in a computer and this process, beginning with the RF pulse excitation, is repeated a number of times. Eventually, when a specific number of FT signals are collected, they are combined to form a final image. A schematic drawing of the imaging sequence, also known as *MR signal processing*, is displayed in Figure 6.1 [11]. Actually, matters are hardly ever this simple as there are numerous possibilities for errors to occur while transferring the MR signal to the computer. A list of just a few of the problems that may arise are: a distorted final image due to the signal to noise ratio, aliasing or unwanted artifacts created by the signal transferring process, nonuniform sampling due to repetitive RF pulsing, image resolution problems and other imaging complications.

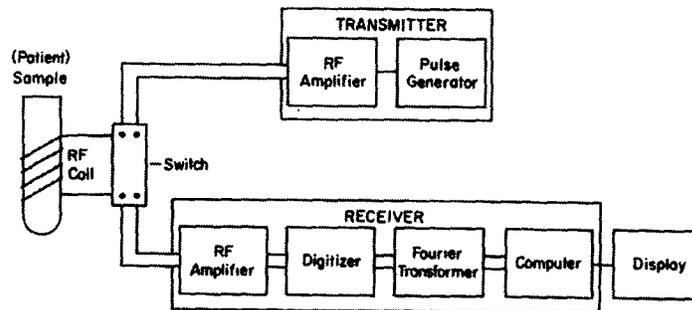


Figure 6.1: A schematic drawing of a general MR Imaging sequence.

6.2 Imaging the Signal

There are three different classes in which an image can be constructed using an RF pulse. They are defined by the dimensions of the signals they collect, namely, 1 Dimension (1D), 2 Dimension (2D) and 3 Dimension (3D), hence,

1D signal coverage may produce a two or three dimensional image. They differ in the way they collect or cover data in K-Space. K-Space is the space of digital information produced by an MR Signal before it is FT into an image [18]. As a general rule, the higher the class dimension, the faster K-Space is covered, which results in the quick production of a final image. However, higher dimensional classes involve an increase in the dynamics of the mathematical interpretation of the MR signal as well as demanding more gradient fields [20]. This essentially leads to additional factors that could pose a threat to image quality, hence, there is a “trade off” between image quality and speed in MR Imaging. As we are particularly interested in analyzing the performance of the VERSE pulse, lower dimensional coverage is more suitable for our investigation.

6.2.1 1D Coverage and Data Collection

As described in Chapter 2, gradients act to setup a one-to-one correspondence between frequency and spatial position, known as *frequency encoding*. To create an image we take a 1D FT of the amplified and digitized signal, as shown in an example illustrated in Figure 6.2. The pulse sequence to produce such an MR signal (Figure 6.2) is shown in Figure 6.3. Although Figure 6.3 is a simple example of what a 1D pulse sequence may look like, a few characteristics of the MR signal in the example should be highlighted. The peaks in the sinc-like signal correspond to different matter or tissues encountered during the pulse and will be an important part of the final image. The smaller local humps could also be different matter, tissues, or most likely noise. Noise is unwanted data information that is collected during signal processing which complicates and distorts the final image. Deciphering whether the data information is noise, or important information to creating an image, is interpreted by MRI software or manually by the user [20]. Thus, a signal that has clear

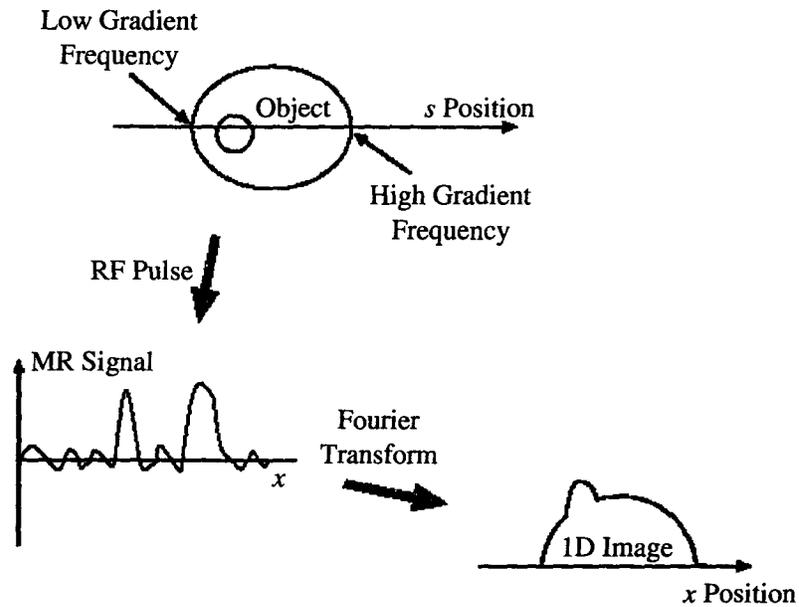


Figure 6.2: A varying gradient field in combination with an RF pulse is applied to an object that produces a signal that can be imaged in 1D.

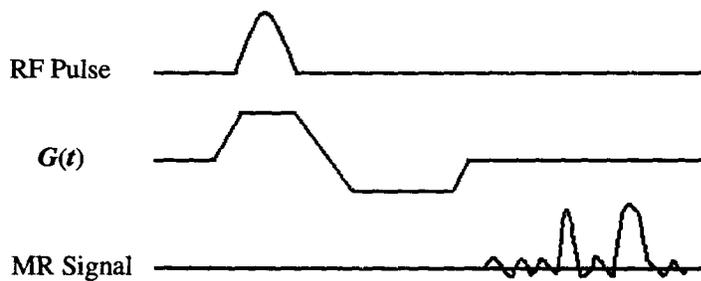


Figure 6.3: A varying gradient field and accompanying RF pulse producing a signal to be imaged in 1D.

divisions on what is matter or tissue and what is not, would produce an optimal final image.

The total received signal, which we will define as $\Upsilon(x)$, can be written by integrating over the entire excited line. Thus, ignoring the relaxation terms, we have

$$\Upsilon(x) = \int v(s)e^{-\varphi 2\pi s x} ds, \quad (6.2.1)$$

where the imaginary unit $\varphi \in \mathbb{C}$, $v(s)$ represents the signal generated at position s and x relates to its position on the x -axis of the image. Equation (6.2.1) represents the FT of $v(s)$, hence,

$$\Upsilon(x) = \text{FT}\{v(s)\} = \int v(s)e^{-\varphi 2\pi s x} ds, \quad (6.2.2)$$

which is the reason a FT is necessary in MR Imaging. Although, 1D coverage is a fairly elementary technique of collecting data in K-Space and would probably not be used in practice, it is an excellent tool to interpret RF pulse and gradient sequence design.

6.2.2 2D and 3D Coverage

Higher dimensional coverage utilizes the same methods as 1D coverage, however, instead of collecting data along a line of K-Space, information is collected about a plane (2D) or cube (3D). The additional information necessary for such coverage is supplied by adding extra gradients and introducing new variables into the received signal. In 2D coverage, we will use variables k_x and k_y to represent the K-Space x and y signal position, such that the total received signal becomes

$$\begin{aligned} \Upsilon(x, y) = \text{FT}\{v(k_x, k_y)\} &= \int_{k_x} \int_{k_y} v(k_x, k_y) e^{-\varphi 2\pi k_x x} e^{-\varphi 2\pi k_y y} dk_x dk_y \quad (6.2.3) \\ &= \int_{k_x} \int_{k_y} v(k_x, k_y) e^{-\varphi 2\pi (k_x x + k_y y)} dk_x dk_y, \end{aligned}$$

where x, y refer to their respective coordinate positions on the axes of the image. The introduction of the position variables x and y accounts for the additional orthogonal gradients necessary for 2D coverage [17].

Consider the extension of the 2D imaging equation (6.2.3) to all three spatial dimensions. With the addition of a third variable in K-Space, k_z , 3D coverage would follow essentially the same criterion

$$\begin{aligned} \Upsilon(x, y, z) &= \text{FT}\{v(k_x, k_y, k_z)\} \\ &= \int_{k_x} \int_{k_y} \int_{k_z} v(k_x, k_y, k_z) e^{-\varphi 2\pi(k_x x + k_y y + k_z z)} dk_x dk_y dk_z, \end{aligned} \quad (6.2.4)$$

where z accounts for the additional variable direction due to the three orthogonal gradients imperative to 3D coverage. As one can observe, 2D and 3D data collection provide more information to be FT'd into an image. Hence, more imaging information is collected per RF pulse and subsequently the time required to produce a final MRI image is decreased. However, the increased number of variables in the generated signal complicates the analysis of such data and makes the signal information difficult to interpret.

6.3 VERSE Simulation

A MRI simulation was implemented in Matlab to test the performance of the VERSE pulse sequence. Due to the excellent analytical signal produced by 1D data collection, the pulse was set such that the signal would collect data for 1D coverage. Using the Bloch equation we created an environment similar to that which is occurring in practical MRI. Thus, by providing the optimized VERSE pulse sequence, the gradient and RF pulse values were supplied to a voxel of protons that would eventually form a final image. Specifically, the VERSE values of $G(t_j)$, $b_x(t_j)$ and $b_y(t_j)$ for $j = 1, \dots, N$ were read into the Bloch equation (2.2.1) for magnetization vectors at different s_1, \dots, s_n posi-

tions. Although we used a total of n coordinate positions in the optimization of our model, the RF pulse and gradient sequence can be applied to $> n$ positions for imaging purposes. Thus, given $\bar{n} > n$ coordinate positions, N time discretizations, and an initial magnetization vector,

$$\vec{M}_0 = \begin{bmatrix} 0 \\ 0 \\ M_0 \end{bmatrix}, \quad (6.3.1)$$

the VESRE pulse sequence, namely $G(t_j)$, $b_x(t_j)$ and $b_y(t_j)$, was inserted into the vector

$$\begin{aligned} \frac{d\vec{M}(t_j, s_i)}{dt} &= \begin{bmatrix} M'_x(t_j, s_i) \\ M'_y(t_j, s_i) \\ M'_z(t_j, s_i) \end{bmatrix} \\ &= \begin{bmatrix} \gamma(-s_i G(t_j) M_y(t_j, s_i) + b_y(t_j) M_z(t_j, s_i)) \\ \gamma(s_i G(t_j) M_x(t_j, s_i) - b_x(t_j) M_z(t_j, s_i)) \\ \gamma(-b_y(t_j) M_x(t_j, s_i) + b_x(t_j) M_y(t_j, s_i)) \end{bmatrix}, \end{aligned} \quad (6.3.2)$$

for $j = 1, \dots, N$ and $i = 1, \dots, \bar{n}$. The integral of equation (6.3.2) was then evaluated

$$\vec{M}(t, s_i) = \int_{t_1}^{t_N} \frac{d\vec{M}(\hat{t}, s_i)}{dt} dt, \quad (6.3.3)$$

for each s_i value, note from Chapter 2 that $\hat{t} = [t_1, t_2, \dots, t_N]^T$. The values for the magnetization vectors were then converted into a signal by simulating the amplification and digitization used in MRI. For a complete description of how (6.3.3) was integrated and amplified, one can refer to the Appendix. At this step we would be able to investigate the signal produced by our simulation and examine its properties. As mentioned in the preceding section, a signal with distinctive peaks and minimal noise would produce a high-quality final image. Also, by changing the value of M_0 in the initial magnetization vector, \vec{M}_0 , we essentially replicate how an MRI processor would interpret different

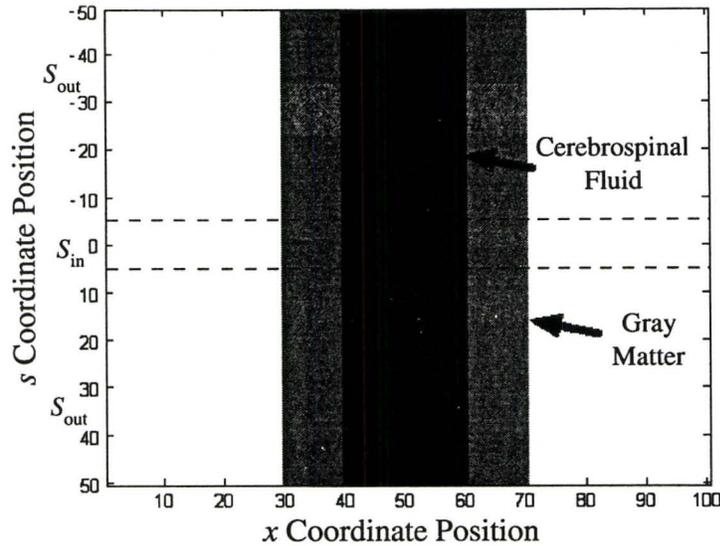


Figure 6.4: The position of cerebrospinal fluid and gray matter to be imaged by our MRI simulation.

tissues or matter.

Using the VERSE 15 slice results for the gradient and RF pulse sequence, an MRI simulation was conducted over two tissues, namely, gray matter and cerebrospinal fluid. The tissues were aligned vertically in the order of gray matter, cerebrospinal fluid, then once again gray matter, illustrated in Figure 6.4. As the signal generated by the pulse has a direct relationship with that of the tissues spin density, each tissues spin density value was substituted into M_0 at its respective position. Thus, a spin density value of 1.0 for cerebrospinal fluid and 0.8 for gray matter was used when performing the MRI imaging simulation described earlier. Also note, the VERSE pulse was designed to tip only the magnetization vectors in S_{in} into the transverse plane. Thus, the coordinate positions $s_i \in S_{in}$ would produce a peak in the signal when the VERSE pulse reaches the tissues for these $s_i \in S_{in}$ voxels. As

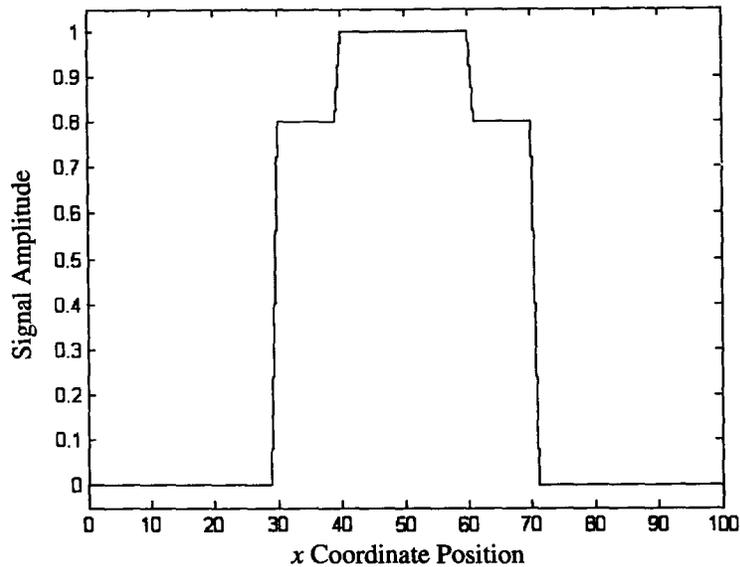


Figure 6.5: The signal produced by the VERSE pulse MRI simulation over two vertically aligned tissues.

detailed in the preceding chapters, voxels $s_i \in S_{in}$ are located at the center coordinate positions, approximately -5 to 5 in Figure 6.4. After amplification and digitization procedures were replicated, the signal produced by the VERSE pulse is shown in Figure 6.5. As it is evident in Figure 6.5, there is no evidence of noise and the peaks in the signal represent the position of the tissues that were imaged. Therefore, this would be a reliable signal for 1D coverage that would produce a high-quality final image.

Although the MRI simulation results seem promising in Figure 6.5, this was a fairly simple example to image since at each x position all the voxels, $s_i \in S$, were either in or out of the tissues. Now we complicate matters by placing the cerebrospinal fluid on an angle and removing the gray matter, as shown in Figure 6.6. As the vertical axis of Figure 6.6 represents the $s_i \in S$ coordinate positions, only the voxels in S_{in} should tip into the transverse

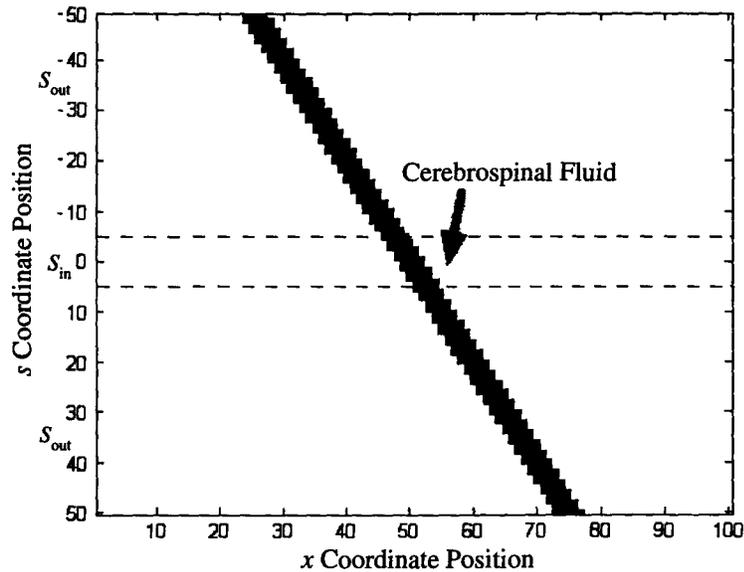


Figure 6.6: The angular position of cerebrospinal fluid to be imaged by our second MRI simulation.

plane, and hence, generate a signal. Again, voxels $s_i \in S_{in}$ are located at the center coordinate positions, approximately -5 to 5 in the illustration. Thus, a signal should only be produced when the VERSE pulse reaches these voxels in the fluid. Figure 6.7 represents the signal generated after the 15 slice VERSE RF pulse and gradient sequence was used to stimulate particular voxels within the cerebrospinal fluid into the transverse plane. As it is shown in Figure 6.7, the large central peak in the signal represents when the VERSE pulse reaches the voxels in S_{in} of the fluid. The peak in the center of the figure is also very distinctive and the overall signal has minimal noise. Figure 6.8 represents the signal produced when a generic RF pulse and gradient sequence, described in Section 4.3, was used. In comparing Figure 6.7 to Figure 6.8, one can see that the signal produced by the VERSE pulse has less noise, a more distinctive peak, and a much clearer division with regards to what is tissue and what is not. In addition, the objective value, which defines the strength of the RF

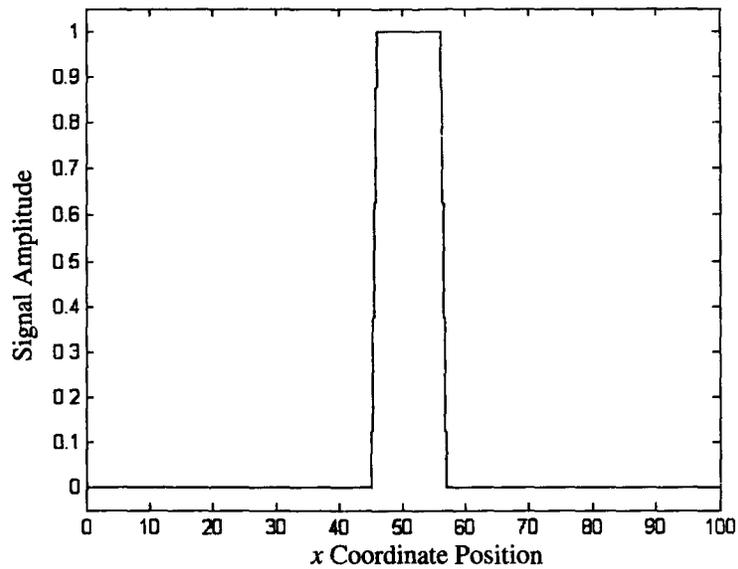


Figure 6.7: The signal produced by the VERSE pulse MRI simulation over the diagonal cerebrospinal fluid.

pulse necessary to produce such a signal, was 0.0385 for the VERSE pulse, substantially lower than that of the conventional pulse, which had an objective value of 0.5923. Thus, Figure 6.5 and Figure 6.7 lead us to conclude that the VERSE pulse generates a reliable signal for 1D coverage in MR Imaging.

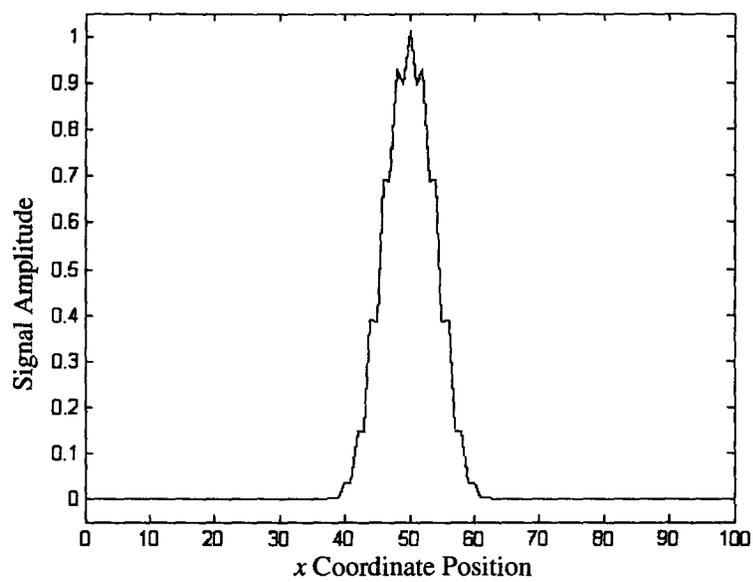


Figure 6.8: The signal produced when a generic RF pulse and gradient sequence is applied to the diagonal cerebrospinal fluid.

Chapter 7

Conclusions and Future Work

We designed the VERSE model to reduce the SAR of RF pulses by maintaining a constant RF pulse strength (\vec{B}_{rf} value) and generating high quality MR signals. It was shown that the VERSE results produced strong MR signals with clear divisions of the location of the tissue being imaged. For this reason various MRI studies utilizing VERSE pulses could be developed in the near future.

The observations noted in Sections 5.2, 5.3, and 5.4 of the Results Chapter deserve some additional reasoning and explanation. To begin, the reader should understand that the symmetry displayed between coordinate position vectors in each of the result cases was precisely designed in (2.3.1) of the VERSE model. However, the precession illustrated by the magnetization vectors was not directly part of the VERSE design, it was a consequence of the Bloch constraint (2.2.3*S*). Nonetheless, the precession shown in our results validated our design since it occurs within the nucleus of atoms in physical MRI. Furthermore, investigating the precession of the magnetization vectors in the 5 slice results, it was shown that they tightened their orbit after a certain number of revolutions. This larger orbit was most probably due to the lack of variables in the 5 slice problem, as a tighter precessional orbit was shown in

the 15 slice results. The magnetization vectors in the 15 slice penalty results, however, had a much larger radial orbit than in either of the other cases. This was due to the fact that the penalty parameters allowed the feasible range of the constraints on these variables to be larger. With respect to precession, the 15 slice penalty results were the most unrealistic, however, penalty variables and parameters allowed the span of the magnetization vectors in S_{in} to be fairly large, which also played a part in terms of practical MRI. In addition, investigating only the coordinate positions in S_{in} one should note that the s_3 magnetization vector in the 5 slice results tips into the transverse plane with a constant y -axis value, hence, the 2D graph in Figure 5.2. This was another essential part of the VERSE pulse design, where the rotating frame of reference was utilized in our formulation. However, in both 15 slice results, motion in these coordinate positions was evident. Coordinate positions s_7 , s_8 and s_9 in the 15 slice results seemed to resonate a fair bit in the y direction at the start of the time interval, as if they have not received enough energy to tip into the transverse plane. Similarly, the y -axis of the 15 slice penalty results increased by a small amount with time. The motion of these vectors is due to a combination of their nonzero s_i position values and the increased dimensions of the problem. In addition, penalty variables relaxed the constraints of the 15 slice penalty results, which did not induce the wave-like motion found in the vectors of the 15 slice results. One could conclude that in order to have improved transverse tipping and increase the length of magnetization vectors in S_{in} , larger ϵ_2 values are necessary, however, whether or not such a large precessional value is a realistic approximation would then become a factor. Finally, as evident in all three result cases, as the coordinate positions in S_{out} approach the border of S_{in} their precessional orbits were not as tight and their radius of precession increased. This is well illustrated in the graphs of the 15 slice results, namely Figures 5.6 and 5.8, which give a good example of the off resonance characteristics that occur in practical MRI. Although off

resonance was an attribute of MRI already considered in the formulation of the VERSE model, its minimal presence validates our results from a physical MR perspective.

The aim of the VERSE pulse was to minimize SAR by maintaining a constant RF pulse ($b_x(t)$ and $b_y(t)$ values), which was established in each of the results of Section 5.2, 5.3 and 5.4. Although the values of $b_x(t)$ were almost identical for each case, $b_y(t)$ values increased with the number of slices. This was expected since an increase in the number of discretization points would require additional energy to tip the voxels into the transverse plane, yielding an increase in the strength of the RF pulse, or larger $b_y(t)$ value. The $b_y(t)$ value for the penalty results were the greatest and were not as constant as the other two cases. This was again due to the penalty variables and parameters, however, the nonlinear portions of the $b_y(t)$ graph only had small differences with respect to the other values. Also, when comparing the VERSE pulse to conventional pulses the VERSE objective value was lower for all three cases, and hence, did not require as much energy to tip the magnetization vectors into the transverse plane. Finally, the most intricate part of the VERSE pulse results is the gradient sequence. Since we optimized for the RF pulse in our model, this process returned the gradient sequence that would allow such a pulse to occur. In other words, in order to use the $b_x(t)$ and $b_y(t)$ pulse design, the accompanying gradient sequence, mainly derived from the Bloch constraint, would have to be imposed to acquire a useable signal. With regards to practical MR gradient sequences, the 15 slice penalty results produced the simplest and most reasonable gradient values to implement, particularly due to its large linear portions. However, if necessary, regardless of the difficulty, one of the other gradients could be implemented. In addition, as the number of slices increased between the result cases they caused the difference between the largest positive and negative peaks in the gradient graphs

to amplify. Finally, all three results had similar features in the sense that they each started off fairly negative and then ended up quite positive. This is a very interesting consequence of the VERSE pulse, as shown in Chapter 4, conventional gradient sequences usually have the opposite characteristics. In terms of our MRI simulation, good signal results were produced for such unique gradient sequences, which would justify further research with VERSE pulses. In fact, the Results and Simulation Chapters demonstrated that the VERSE RF pulse and gradient sequence were viable and could be applied to practical MRI.

Future Work

The VERSE pulse proved to have encouraging MRI results and performed to be better than anticipated with respect to useable MR imaging signals. Due to limitations on time, there are still areas left for investigation and various elements of the VERSE model that can be improved. A few of the issues that should be taken into account for future developments are:

- Add rotation into the equations;
- Apply the VERSE model to more than 50 slices;
- Add spin-lattice and spin-spin proton interactions into the VERSE formulation;
- Apply alternative optimization software to the problem;
- Investigate other variations of VERSE pulses;
- Test on an MRI machine.

The first five issues could possibly improve the VERSE pulse model, or at least identify the items that are necessary for the potential advancements of this RF pulse. The issues are listed in sequential order, starting with what we

believe is the most important item to be addressed. As most are self explanatory, adding rotation into the equations was one of the factors that deemed to be important after the results were examined. By integrating the rotating frame of reference into our equations we eliminated the y -axis. It is possible that this was a source of singularities when optimizing and therefore caused SOCS to increase the size of its working array, potentially creating memory problems. Although this issue was taken into account, further investigation is warranted to intelligently integrate rotation into our model. Finally, the last item would be more or less of a final approval for such an RF pulse sequence.

The research and work done with the VERSE pulse has built an excellent foundation for future developments. This study illustrates that optimization can have a great effect on a highly dynamical processes such as RF pulses in Magnetic Resonance Imaging.

Appendix

The MR Signal

To produce the MR signals generated in Chapter 6, we integrated the Bloch equation (6.3.3) shown on page 103 by first taking the integral of the magnetization vector in the static external field B_0 , then rotating it by the magnetic field generated by the VERSE RF pulse. In order to accomplish this, we first calculated the ℓ_2 norm of $s_i G(t)$ and $b_y(t)$,

$$N(t, s_i) = \sqrt{(s_i G(t))^2 + b_y^2(t)},$$

where $i = 1, \dots, n$. Then substituting the norm into the z component of the external magnetization matrix, the Bloch equation for a static magnetic field is equivalent to

$$\frac{d\vec{M}(t, s_i)}{dt} = \begin{bmatrix} 0 & \frac{N(t, s_i)}{\delta} & 0 \\ -\frac{N(t, s_i)}{\delta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} M_x(t, s_i) \\ M_y(t, s_i) \\ M_z(t, s_i) \end{bmatrix}, \quad (\text{A.1})$$

where δ is an scaling parameter. Equation (A.1) produces three differential equations,

$$\frac{dM_x(t, s_i)}{dt} = \frac{N(t, s_i)}{\delta} M_y(t, s_i), \quad \frac{dM_y(t, s_i)}{dt} = -\frac{N(t, s_i)}{\delta} M_x(t, s_i),$$

$$\frac{dM_z(t, s_i)}{dt} = 0,$$

and after taking the derivative of each we are left with the following second order equations,

$$M_x''(t, s_i) + \frac{N(t, s_i)}{\delta} M_x(t, s_i) = 0, \quad M_y''(t, s_i) + \frac{N(t, s_i)}{\delta} M_y(t, s_i) = 0. \quad (\text{A.2})$$

Note, we excluded writing $M_z''(t, s_i)$ since it was equal to zero. Integrating the two equations in (A.2) and making the appropriate constant substitutions generates the following well-known solutions,

$$M_x(t, s_i) = M_x(0, s_i) \cos\left(\frac{N(t, s_i)}{\delta} t\right) + M_y(0, s_i) \sin\left(\frac{N(t, s_i)}{\delta} t\right) \quad (\text{A.3})$$

and

$$M_y(t, s_i) = M_y(0, s_i) \cos\left(\frac{N(t, s_i)}{\delta} t\right) - M_x(0, s_i) \sin\left(\frac{N(t, s_i)}{\delta} t\right). \quad (\text{A.4})$$

If we approximate the continuous controls $b_x(t)$, $b_y(t)$ and $s_i G(t)$ by piecewise constant functions, then on each constant interval we can exactly integrate the Bloch equation by making a coordinate transformation. Using (A.3) and (A.4), we constructed a matrix, \mathbf{R} , representing the integration of the Bloch equation for an external magnetic field in the z direction, hence

$$\mathbf{R} = \begin{bmatrix} \cos\left(\frac{N(t, s_i)}{\delta}\right) & \sin\left(\frac{N(t, s_i)}{\delta}\right) & 0 \\ -\sin\left(\frac{N(t, s_i)}{\delta}\right) & \cos\left(\frac{N(t, s_i)}{\delta}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{A.5})$$

In the special case $b_x(t) = 0$, the matrix

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{s_i G(t)}{N(t, s_i)} & -\frac{b_y(t)}{N(t, s_i)} \\ 0 & \frac{b_y(t)}{N(t, s_i)} & \frac{s_i G(t)}{N(t, s_i)} \end{bmatrix}, \quad (\text{A.6})$$

transforms the vector $[0, b_y(t), s_i G(t)]^T$ into $[0, 0, N(t, s_i)]^T$, and

$$\mathbf{Q}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{s_i G(t) N(t, s_i)}{(s_i G(t))^2 + b_y^2(t)} & \frac{b_y(t) N(t, s_i)}{(s_i G(t))^2 + b_y^2(t)} \\ 0 & -\frac{b_y(t) N(t, s_i)}{(s_i G(t))^2 + b_y^2(t)} & \frac{s_i G(t) N(t, s_i)}{(s_i G(t))^2 + b_y^2(t)} \end{bmatrix}, \quad (\text{A.7})$$

transforms it back. Multiplying these three matrices, (A.5) - (A.7), by the magnetization vector $\vec{M}^i \equiv \vec{M}(t, s_i)$, we have

$$\mathbf{Q}^{-1} \mathbf{R} \mathbf{Q} \vec{M}^i = \begin{bmatrix} \cos\left(\frac{N^i(t)}{\delta}\right) M_x^i + \frac{\sin\left(\frac{N^i(t)}{\delta}\right) s_i G(t)}{N^i(t)} M_y^i - \frac{\sin\left(\frac{N^i(t)}{\delta}\right) b_y(t)}{N^i(t)} M_z^i \\ -s_i G(t) \Theta_1 M_x^i + \frac{(s_i G(t))^2 \cos\left(\frac{N^i(t)}{\delta}\right) + b_y^2(t)}{(s_i G(t))^2 + b_y^2(t)} M_y^i + \Theta_2 M_z^i \\ b_y(t) \Theta_1 M_x^i + \Theta_2 M_y^i + \frac{b_y^2(t) \cos\left(\frac{N^i(t)}{\delta}\right) + (s_i G(t))^2}{(s_i G(t))^2 + b_y^2(t)} M_z^i \end{bmatrix}, \quad (\text{A.8})$$

where $N^i(t) \equiv N(t, s_i)$,

$$\Theta_1 = \frac{N^i(t) \sin\left(\frac{N^i(t)}{\delta}\right)}{(s_i G(t))^2 + b_y^2(t)},$$

and

$$\Theta_2 = \frac{s_i G(t) b_y(t) - s_i G(t) b_y(t) \cos\left(\frac{N^i(t)}{\delta}\right)}{(s_i G(t))^2 + b_y^2(t)}.$$

Using matrix (A.8), we constructed a loop for the total number of time discretizations and input the VERSE pulse values for each s_i coordinate position, which produced the signals shown in Chapter 6.

This was just one of the many possible methods of integrating (6.3.3), other methods could be applied, however, they would involve similar ingredients to the ones illustrated. The main advantage of exactly integrating the equations on each interval is that $|\vec{M}(t, s_i)|$ is constant, as it should be. By

including the norm in the integration $\vec{M}(t, s_i)$ did not blow up, which may occur with other methods. Integrating techniques such as the Trapezoid rule, Simpson rule, Riemann integral and taking finite differences proved to be numerically unstable and were unable to integrate standard test cases accurately. This was because such integrating methods do not take rotation into account, which was an important part of the signal information. Hence, methods that include rotation and consider some type of normalization between time increments would provide a strong integrating tool for generating MR signals.

Bibliography

- [1] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 1995.
- [2] J. T. Betts. Issues in the direct transcription of optimal control problems to sparse nonlinear programs. *Control Applications of Optimization*, 115:3–17, 1994.
- [3] J. T. Betts. A sparse nonlinear optimization algorithm. *Journal of Optimization Theory and Applications*, 82:519–541, 1994.
- [4] J. T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, Philadelphia, 2001.
- [5] J. T. Betts and W. P. Huffman. SOCS Manual: Release 6.2 M and CT-TECH-01-014, The Boeing Company, PO Box 3707, Seattle, WA 98124-2207, 2001.
- [6] Z. Bia, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Society for Industrial and Applied Mathematics, Philadelphia, 2000.
- [7] S. C. Bushong. *Magnetic Resonance Imaging: Physical and Biological Principles*. Mosby, Toronto, 2nd edition, 1996.

- [8] M. M. Connors and D. Teichroew. *Optimal Control of Dynamic Operations Research Models*. International Textbook Company, Pennsylvania, 1997.
- [9] S. M. Conolly, D. G. Nishimura, and A. Macovski. Variable-rate selective excitation. *Journal of Magnetic Resonance*, 78:440–458, 1988.
- [10] L. Cooper. *Applied Nonlinear Programming: for Engineers and Scientists*. Aloray Inc., Englewood, New Jersey, 1974.
- [11] T. S. Curry, J. E. Dowdey, and R. C. Murry. *Christensen's Physics of Diagnostic Radiology*. Lippincott Williams and Wilkins, New York, 4th edition, 1990.
- [12] E. de Klerk, C. Roos, and T. Terlaky. *Nonlinear Optimization (WI3 031)*. Delft University of Technology, January 2003.
- [13] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, New York, 1985.
- [14] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Some Theoretical Properties of an Augmented Lagrangian Merit Function, Tech. Report SOL 86-6. Department of Operations Research, Stanford University, Apr. 1986.
- [15] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, Toronto, 1981.
- [16] N. I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. *Mathematical Programming*, 32:90–99, 1985.

- [17] E. M. Haacke, R. W. Brown, M. R. Thompson, and R. Venkatesan. *Magnetic Resonance Imaging: Physical Principles and Sequence Design*. John Wiley and Sons, Toronto, 1999.
- [18] Z. P. Liang and P. C. Lauterbur. *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. IEEE Press, New York, New York, 2001.
- [19] G. P. McCormick. *Nonlinear Programming: Theory, Algorithms, and Applications*. John Wiley and Sons, Toronto, 1983.
- [20] D. G. Nishimura. *Principles of Magnetic Resonance Imaging*. Department of Electrical Engineering, Stanford University, 1996.
- [21] R. T. Rockafellar. The multiplier method of Hestenes and Powell applied to convex programming. *Journal of Optimization Theory and Applications*, 12:555–562, 1973.