

University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2023

Optimization in MRI Experiment Design and Image Reconstruction

Mark Armstrong
University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>



Part of the [Health Information Technology Commons](#), and the [Other Analytical, Diagnostic and Therapeutic Techniques and Equipment Commons](#)

Recommended Citation

Armstrong, Mark, "Optimization in MRI Experiment Design and Image Reconstruction" (2023). *Electronic Theses and Dissertations*. 8966.

<https://scholar.uwindsor.ca/etd/8966>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Optimization in MRI Experiment Design and Image Reconstruction

By

Mark Armstrong

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Physics
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2023

© 2023 Mark Armstrong

Optimization in MRI Experiment Design and Image Reconstruction

by

Mark Armstrong

APPROVED BY:

D. Yang
Department of Mathematics and Statistics

J. Rau
Department of Physics

D. Xiao, Advisor
Department of Physics

January 17, 2023

DECLARATION OF CO-AUTHORSHIP/ PREVIOUS PUBLICATION

I. Co-Authorship

I hereby declare that this thesis incorporates material that is result of joint research, as follows:

- Chapter 3 incorporates unpublished material co-authored with Enrica Wilkens under the supervision of professors Dr. Dan Xiao and Dr. Cornelius Faber. My primary contributions towards the publications were developing and coding the CS reconstruction algorithm, training the dictionary for DL, coding the simulations used to test the method, and writing the manuscript. The contributions of Enrica were primarily through animal handling, acquiring images, identifying and counting cells in images, and editing the manuscript. Dr. Xiao and Dr. Faber contributed to experiment design, refinement of ideas and editing the manuscript.

I am aware of the University of Windsor Senate Policy on Authorship, and I certify that I have properly acknowledged the contribution of other researchers to my thesis and have obtained written permission from each of the co-authors to include the above material in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Previous Publication

This thesis includes one original paper that have been previously published/submitted to journals for publication, as follows:

Thesis Chapter	Publication title	Publication status
<i>Chapter 3</i>	<i>Dynamic Cell Tracking MRI with Variable Temporal Resolution Cartesian Sampling</i>	<i>Submitted</i>

I certify that I have obtained a written permission from the copyright owners to include the above published material in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

III. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owners to include such materials in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Magnetic Resonance Imaging (MRI) is a non-invasive imaging modality with numerous applications due to its flexible contrast and high resolution. Recent improvements in computation power have enabled optimizations which were previously out of reach. This has led to improvements in image reconstruction and experiment design.

Compressed sensing (CS) allows for images to be reconstructed using less data than is normally required leading to faster image acquisitions. In this thesis, CS is applied to experiments tracking individual cells in time lapse MRI. The faster image acquisition with CS reduces blurring from cell motion, improving the Contrast-to-Noise ratio (CNR) of moving cells and allows faster cells to be detected.

Pi Echo Planar Imaging (PEPI) is an MRI pulse sequence that allows high resolution images to be acquired quickly with relatively low gradient duty cycle. Low field applications benefit significantly from low gradient duty cycle as it reduces concomitant magnetic field artifacts, so PEPI is an attractive option for affordable low field scanners. However, there are challenges in implementing PEPI, due to its high requirement on the flip angle of the π RF pulses. Deviation of the flip angle causes coherence pathway artifacts restricting PEPI to small samples in the homogeneous region of the RF coil and preventing 2D slice selective experiment. In this thesis, the coherence pathway artifacts are addressed using an optimized phase cycling scheme, reducing the flip angle sensitivity, and enabling a slice selective PEPI sequence.

ACKNOWLEDGEMENTS

I would like to thank my research supervisor, Dr. Dan Xiao for all your guidance and wisdom. I have learned a lot under your guidance, and I am grateful for the many hours spent helping me edit my presentations and manuscripts.

Thank you to Dr. Rehse for sparking my first interest in research, and to Dr. Kim for the many tools he added to my toolbox. Thank you to my lab mates current and former for the good times. I am especially thankful to Layale, Tristhal, and Jonathan for the time we spent discussing various topics, and their help preparing for presentations.

Thank you to my family and especially my wife Calli for their encouragement throughout the years. I am grateful for your unconditional love and support.

Thank you to the Government of Ontario (OGS) and NSERC for funding.

TABLE OF CONTENTS

DECLARATION OF CO-AUTHORSHIP/ PREVIOUS PUBLICATION	iii
ABSTRACT.....	v
ACKNOWLEDGEMENTS.....	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
NOMENCLATURE	xv
CHAPTER 1 Introduction.....	1
1.1 Overview of Work	1
1.2 Summary of Master’s Work.....	2
1.3 Thesis Outline	3
CHAPTER 2 Magnetic Resonance Imaging Basics	4
2.1 Nuclear Magnetic Resonance.....	4
2.2 Magnetic Resonance Imaging.....	6
2.2.1 Slice Selection.....	6
2.2.2 Spatial Encoding	7
2.2.3 K-space Trajectory	8
2.3 MRI Pulse Sequences	9
2.3.1 Gradient Echo	9
2.3.2 Spin Echo and Fast Spin Echo	11
2.3.3 PEPI	12
CHAPTER 3 Dynamic Cell Tracking MRI with Variable Temporal Resolution Cartesian Sampling	14
3.1 Introduction.....	14
3.2 Theory	17
3.2.1 Flexible Sampling Scheme.....	17

3.2.2 Dictionary Learning and Low Rank Regularized Compressed Sensing	18
3.3 Methods.....	22
3.3.1 Simulations using a Virtual Phantom.....	22
3.3.2 In Vitro Phantom Experiments	23
3.3.3 In Vivo Experiments	24
3.3.4 MRI Acquisition	25
3.3.5 Reconstruction	25
3.4 Results.....	26
3.4.1 Simulations	26
3.4.2 Phantom Experiments	29
3.4.3 In Vivo Experiments	33
3.5 Discussion.....	36
3.6 Conclusion	39
CHAPTER 4 Optimized Phase Cycling for Coherence Pathway Selection in PEPI Experiments	41
4.1 Introduction.....	41
4.2 Theory	43
4.2.1 PEPI Pulse Sequence	43
4.2.2 The Partitioning Method	43
4.2.3 Cost Function	47
4.3 Methods.....	48
4.3.1 Cost Function	48
4.3.2 Simulations	49
4.3.3 Phantom Experiments	49
4.3.4 Mouse Head Experiments	50
4.4 Results.....	50
4.4.1 Cost Function	50
4.4.2 Simulations	51
4.4.3 Phantom Experiments	52
4.4.3 Mouse Head Experiments	55
4.4 Discussion.....	57
4.5 Conclusion	60

CHAPTER 5 Conclusions and Future Work	61
5.1 Conclusions.....	61
5.2 Future Work.....	62
REFERENCES	63
APPENDICES	68
Appendix 1.....	68
A1.1 Partition Method for 3 Pulses.....	68
A1.2 Implementation of Pathway Calculation.....	75
Appendix 2.....	83
VITA AUCTORIS	94

LIST OF TABLES

Table 1. The contrast to noise ratio for each cell in each reconstruction of the simulated phantom experiment in Fig. 3-3. The inner two cells ($< 5 \mu\text{m/s}$) in each spoke are considered slow, while the outer three ($> 5\mu\text{m/s}$) are fast. Contrasts greater than 0.25 are categorized as high contrast, while the cells below 0.25 are low contrast.

27

Table 2. The number of pixels for each cell along the 0.4 contrast spoke. It was evaluated by manually selecting a region around the cell. The number of pixels was determined by a threshold, which was a weighted average of the minimum intensity ($\times 0.4$) and the background intensity ($\times 0.6$).

29

LIST OF FIGURES

- Fig. 2-1. The pulse sequence diagram for a 2D GRE. A slice select gradient (G_s) and RF pulse are used to excite signal in a plane. Frequency and phase encoding gradients (G_f and G_p) are used to encode the remaining two directions. The excitation is repeated with different phase encoding gradient strengths until the full Cartesian k-space is covered. 10
- Fig. 2-2. The pulse sequence diagram for a 2D SE. A slice selective excitation pulse is used to excite signal in a plane similar to a GRE. A slice selective refocusing pulse is then applied to create a spin echo. The spatial encoding is applied similar to a GRE and the timing is set so the gradient echo and spin echo align. 11
- Fig. 2-3. The pulse sequence diagram for a 2D FSE. The experiment is similar to an SE except that a chain of refocusing pulses is used instead of one. This generates a train of echoes with different spatial encodings. The gradients are balanced within each echo interval so that the net phase accumulation is 0. 12
- Fig. 2-4. The pulse sequence diagram for a 3D PEPI. The first phase encoding gradient is applied before every other echo. The second phase encoding gradient is applied only after the excitation RF pulse. The 180° refocusing RF pulse changes k to $-k$. Other k-space trajectories are possible. 13
- Fig. 3-1. Variable density sampling scheme. (a) 2D k-space is divided into bins of high frequency lines and a low frequency portion (orange) which is acquired in the center of the fully sampled data. Each of the high frequency groups is sampled in a short time interval, as indicated by the same color, and provides a high undersampling ratio frame with a relatively uniform incoherent k-space coverage. Sets of neighboring frames can be combined for reconstructions with (b) a lower undersampling ratio or (c) a fully sampled k-space. 18
- Fig. 3-2. Example atoms of simulated cell features (top 2 rows), trained contour features (middle 2 rows), and trained brain features (bottom 2 rows). 20
- Fig. 3-3. Simulations using the variable density sampling scheme. A stationary phantom (a) was simulated containing cells represented by hypointense spots of varying contrasts as indicated at each spoke. The phantom was then rotated along the isocenter at a frequency of $2.93 \cdot 10^{-4}$ rotations/second to create different linear velocities ($1.7 \mu\text{m/s}$, $3.5 \mu\text{m/s}$, $5.2 \mu\text{m/s}$, $7.0 \mu\text{m/s}$, $8.7 \mu\text{m/s}$ at the five radial positions of the hypointense spots, respectively). The variable Cartesian sampling k-space data was processed to reconstruct the fully sampled (1X) image (b), 2.4X (c) and 4.8X (d) undersampled images. CNR for all cells in each reconstruction are listed in Table 1. Cells with various velocities and contrasts are highlighted for comparison showing blurring for fast-moving cells as well as reduction of the “T” shaped artifact and improvement in contrast in the accelerated reconstructions. 28

Fig. 3-4. Rotating phantom experiments with the variable density sampling scheme. An agarose phantom with embedded MPIOs was scanned in a stationary position (a) as a reference and for a rotation of $2.93 \cdot 10^{-4}$ rotations/second mimicking cells which speeds of up to $9.2 \mu\text{m/s}$. In the rotational case, the k-space data was processed to reconstruct the fully sampled (1X) image (b), 2.4X (c) and 4.8X (d) undersampled images. Zoomed image details with adjusted window/level settings and the coil sensitivity corrected show examples of particles with various velocities and contrasts. They demonstrate temporal blurring of fast-moving particles and improvement in contrast using undersampled reconstructions.

30

Fig. 3-5. Comparison of particle CNR in the fully sampled (1X) and 4.8X undersampled reconstructions in the rotating phantom experiment. Single data points represent individual particles with the intensity being particle velocity ($\mu\text{m/s}$) in (a) and CNR in the stationary reference image in (b). Almost all cells are above the 45° dashed line, indicating increased CNRs in the undersampled image. Fast-moving particles and those with high initial contrast in the stationary phantom benefitted to a greater extent from undersampling.

32

Fig. 3-6. Velocity dependent improvement of particle CNR from the rotational phantom experiment. The percentage improvement is calculated as the difference in CNR, between the 4.8X undersampled and fully sampled images, divided by the average CNR of the two methods. Individual data points represent single particles, and the solid line is a linear regression. The positive slope shows an average of 6.1% increase in contrast per $\mu\text{m/s}$.

33

Fig. 3-7. In vivo single cell tracking with the variable density sampling scheme. In exemplary slices at different positions in the mouse brain (1st column), image details, indicated by the red rectangle, show representative examples of cells which were only detectable using the undersampled reconstruction (marked with red arrowhead). Fully sampled images (1X; 2nd column) are compared with the undersampled reconstructions. The k-space data of 2.4X reconstructions (3rd column) corresponded to one of the frames in Fig. 1b, which were further accelerated to reconstruct two 4.8X images (4th and 5th columns). Line profiles corresponding to the marked lines in the image details are shown in the 6th column. SNR were improved in the undersampled reconstructions. Improved contrast in one of the 4.8X frames indicated the cells were present for a short duration leading to their absence when the temporal blurring was severe.

35

Fig. 3-8. In exemplary slices at different positions in the mouse brain (1st column, same as in Fig. 3-7), image details, indicated by the red rectangle, show representative examples of cells which were only detectable using the DLLR undersampled reconstruction. Cells are not visible in the fully sampled images (2nd column, same as in Fig. 3-7). However, the cells are easily visible in the 4.8X DL + Low Rank reconstruction (3rd column, same as in Fig. 3-7), while the 4.8X Wavelet + TV reconstruction (4th column) did not enhance the cell contrast sufficiently for it to be detectable. Line profiles corresponding to the marked lines in the image details are shown in the 5th column. 36

Fig. 4-1. The pulse sequence diagram for 2D PEPI. The phase encoding gradient is applied after every other refocusing pulse. The frequency encoding gradient is also unbalanced. 43

Fig. 4-2. An example output of the optimized phase cycling scheme. Each coloured line shows the RF phases of one average. The proposed method has random like oscillations compared to the XY-16 phase cycling in black. 51

Fig. 4-3. The PEPI experiments on a phantom (a) were simulated with Bloch equation with 160° refocusing pulses and 16 ETL. The CPMG phase scheme (b) resulted in substantial banding artifacts along the phase encoding direction. The CPMG with composite refocusing pulses (c) slightly reduced the artifact. An XY-16 phase scheme (d) led to low intensity artifacts in the phase encoding direction. Each repetition from the optimized phase cycling scheme is shown in (e), where the artifacts varied with different RF phases. The average (f) of the complex images in (e) is comparable to (a) with the artifacts mostly removed. 52

Fig. 4-4. 3D PEPI experiments (16 ETL and 4 averages) with varying refocusing flip angles were performed with the proposed phase cycling scheme (1st row) and XY-16 (2nd row). The first phase encoding direction with blips applied every second echo is horizontal while the second phase encoding direction is vertical. The image quality was good at a nominal 180° for both methods (1st column). With a 170° nominal flip angle, coherence pathway artifacts were apparent in the XY-16 image (e), while the optimized method (b) remained relatively artifact free. A 160° refocusing pulse resulted in significant artifacts in both images (3rd column). The XY-16 image (f) quality was extremely degraded. 53

Fig. 4-5. 2D PEPI experiments (16 ETL and 4 averages) with varying refocusing flip angles were performed with the proposed phase cycling scheme (1st row) and XY-16 (2nd row). The phase encoding direction is horizontal while the second phase encoding direction is vertical. The image quality was good at a nominal 180° for the optimized method (a), while low intensity artifacts are observed with XY-16 (d). With a 170° nominal flip angle, coherence pathway artifacts were significant in the XY-16 image (e), while the optimized method (b) shows only minor artifacts. A 160° refocusing pulse resulted in artifacts in both images (3rd column) where the artifacts were of much higher intensity in the XY-16 (f). 55

Fig. 4-6. 2D PEPI image of a mouse brain using the optimized phase cycling scheme (1st column) and XY-16 (2nd column). The 1st row is the slice at the gradient isocenter and the 2nd row is offset 2mm from the isocenter. Significant coherence pathway artifacts are present in XY-16. 2D FSE images (3rd column) had reduced quality. Additional blurring away from the gradient isocenter was observed, as highlighted in (f), due to the concomitant magnetic field, while features close to the gradient isocenter were less degraded, such as highlighted in (c).

56

Fig. A1-1. A pulse sequence diagram for the first three pulses in a PEPI experiment with non-ideal refocusing pulses. Numbers (1-9) indicate time points of interest at which the coherence pathways are calculated with the partition method.

68

NOMENCLATURE

CNR	Contrast-to-Noise Ratio
CS	Compressed Sensing
DL	Dictionary Learning
ES	Echo Spacing
FSE	Fast Spin Echo
GRE	Gradient Echo
ION	Iron-Oxide Nanoparticles
LR	Low Rank
MTF	Modulation Transfer Function
OMP	Orthogonal Matching Pursuit
PEPI	π Echo Planar Imaging
PSF	Point Spread Function
RF	Radiofrequency
SNR	Signal-to-Noise Ratio
TE	Echo Time
TR	Repetition Time
TV	Total Variation

CHAPTER 1

Introduction

1.1 Overview of Work

Magnetic Resonance Imaging (MRI) is a non-invasive imaging modality which provides high resolution images with a flexible soft tissue contrast. There is a wide variety of MRI applications, such as cell tracking and porous media imaging. The MRI image properties, including contrast and resolution, are determined by the sequence and timing of radiofrequency (RF) and magnetic field gradient pulses, termed the MRI pulse sequence. A mathematical operation is required to transform the MRI signals to image domain. This thesis focuses on optimization problems in both MRI pulse sequence design and image reconstruction.

Recent increases in computation power have enabled new MRI techniques, such as Compressed Sensing (CS) [1]. CS allows data to be sampled below the Nyquist criteria leading to faster image acquisition [1]. CS could increase the temporal resolution, which is especially important for systems with motion. One example is cell tracking, in which cells are tagged with Iron Oxide Nanoparticles (ION), distorting the magnetic field around the cells allowing individual cells to be tracked [2]. A long acquisition time results in motion blurring so that it is difficult or impossible to detect fast moving cells. Cell tracking can benefit from CS. Choosing an appropriate regularizer and maintaining low contrast features are two major challenges in CS. Chapter 3 explores an adaptive regularizer for cell tracking MRI. A variable undersampling ratio cartesian sampling scheme is proposed to ensure low contrast stationary cells are visible in the CS reconstruction.

The MRI pulse sequence design can be considered as an optimization problem. The Pi Echo Planar Imaging (PEPI) pulse sequence improved upon the Fast Spin Echo (FSE) sequence with more efficient use of gradients, reducing concomitant magnetic field and eddy current effects [3,4]. These are particularly advantageous for high resolution and/or low magnetic field imaging. Low field MRI scanners have the potential to reduce cost and improve accessibility. Current PEPI experiments require a very high quality π pulse to prevent coherence pathway artifacts [3]. This has limited past PEPI experiments to samples restricted to the homogeneous region of the RF coil. In Chapter 4, a phase cycling scheme is determined by solving an optimization problem to minimize the coherence pathway artifacts. This leads to good quality images with non-ideal refocusing RF pulses. The slice selective PEPI experiment is enabled with this phase cycling scheme.

1.2 Summary of Master's Work

During my master's, I programmed, tested, and debugged MRI pulse sequences on a 1T Aspect M2 small animal scanner using a MATLAB interface. The relevant sequences for this thesis were a 2D and 3D PEPI experiments as well as a 3D FSE. In addition, I created a 3D multi-echo-spin echo (3DMESE) and a Gradient And Spin Echo (GRASE). I also collaborated on creating our Carr-Purcell-Meiboom-Gill (CPMG), Saturation Recovery (SR), SR-CPMG and inversion recovery (IR) sequences.

I wrote a Python script to generate the gradient table in Chapter 3, as well as Matlab and python code to reconstruct and process the images. I performed the phantom simulations in Matlab (motion artifacts) and Python (Bloch equation) in Chapters 3 & 4. I developed the cost function in Chapter 4 and wrote a Python code to execute the

optimization of coherence pathway cancellation. In addition to the work included in this thesis, I also programmed an optimized T_2 mapping script which processed the T_2 mapping experiment data in an order of magnitude less time than our previous script. I included a simple modulation transfer function (MTF) correction for T_2 mapping with FSE experiments. I also collaborated on a T_2 mapping algorithm which used an optimization for MTF correction.

I have presented my work at many conferences including Canadian Association of Physicists (CAP) conference, Imaging Network Ontario (ImNO), International Society for Magnetic Resonance in Medicine (ISMRM) Annual Meeting and Magnetic Resonance in Porous Media (MRPM).

1.3 Thesis Outline

Chapter 1 introduces the work performed for this thesis.

Chapter 2 provides an overview of MR physics and the basic imaging experiments.

Chapter 3 presents the CS method developed for improving the temporal resolution of cell tracking MRI.

Chapter 4 describes the optimized phase cycling algorithm and its application to PEPI experiments.

Chapter 5 summarizes the work and discusses future research directions for these projects.

CHAPTER 2

Magnetic Resonance Imaging Basics

2.1 Nuclear Magnetic Resonance

This chapter mostly follows the reference [5] and considers non-interacting ^1H nuclei.

Nuclear magnetism originates from the microscopic magnetic fields associated with nuclear spin. These magnetic moments are vector quantities which, in the presence of an external magnetic field \mathbf{B}_0 , precess about the field at a frequency

$$\omega_0 = \gamma B_0, \quad (2.1)$$

where γ is a physical constant for the chosen nuclei. The macroscopic bulk magnetization is a vector \mathbf{M} summed over all nuclear magnetic moments. At thermal equilibrium, the bulk magnetization aligns in the direction of the static field \mathbf{B}_0 , which is conventionally in the z-axis ($M_z = M$, $M_x = M_y = 0$).

The M_z component of the magnetization is difficult to measure, while the transverse component ($M_{xy} = M_x + iM_y$) precesses at a rate defined by Eqn. 2.1. The precession induces a voltage in a pickup coil, which is the MR signal. To generate a nonzero M_{xy} magnetization, an RF pulse is applied to rotate the thermal magnetization from the z-axis into the xy-plane. Consider a circularly polarized magnetic field with carrier frequency $\omega_{rf} = \omega_0$ (on-resonance) and phase φ :

$$\mathbf{B}_1(t) = B_1^e(t) [\cos(\omega_{rf}t + \varphi)\mathbf{i} - \sin(\omega_{rf}t + \varphi)\mathbf{j}], \quad (2.2)$$

where $B_1^e(t)$ is the envelope function or amplitude of the RF field over time. In a reference frame rotating at ω_{rf} about the z-axis, the RF magnetic field is stationary and in the xy-plane with its direction defined by φ . This results in a rotation of the bulk magnetization about an axis in the xy-plane. The creation of M_{xy} from thermal equilibrium is termed signal excitation.

The frequency of rotation under an RF pulse can also be described by Eqn. 2.1, replacing B_0 with $B_1^e(t)$, which is generally time dependent

$$\omega_1(t) = \gamma B_1^e(t). \quad (2.3)$$

The total angle of the rotation α from an RF pulse of duration t , and envelope $B_1^e(t')$, can be calculated by the integral

$$\alpha = \gamma \int_0^t B_1^e(t') dt'. \quad (2.4)$$

α is often called flip angle. Consider the magnetization immediately after an α degree RF-pulse. The z-component is $M_z(0) = M_0 \cos(\alpha)$, while the amplitude of the transverse component is $|M_{xy}(0)| = M_0 \sin(\alpha)$. There are two relaxation times which describe how the magnetization returns to thermal equilibrium. The T_1 describes the longitudinal or M_z component

$$M_z(t) = M_0 - (M_0 - M_z(0))e^{-\frac{t}{T_1}}. \quad (2.5)$$

T_2 describes the magnitude of the transverse or M_{xy} magnetization as it returns to equilibrium

$$|M_{xy}(t)| = |M_{xy}(0)|e^{-\frac{t}{T_2}}. \quad (2.6)$$

Note that the precession frequency can vary slightly due to ΔB_0 , including static magnetic field inhomogeneity, susceptibility, etc. $\rho(\omega)$ is the frequency distribution of the magnetization density, called spectral density. The term isochromat is used to describe all spins precessing at the same frequency. The signal $S(t)$ is an integral over spectral density with a phase term to account for the precession frequency:

$$S(t) = \sin(\alpha) \int_{-\infty}^{\infty} \rho(\omega) e^{-i\omega t} e^{-\frac{t}{T_2}} d\omega. \quad (2.7)$$

The additional phase term leads to dephasing and signal loss in addition to the T_2 decay, reducing the signal lifetime to T_2^* . The signal can be acquired after the RF pulse, which is termed Free Induction Decay (FID). The signal loss due to ΔB_0 dephasing can be valuable in certain situations where contrast related to the field is desired. For example, in the cell tracking experiments in Chapter 3, ION tagged cells cause strong susceptibility effects around them resulting in rapid intravoxel dephasing so that they appear dark in the image. In other situations, a long signal lifetime is desirable, such as in PEPI experiments where refocusing RF pulses are employed to reverse the effects of dephasing, yielding a $1/T_2$ decay rate.

2.2 Magnetic Resonance Imaging

2.2.1 Slice Selection

MR measures signal proportional to the bulk xy-magnetization, but spatial information must be encoded to reconstruct an image. A magnetic field gradient \mathbf{G} creates a position dependent frequency. It is important to note that the gradient field is in the same direction as the static magnetic field. It is the magnitude of the gradient field which linearly varies over space.

At low flip angles, the frequency domain response of an RF pulse is related to the Fourier transform of the RF field. For example, an RF pulse with a sinc shaped envelope excites a box function in the frequency domain. By applying a gradient, the frequency varies linearly with spatial position. In 2D MRI imaging, a slice selection gradient is applied at the same time as an RF pulse to excite a range of frequencies and therefore a slice of the subject. A stronger slice selection gradient and/or a broader RF pulse can be used to select a narrower slice. Another important consideration for slice selective pulses is the slice profile. In practice, a pulse cannot be infinitely long, so a true sinc pulse is not possible. As a result, a perfect box function in the frequency domain is not achieved and the flip angle varies across the slice. In the case of a truncated sinc pulse the slice profile is a box with ringing at the edges. For Shinnar–Le Roux (SLR) pulses [6] there is a flat top with a smooth transition to zero. For most experiments this is not an issue. For experiments that are highly sensitive to flip angle such as the PEPI to be discussed in Chapter 4, it is however very challenging to perform 2D slice selective experiments.

2.2.2 Spatial Encoding

The precession frequency in the presence of a gradient field can be written as a function of position as:

$$\omega(\mathbf{r}) = \gamma(B_0 + \mathbf{r} \cdot \mathbf{G}). \quad (2.8)$$

In the rotating frame of reference, the B_0 term is removed. Ignoring relaxation effects and assuming all the magnetization is excited into the xy-plane, the signal can be written as

$$S(\mathbf{k}) = \int_{-\infty}^{\infty} \rho(\mathbf{r}) e^{-i2\pi\mathbf{r} \cdot \mathbf{k}} d\mathbf{r}, \quad (2.9)$$

where the gradient effect at time t is defined by

$$\mathbf{k} = \frac{\gamma}{2\pi} \int_0^t \mathbf{G}(t') dt'. \quad (2.10)$$

Clearly the measured signal $S(\mathbf{k})$ is related to the proton density $\rho(\mathbf{r})$ by the Fourier transform relationship. The proton density can then be expressed in terms of the signal as:

$$\rho(\mathbf{r}) = \int_{-\infty}^{\infty} S(\mathbf{k}) e^{i2\pi\mathbf{r}\cdot\mathbf{k}} d\mathbf{k}. \quad (2.11)$$

This equation can be discretized, so an image with pixel intensity $\rho(\mathbf{r})$ can be obtained by appropriately sampling $S(\mathbf{k})$ and applying the inverse discrete Fourier transform.

In practice, relaxation effects are present during imaging. As a result, the image reconstructed is generally not $\rho(\mathbf{r})$ but a certain combination of ρ , T_1 , and T_2 , determined by the MRI pulse sequence.

2.2.3 K-space Trajectory

In 2D slice selective experiments, there are two spatial dimensions which need to be encoded with gradients. The frequency encoding gradient and phase encoding gradients are usually applied in orthogonal directions. The frequency encoding gradient stays on while k-space data points are being acquired. During the acquisition, the integrated area of the gradient increases, so that the \mathbf{k} value increases in the frequency encoding direction and a line of k-space is read out. The phase encoding gradient is applied prior to the acquisition setting the \mathbf{k} value in the phase encoding direction. Multiple acquisition windows are required to cover the 2D Cartesian k-space line by line. This is called spin warp imaging. In the case of 3D imaging, a second phase encoding gradient is added.

The signal at each acquired point has some weighting of proton density and different relaxation effects. The image contrast is typically defined by the signal expression at $\mathbf{k} = \mathbf{0}$ or the k-space center, so MRI has great flexibility in the contrast as the location of the k-space center can be chosen. The k-space trajectory is determined by the order the k-space points are acquired in. Based on the pulse sequence parameters and k-space trajectory, the amplitude modulation of k-space data (modulation transfer function) due to relaxations can lead to a low pass or a high pass filter in the image, corresponding to blurring and edge enhancement, respectively. The true image resolution is altered. The k-space trajectory also affects the appearance of image artifacts, such as due to motion or unwanted coherence pathway. These artifacts will be discussed in detail in Chapters 3 & 4.

2.3 MRI Pulse Sequences

2.3.1 Gradient Echo

A 2D Gradient Echo (GRE) pulse sequence is one of the simplest MRI experiments [7]. A slice selective RF pulse and slice select gradient (G_s) are applied simultaneously, as shown in Fig. 2-1. This excites the magnetization within a 2D plane which is then spatially encoded in the frequency and phase directions. The negative lobe of the frequency encoding gradient (G_f) brings the \mathbf{k} vector to $-k_{max}$ along the frequency encoding direction so that a whole line is read out during the acquisition. At the same time a phase encoding gradient (G_p) is applied. The ladder shown in Fig. 2-1 indicates that different amplitudes are used when the RF pulse is repeated, so that different lines in k-space are read out. A frequency encoding gradient with a duration of twice the preparation gradient

is on while the k-space points are acquired. This reads out a line from $-k_{max}$ to 0 in the center and to k_{max} at the end. The gradient echo occurs at the point where $k_f = 0$.

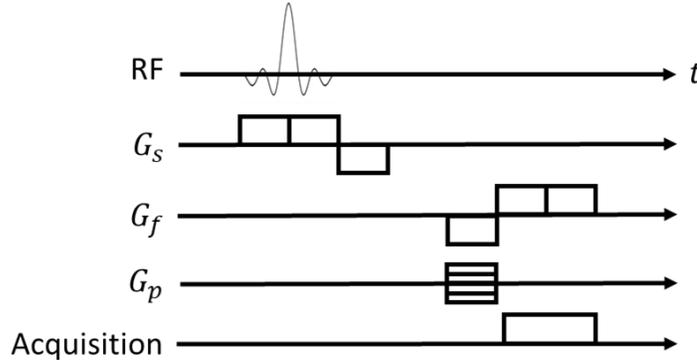


Fig. 2-1. The pulse sequence diagram for a 2D GRE. A slice select gradient (G_s) and RF pulse are used to excite signal in a plane. Frequency and phase encoding gradients (G_f and G_p) are used to encode the remaining two directions. The excitation is repeated with different phase encoding gradient strengths until the full Cartesian k-space is covered.

The FID signal decays from T_2 relaxation as well as dephasing due to ΔB_0 , at a rate of $1/T_2^*$. The time between the RF pulse and the center of the acquisition block ($\mathbf{k} = 0$) is called TE, because it is the “time to echo”. The T_2^* contrast can be chosen based on TE. Increasing T_2^* weighting can be useful in some experiments such as cell tracking, where the additional dephasing around ION tagged cells leads to lower signal compared to neighboring pixels. There is a trade-off in that the signal decreases with more T_2^* decay, reducing Signal-to-Noise Ratio (SNR) and resulting in a grainier image. These considerations must be balanced carefully when choosing the optimal parameters.

There is also contrast from proton density and possible T_1 relaxation. The signal excitation is repeated multiple times for the different phase encodings, with a delay time between repetitions called TR. The M_z magnetization recovers at a rate of T_1 during TR, which is generally not complete. The flip angle, TR and T_1 determine M_z at the start of the next RF pulse and therefore the signal amplitude.

2.3.2 Spin Echo and Fast Spin Echo

A simple variation on the GRE is to add a refocusing pulse [8] as shown in Fig. 2-2. The isochromats dephase due to ΔB_0 during a time period $\frac{TE}{2}$ between the excitation and 180° refocusing pulse. The 180° refocusing pulse flips the spins and they rephase generating a spin echo when the ΔB_0 dephasing is completely reversed at TE. The frequency encoding gradient is arranged so that the gradient echo and spin echo overlap. Since acquisition is centered on the spin echo, the image has a T_2 contrast instead of a T_2^* contrast. The TE and TR are defined similarly to the GRE.

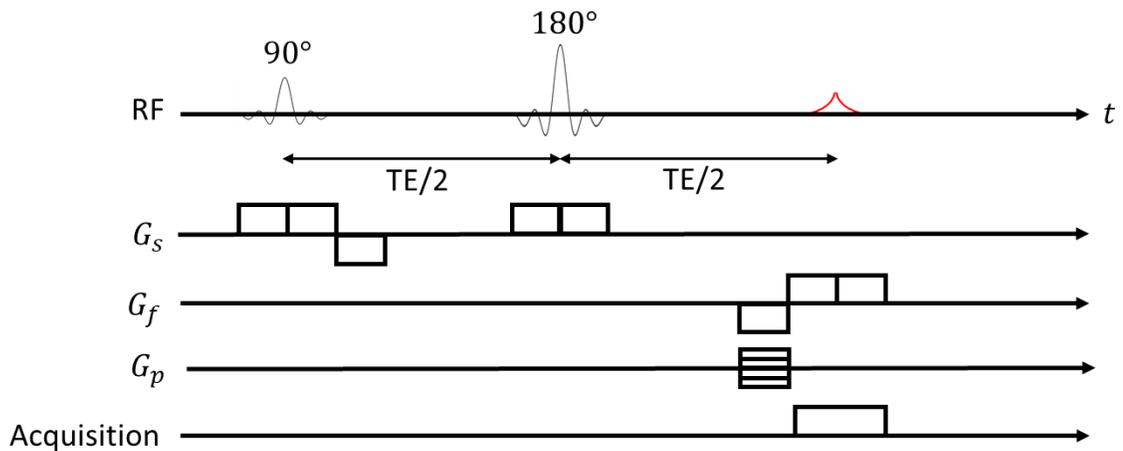


Fig. 2-2. The pulse sequence diagram for a 2D SE. A slice selective excitation pulse is used to excite signal in a plane similar to a GRE. A slice selective refocusing pulse is then applied to create a spin echo. The spatial encoding is applied similar to a GRE and the timing is set so the gradient echo and spin echo align.

Fast spin echo (FSE) [4] modifies the SE sequence by adding a chain of 180° refocusing pulses, where an echo occurs after each refocusing pulse, as shown in Fig. 2-3. This series of spin echoes is called an echo train. Typically, the CPMG phase scheme is used, where the phase of the refocusing pulse is offset by 90° from the excitation pulse. One line of k-space is acquired on each echo, so that multiple lines are traversed with a single excitation leading to the faster image acquisition. The T_2 decay along the echo train

results in an MTF in the phase encoding direction. The effect of the MTF depends on how the lines of k-space are arranged. The echo spacing (ES) is the time between two consecutive echoes, which is also the time between the excitation and first spin echo. The TE is the time between the excitation and the point where $k_f = 0$ and $k_p = 0$. This could happen at a later echo, where the TE and ES are not equal.

In conventional FSE experiments balancing gradients are applied to return to the k-space center prior to the next refocusing pulse as shown in Fig. 2-3. This is important to prevent artifacts from coherence pathways when the 180° refocusing pulse is imperfect, as will be discussed in detail in Chapter 4.

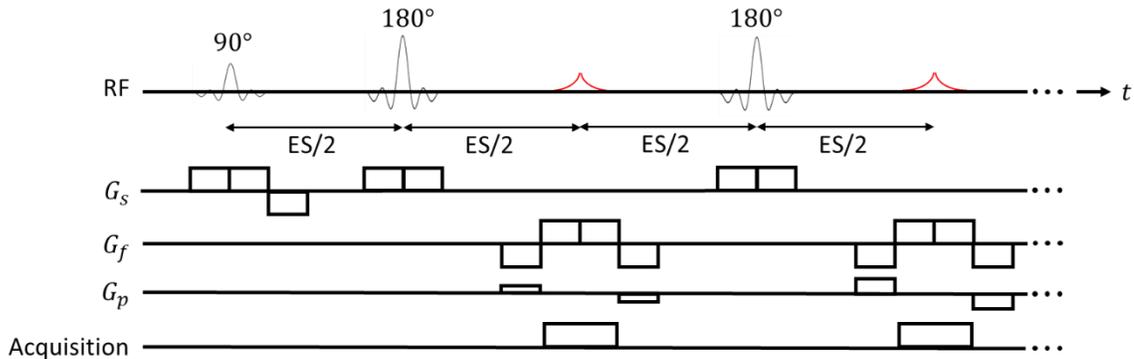


Fig. 2-3. The pulse sequence diagram for a 2D FSE. The experiment is similar to an SE except that a chain of refocusing pulses is used instead of one. This generates a train of echoes with different spatial encodings. The gradients are balanced within each echo interval so that the net phase accumulation is 0.

2.3.3 PEPI

Some sequences, including BLIPPED and PEPI, have shown that gradients can be accumulated throughout the echo train if the refocusing pulses are sufficiently close to 180° [3,9]. Since all phase effects are reversed, the \mathbf{k} vector changes to $-\mathbf{k}$. A 3D PEPI experiment is shown in Fig. 2-4. It has a similar structure of the FSE with a chain of

refocusing pulses and spatial encoding for each echo. The slice selection has been replaced with a second phase encoding gradient, which is applied only after the excitation pulse. The phase 1 encoding gradient is applied before every other echo. The frequency encoding gradient is also unbalanced. Other k-space trajectories are also possible. Generally, it is preferred to avoid large jumps in the magnitude of each k-space coordinate so that only small amplitude gradients are required.

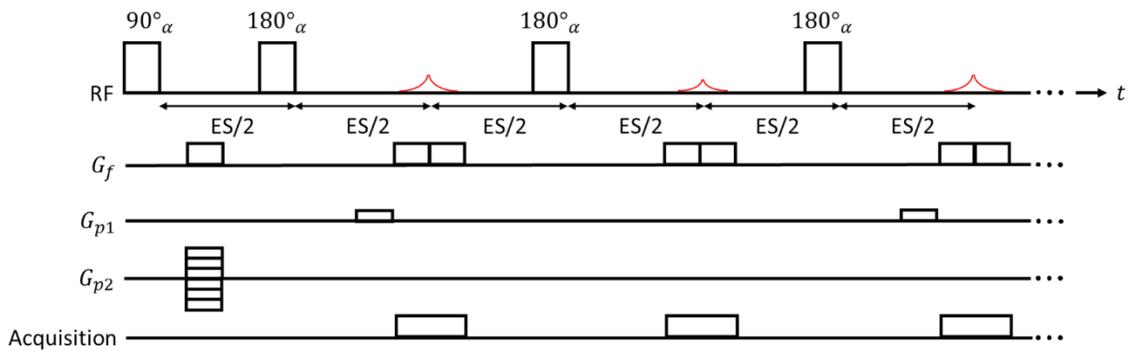


Fig. 2-4. The pulse sequence diagram for a 3D PEPI. The first phase encoding gradient is applied before every other echo. The second phase encoding gradient is applied only after the excitation RF pulse. The 180° refocusing RF pulse changes k to -k. Other k-space trajectories are possible.

Compared to FSE, PEPI reduces gradient switching, gradient amplitude, and gradient duty cycle. This can result in improved image quality due to the reduced eddy current and concomitant magnetic field effects. Concomitant magnetic fields are gradients in the transverse direction that create a spatially dependent frequency variation. This effect is particularly significant in the case of low field or high resolution as the frequency variation introduced is inversely proportional to B_0 and grows quadratically with the magnetic field gradient [10]. The requirement for a near perfect 180° refocusing pulse in PEPI is however quite restrictive. The PEPI experiments can only be performed on small samples within the homogeneous region of the B_1 field and slice selective 2D experiments have not been feasible. In Chapter 4, a method of reducing coherence pathway artifacts is explored to relax the constraint on the refocusing pulses.

CHAPTER 3

Dynamic Cell Tracking MRI with Variable Temporal Resolution Cartesian Sampling

3.1 Introduction

Dynamic single cell tracking would be of great benefit to diagnostic imaging in inflammation or cancer. Among the available imaging modalities for this purpose, MRI stands out, due to its non-invasiveness, absence of ionizing radiation, excellent soft tissue contrast and comparatively high spatial resolution [2,11,12]. However, for cellular resolution, MRI only affords limited contrast and sensitivity. This limitation makes a potent cell labeling mandatory, which can be achieved with high efficiency by using superparamagnetic iron-oxide nanoparticles (ION) [12–15]. Such labeling enables detection of single cells as hypointense spots in T_2^* weighted images, both in vitro and in vivo [16–18]. Despite the need for scan times of several minutes, which are required to reach sufficiently high spatial resolutions to visualize individual cells, observation of motion of single cells in vivo with frame rates of down to eight minutes is possible by using time lapse MRI [19,20]. Since image contrast is determined by the peculiarities of k-space, feasible frame rates previously translated into a limit of detection of 1 $\mu\text{m/s}$ for the speed of cells [20]. Faster moving cells could not be detected due to temporal blurring. While this speed limit was sufficient to resolve a basic inflammatory response in vivo [21], higher temporal resolution is required for more detailed insight into inflammatory processes.

Temporal resolution in MRI can be improved by applying CS techniques [1] to reconstruct images with undersampled k-space data, which can be acquired in shorter scan times. CS reconstructs an image with data sampled below the Nyquist criteria by solving

an optimization problem which utilizes the sparsity of MR images. A fidelity term enforces consistency with the measurements, and additional regularization terms enforce sparsity in some domain or domains. A sparsifying transform can be adaptive such as Dictionary Learning (DL) [22] or generic like the wavelet transform and total variation (TV) [1]. Typically, one uses either an l_1 -norm [1] or an approximation of an l_0 -norm [23,24] to promote the transform sparsity [1].

Recently, DL has been explored as an adaptive CS regularizer [23–25]. DL represents patches of an image using sparse combinations of so-called “atoms” in a dictionary [26,27]. This dictionary can be predefined or trained from the target image with K-Singular Value Decomposition (K-SVD) [26]. The sparse representation can be achieved through Orthogonal Matching Pursuit (OMP) [26]. DL can achieve a higher sparsity level than generic transforms, like wavelet, because it is trained to represent a certain class of images such as mouse brains, for example. Additionally, the dictionary may be overcomplete allowing a higher level of sparsity compared to orthogonal transformations. A higher sparsity level generally permits images to be reconstructed with a higher undersampling ratio. l_2 -norm regularization has been employed in DL CS [24]. Caballero [23] applied a 3D spatial-temporal dictionary for dynamic MR reconstructions with an l_1 norm term on the derivative (total variation) along the temporal dimension. Real and imaginary parts of the images were processed separately using the same real-valued dictionary.

Nuclear norm, the sum of singular values of a matrix, has also been explored to promote a low rank solution. This is of particular interest for dynamic MRI [28], where

redundancy in singular values is present in the spatial-temporal matrix. A nearly stationary subject can be approximated with a very low rank spatial-temporal image.

Since incoherent data acquisition is essential for CS, different sampling schemes [29–31] have been developed. The undersampling pattern is usually pre-defined based on the achievable acceleration ratio, which is determined by the sparsity, SNR and image feature contrast. Since these factors might change between subjects and scans, it is advantageous to employ a scheme that permits flexible retrospective undersampling. Recently, the golden angle radial scheme [32] was used for undersampling, by allowing an arbitrary number of radial spokes for each temporal frame [29]. However, radial trajectory does not cover the periphery of k-space with high density, where high spatial frequency features are encoded. Therefore, a radial scheme is suboptimal for single cell imaging that targets small signal voids which are only a few pixels in size.

In this work, a variable density Cartesian sampling scheme has been developed, which permits the reconstruction of fully sampled images and images at various undersampling ratios simultaneously [33], using a DL and low rank CS reconstruction. Simulations of a rotating circular phantom with signal voids of different intensities were performed. The method was implemented for cell tracking in phantom and in vivo mouse experiments. Improved detection limits in time lapse MRI single cell tracking experiments were demonstrated.

3.2 Theory

3.2.1 Flexible Sampling Scheme

In the proposed sampling scheme, as shown in Fig. 3-1a, the 2D k-space is divided into a number of bins, determined by the highest acceleration factor. One random phase encoding line is sampled from each high frequency bin, combining with the central lines to form an undersampled frame with the highest acceleration factor. The low frequency central lines are acquired after half of the total number of frames. No duplicate phase encoding lines are sampled until the full k-space is covered. A smaller undersampling ratio can be chosen in the reconstruction stage by grouping a larger number of phase encoding lines, as shown in Fig. 3-1b. The undersampling ratio cannot be chosen arbitrarily with this scheme as the number of phase encoding lines is the number of central lines plus an integer multiple of the number of bins. However, it does allow sufficient flexibility, since the highest acceleration factor is chosen to be sufficiently large. Acquiring all the low frequency lines in the middle of the scan, rather than distributing them throughout the high temporal resolution frames, results in a reduced acquisition time for the high frequency dynamic frames. All lines can be combined to create a fully sampled image as shown in Fig. 3-1c. The phase encoding lines in each undersampled frame vary in successive fully sampled images, to ensure incoherence in the temporal dimension.

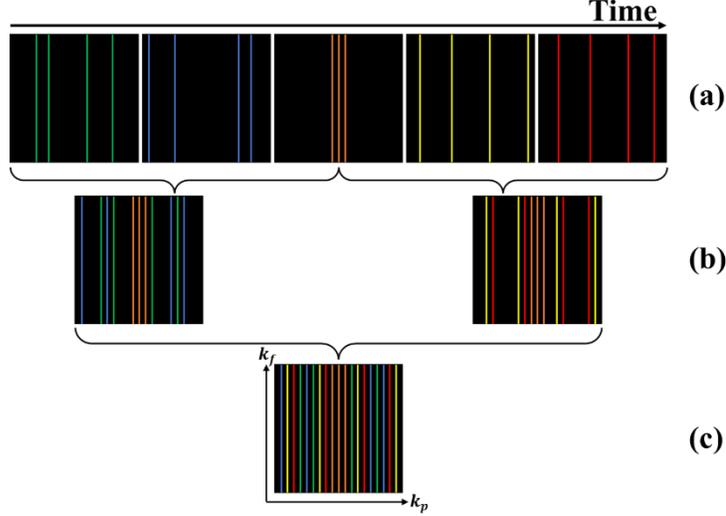


Fig. 3-1. Variable density sampling scheme. (a) 2D k-space is divided into bins of high frequency lines and a low frequency portion (orange) which is acquired in the center of the fully sampled data. Each of the high frequency groups is sampled in a short time interval, as indicated by the same color, and provides a high undersampling ratio frame with a relatively uniform incoherent k-space coverage. Sets of neighboring frames can be combined for reconstructions with (b) a lower undersampling ratio or (c) a fully sampled k-space.

3.2.2 Dictionary Learning and Low Rank Regularized Compressed Sensing

The DL CS method proposed in [23] was modified to enforce a solution which can be represented sparsely by a 2D spatial dictionary and is low rank along the temporal and phase encoding direction. 2D dictionary was chosen to reduce the computational burden. The flexible sampling scheme allowed a fully sampled image to be reconstructed for phase correction, so it was unnecessary to process the real and imaginary parts separately.

The dictionary learning and low rank (DLLR)-CS method is formulated as:

$$\min_{X, X_{LR}, \tau} \lambda_1 \sum_{ij} \|R_{ij}X - D\alpha_{ij}\|_2^2 + \lambda_2 \|X - X_{LR}\|_2^2 + \|y - F_u X\|_2^2 \quad (3.1)$$

$$s. t. \|\alpha_{ij}\|_0 \leq T \quad \forall ij, \quad \text{rank}(X_{jk,LR}) \leq r \quad \forall jk,$$

where X is the image to be reconstructed, and y is the acquired data in k-space. F_u is the undersampled Fourier transform operator. The indexes i, j , and k denote the frequency,

phase, and temporal dimensions, respectively. R_{ij} is the operator which extracts a 2D patch ij in the frequency and phase encoding dimensions. D is the dictionary, and $\tau = \{\alpha_{ij}\}$ is the set of coefficients for all patches. X_{LR} is the low rank approximation of the image. $X_{jk,LR}$ is a slice of the low rank image along the phase and temporal dimensions. λ_1 and λ_2 are parameters which determine the weighting of the DL and low rank regularizers, respectively.

The first term in Eqn. 3.1 and the first constraint enforce that the image patches have a sparse representation with T or fewer atoms under the predefined DL transform. The second term minimizes the difference between the image and X_{LR} , a stack of rank no larger than r spatial-temporal matrices $X_{jk,LR}$, as enforced by the second constraint. The third term enforces consistency of the image with the measurements. An adaptive regularizer was used along the spatial dimensions to ensure that image features corresponding to the tagged cells and mouse brain structure were well represented under the sparsifying transform. Example cells from the trained dictionary are shown in Fig. 3-2. The temporal dimension is much simpler as the brain is stationary, which should be approximately constant except for where a moving cell is present. Therefore, the low rank approximation was used along the temporal and phase encoding directions, which was also more computationally efficient than DL.

In this work, a pretrained dictionary was used, in contrast to optimizing the dictionary for each reconstruction, as each mouse was imaged in the same orientation leading to only little variation in mouse brain features. With the pretrained dictionary, one can be certain that the features corresponding to tagged cells are well represented, preventing them from being removed by the regularization.

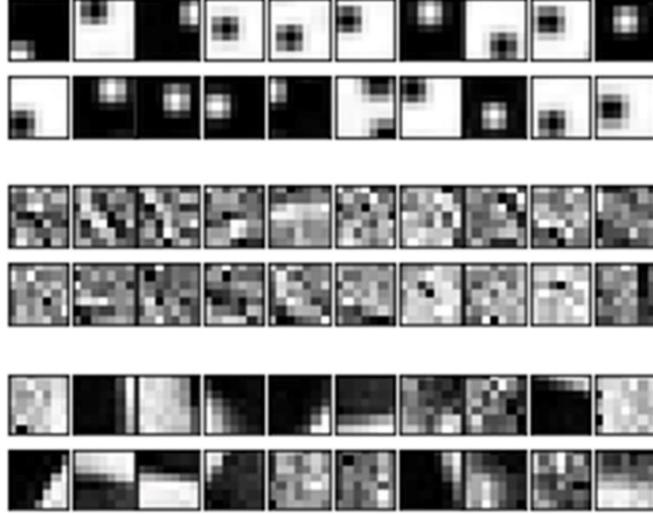


Fig. 3-2. Example atoms of simulated cell features (top 2 rows), trained contour features (middle 2 rows), and trained brain features (bottom 2 rows).

Eqn. 3.1 is solved by alternating minimization with respect to each variable while holding the others constant. The minimization w.r.t τ is:

$$\min_{\tau} \sum_{ij} \|R_{ij}X - D\alpha_{ij}\|_2^2 \quad s.t. \quad \|\alpha_{ij}\|_0 \leq T \quad \forall ij, \quad (3.2)$$

which is simply the dictionary learning problem with fixed dictionary. This is efficiently solved using the greedy OMP algorithm [26].

Minimization w.r.t X_{LR} yields:

$$\min \|X - X_{LR}\|_2^2 \quad s.t. \quad rank(X_{jk,LR}) \leq r \quad \forall jk, \quad (3.3)$$

where each $X_{jk,LR}$ is approximated by the truncated SVD of X_{jk} .

Minimization with respect to X results:

$$\min_X \lambda_1 \sum_{ij} \|R_{ij}X - D\alpha_{ij}\|_2^2 + \lambda_2 \|X - X_{LR}\|_2^2 + \|y - F_u X\|_2^2. \quad (3.4)$$

This problem is convex and can be solved by setting the first derivative to zero:

$$\begin{aligned}
& \lambda_1 \sum_{ij} R_{ij}^T (R_{ij}X - D\alpha_{ij}) + \lambda_2 (X - X_{LR}) - F_u^H (y - F_u X) = 0, \\
& \left(\lambda_1 \sum_{ij} R_{ij}^T R_{ij} + \lambda_2 I + F_u^H F_u \right) X = \lambda_1 \sum_{ij} R_{ij}^T D\alpha_{ij} + \lambda_2 X_{LR} + F_u^H y \\
& \left(\lambda_1 F \sum_{ij} R_{ij}^T R_{ij} F^H + \lambda_2 I + F F_u^H F_u F^H \right) FX \\
& = \lambda_1 F \sum_{ij} R_{ij}^T D\alpha_{ij} + \lambda_2 FX_{LR} + F F_u^H y
\end{aligned} \tag{3.5}$$

In the last step, $F^H F$ (the identity matrix) was inserted between $(\sum_{ij} R_{ij}^T R_{ij} + \lambda F_u^H F_u)$ and X , and F was multiplied on both sides to bring the expression into the Fourier domain. $F F_u^H F_u F^H$ is a diagonal matrix, with 1 corresponding to sampled k-space locations and 0 corresponding to unsampled points. $\sum_{ij} R_{ij}^T R_{ij}$ yields the identity matrix so that Eqn. 3.5 is trivially invertible. $\sum_{ij} R_{ij}^T D\alpha_{ij}$ is the patch averaged DL reconstruction, denoted by X_{DL} . One obtains the update step for X :

$$FX = \begin{cases} \frac{F(\lambda_1 X_{DL} + \lambda_2 X_{LR} + F_u^H y)}{1 + \lambda_1 + \lambda_2}, & \text{Sampled} \\ \frac{F(\lambda_1 X_{DL} + \lambda_2 X_{LR})}{\lambda_1 + \lambda_2}, & \text{Unsampled} \end{cases} \tag{3.6}$$

This result is similar to that presented by Caballero et al [23], with the weighted average performed on the DL reconstruction and the low rank approximation, instead of DL and temporal gradient reconstructions.

The initial guess for the image was chosen to be the nearest fully sampled magnitude image with added noise. Gaussian noise was used with a standard deviation of $1/15^{\text{th}}$ the max pixel intensity of the fully sampled image. This improved the chances that a slow-moving cell detectable in the fully sampled images would remain in the undersampled images, while the added noise allowed additional cells to be detected. It also prevented coherent artifacts in the undersampled image, should the undersampling scheme have insufficient incoherence. Coherent artifacts such as banding tend to be difficult to remove by regularization with small size dictionary atoms.

3.3 Methods

3.3.1 Simulations using a Virtual Phantom

To reproduce different contrasts and velocities observed in in vivo experiments, four classes of cells were simulated: fast-moving ($> 5 \mu\text{m/s}$) high-contrast (> 0.25) cells, fast-moving low-contrast (< 0.25) cells, slow-moving ($< 5 \mu\text{m/s}$) high-contrast cells, and slow-moving low-contrast cells. A virtual circular phantom with 10 spokes of cell feature was simulated, to test the methods ability to detect these different types of cells. The phantom had an intensity of 1 and the signal voids representing single cells had contrasts ranging from 0.1 to 0.4. Five cells were equally spaced on each spoke, as shown in Fig. 3-3a. The phantom was rotated at a rate of $2.93 \cdot 10^{-4}$ rotations/second resulting in the velocities listed in Table 1. The phantom was simulated with four times higher resolution in each dimension to preserve partial volume effects in the rotation and downsampled onto a 180×192 grid. With a resulting voxel size of $73 \times 73 \mu\text{m}^2$, the farthest cell had a linear velocity of $8.7 \mu\text{m/s}$. These parameters were chosen to match the phantom experiments

described in the next section. Noise was added to achieve an SNR of 50, leading to a Contrast-to-Noise ratio (CNR) range of 5 to 20 for the cell features. The k-space data was simulated with a TR of 649 ms. The central 32 lines were acquired in the middle of each fully sampled k-space data set. The remaining 160 phase encoding lines were divided into 8 high frequency bins resulting in acceleration factors of 9.6, 4.8, or 2.4, depending on how many lines per bins were combined in the reconstruction. As the low frequency lines were used in multiple frames, the undersampling ratio was defined as the total number of lines divided by the number of high frequency lines. To quantify the reconstruction quality, the number of pixels of each cell was evaluated by selecting a region around the void and thresholding the pixel values below a weighted average of the minimum value and the background intensity.

3.3.2 In Vitro Phantom Experiments

A cylindrical phantom (2-mL Eppendorf cap) was created, with microparticles of iron oxide (MPIOs; COMPEL™, Bangs Laboratories; mean diameter 8.2 μm ; 5000 particles/ml) suspended in agar gel (1%). The iron particles produced signal voids with contrast similar to the contrast of labeled immune cells both in vitro and in vivo in the mouse brain [20,21]. The agar was doped with 2 mmol/l gadolinium (Gadovist, Bayer AG, Germany) to achieve a similar SNR to the mouse brain images. As a reference, the phantom was first scanned in a stationary position. To create an in-plane velocity distribution of the iron particles based on the radial distance, the phantom was rotated along the cylindrical axis during the MR acquisition. The low rotation frequency of $2.93 \cdot 10^{-4}$ Hz, leading to a velocity range of 0-9.2 $\mu\text{m/s}$, was achieved by a custom-made device, driven by a stepper motor (ST4209S1006-B, Nanotec, Germany) with mounted planetary gear (GPLE40-3S-

512, Nanotec, Germany). Particle velocities were calculated based on the radial distance from the axis of rotation in the stationary image. For the rotating phantom, images were reconstructed using the fully sampled k-space data as well as for acceleration factors of 2.4, 4.8 and 9.6. A percentage improvement in particle CNR was calculated as the difference in CNR between the accelerated and fully sampled images divided by the average of the CNR for the two methods. Linear regression was performed using the `linregress` function in `scipy`'s `stats` module [34].

3.3.3 In Vivo Experiments

Animal experiments were carried out according to local animal welfare guidelines and were approved by local authorities (ID: T81-02.04.2020.A194). Female BALB/c mice ($n = 9$) were obtained from Charles River Laboratories (Sulzfeld, Germany) and housed under a 12 h light-dark cycle and provided with food and water ad libitum.

In vivo time-lapse MRI of the brain was performed 24h after in vivo labeling of cells by i.v. injection of 3 mL per kg body weight Ferucarbotran (Resovist, Bayer AG, Germany) via the tail vein. Mice were anesthetized with 1-1.5% isoflurane in 1 L per minute of oxygen and compressed air (20:80) under continuous respiratory and temperature monitoring. To avoid body cooling, mice were kept at physiologic temperature by a custom-designed animal heating device. Pronounced reduction of body temperature or breathing frequency despite lowering of anesthetic dose were stop criteria for the measurements. First, $n = 8$ mice were scanned to optimize MR acquisition and establish the variable sampling scheme. In vivo time-lapse MRI analysis was then performed on the optimized data for $n = 1$ mice.

Images were again reconstructed using the fully sampled k-space data and at acceleration factors of 2.4, 4.8 and 9.6 using DLLR reconstruction and of 4.8 for wavelet reconstruction for comparison.

3.3.4 MRI Acquisition

MRI was performed on a 9.4 T Biospec (Bruker Biospin, Ettlingen, Germany) using a cryogenic probe. A T_2^* weighted gradient echo sequence was used with these scan parameters: TR: 649 ms, TE: 8 ms, FA: 60°, averages: 1, frequency encodings: 180, phase encodings: 192, in-plane resolution: $61 \times 73 \mu\text{m}^2$, slices 38, slice thickness: 300 μm , scan time per fully sampled image: 2 min 6 s. 20 fully sampled timeframes were acquired for each in vivo experiment and 5 for each phantom experiment. In both phantom and in vivo measurements, individual cells were identified as hypointense spots. In addition, in vivo detected cells were categorized according to their motion behavior into short-term (one or two consecutive fully sampled timeframes), long-term short-range (three or more consecutive fully sampled timeframes) and long-term long-range cells (three or more consecutive fully sampled timeframes with in-plane motion of more than 1.5 pixels) [21].

3.3.5 Reconstruction

The DLLR-CS algorithm was programmed in Python. DL was performed using the scikit-learn minibatch dictionary learning module [35]. The DLLR-CS algorithm used a dictionary composed of 50 dot atoms for the cell feature, 200 brain atoms, and 100 residual atoms representing the contour of brain features. Fig. 3-2 shows some examples of these atoms. This combined dictionary was tested with mouse brain images containing tagged cells and the 100 most infrequently used atoms were removed from the dictionary. Using

4 atoms for the DL fit and 4 threads, the reconstruction took 5 hours 17 minutes with an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz and 16GB of RAM. The weightings of the DL term λ_1 and LR term λ_2 were 0.05 and 0.01, respectively. A rank 5 (r) SVD was used. For the wavelet + TV reconstruction, weightings for the wavelet and TV terms were 0.1 and 0.05, respectively. High pass filters were applied on the phantom and in vivo images after the DLLR-CS reconstruction. The filter was of the form $1 - A \exp[-(k_f^2 + k_p^2)]$, where k_f and k_p were linearly spaced from -0.5 to 0.5. $A = 0.87$ for 2.4X phantom and $A = 0.95$ for all the other reconstructions.

3.4 Results

3.4.1 Simulations

Simulated images of the virtual phantom were reconstructed with fully sampled and undersampled k-space data. The fully sampled image (1X) is shown in Fig. 3-3b, where the low-contrast fast-moving cells, such as the one highlighted in green (CNR 1.4), were not visible. The high-contrast cells in motion were distorted, such as highlighted in blue, which had a strongly reduced CNR of 4.2 compared to in the stationary image (CNR 20, Fig. 3-3a). An elongation of the cell in the phase encoding direction (horizontal) was observed in addition to the elongation in the direction of linear motion, if they differed, resulting in a “T” shape artifact (blue circle in Fig. 3-3b). A slow-moving low-contrast cell is highlighted in red, which was barely visible in the fully sampled image (CNR 2.9). CNR values for all simulated cells and all undersampling factors are listed in Table 1. The CNR was generally improved with CS, even when the stationary contrast was so low that the cells were virtually not discernible.

Table 1. The contrast to noise ratio for each cell in each reconstruction of the simulated phantom experiment in Fig. 3-3. The inner two cells (< 5 $\mu\text{m/s}$) in each spoke are considered slow, while the outer three (> 5 $\mu\text{m/s}$) are fast. Contrasts greater than 0.25 are categorized as high contrast, while the cells below 0.25 are low contrast.

Stationary Contrast	Velocity ($\mu\text{m/s}$)	Fully Sampled CNR	2.4X CNR	4.8X CNR
0.40	1.7	7.3	9.2	11.0
0.40	3.5	4.7	7.2	10.6
0.40	5.2	5.0	5.0	7.4
0.40	7.0	4.1	5.7	6.5
0.40	8.7	4.1	4.8	4.2
0.37	1.7	8.3	8.7	9.0
0.37	3.5	5.7	6.4	7.5
0.37	5.2	4.1	4.1	6.8
0.37	7.0	4.2	4.0	6.1
0.37	8.7	4.1	3.7	5.7
0.33	1.7	7.3	7.4	8.1
0.33	3.5	5.1	5.2	8.3
0.33	5.2	4.7	3.6	6.0
0.33	7.0	3.8	4.0	5.7
0.33	8.7	3.2	3.8	4.5
0.30	1.7	5.8	6.5	9.1
0.30	3.5	5.8	5.4	7.2
0.30	5.2	3.8	4.1	6.2
0.30	7.0	3.4	3.8	5.6
0.30	8.7	3.0	3.5	4.3
0.27	1.7	4.7	6.7	7.5
0.27	3.5	3.7	4.6	5.5
0.27	5.2	4.1	2.5	5.3
0.27	7.0	3.4	2.9	6.1
0.27	8.7	3.0	3.1	3.9
0.23	1.7	4.8	5.1	6.2
0.23	3.5	3.3	3.6	4.7
0.23	5.2	2.6	3.9	5.4
0.23	7.0	3.0	3.1	4.9
0.23	8.7	1.4	2.5	6.6
0.20	1.7	4.0	3.9	5.0
0.20	3.5	4.2	3.5	3.9
0.20	5.2	3.2	2.5	4.3
0.20	7.0	2.6	2.4	4.6
0.20	8.7	2.4	1.9	3.2
0.17	1.7	2.9	4.8	4.0
0.17	3.5	2.7	3.1	3.8
0.17	5.2	2.5	2.4	3.4
0.17	7.0	2.8	2.9	3.6
0.17	8.7	N/A	1.5	2.4
1.33	1.7	3.3	2.0	2.7
1.33	3.5	2.6	3.0	2.9
1.33	5.2	1.8	1.0	2.2
1.33	7.0	2.9	2.0	5.7
1.33	8.7	N/A	N/A	N/A
0.10	1.7	2.7	1.9	4.2
0.10	3.5	1.9	1.3	2.9
0.10	5.2	N/A	N/A	N/A
0.10	7.0	N/A	N/A	N/A
0.10	8.7	N/A	N/A	N/A

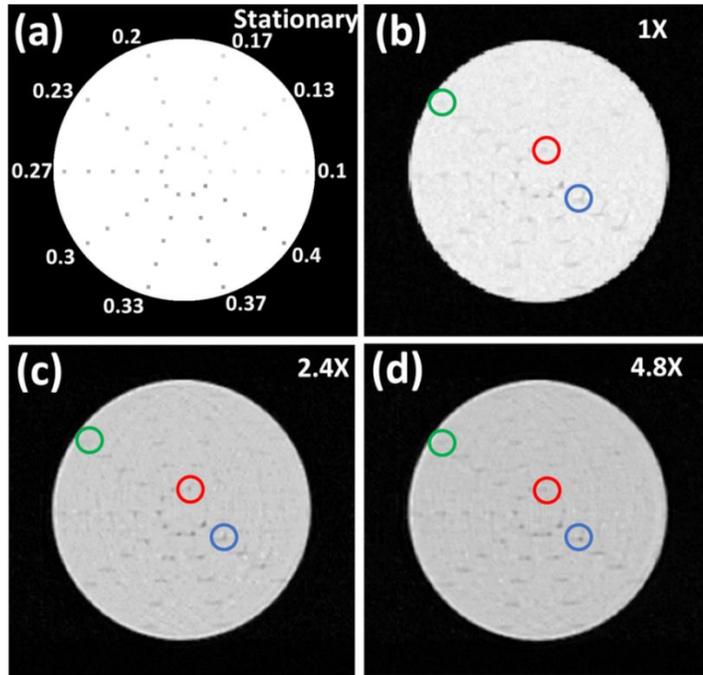


Fig. 3-3. Simulations using the variable density sampling scheme. A stationary phantom (a) was simulated containing cells represented by hypointense spots of varying contrasts as indicated at each spoke. The phantom was then rotated along the isocenter at a frequency of $2.93 \cdot 10^{-4}$ rotations/second to create different linear velocities ($1.7 \mu\text{m/s}$, $3.5 \mu\text{m/s}$, $5.2 \mu\text{m/s}$, $7.0 \mu\text{m/s}$, $8.7 \mu\text{m/s}$ at the five radial positions of the hypointense spots, respectively). The variable Cartesian sampling k-space data was processed to reconstruct the fully sampled (1X) image (b), 2.4X (c) and 4.8X (d) undersampled images. CNR for all cells in each reconstruction are listed in Table 1. Cells with various velocities and contrasts are highlighted for comparison showing blurring for fast-moving cells as well as reduction of the “T” shaped artifact and improvement in contrast in the accelerated reconstructions.

The “T” shaped motion artifact was reduced in the 2.4X undersampled DLLR-CS reconstruction (Fig. 3-3c) and further reduced in the 4.8X reconstruction (Fig. 3-3d). As a measure to assess the artifact size, the number of pixels of the cell feature in the 0.4 contrast spoke are compared for all the reconstructions in Table 2. Each void in the stationary image had 4 pixels. The number of pixels increased with increasing speed for all but the last void in the fully sampled image due to the significant reduction in contrast. Generally, the shape of moving cells was improved in the CS reconstruction where the cells spread over fewer pixels.

Table 2. The number of pixels for each cell along the 0.4 contrast spoke. It was evaluated by manually selecting a region around the cell. The number of pixels was determined by a threshold, which was a weighted average of the minimum intensity (X 0.4) and the background intensity (X 0.6).

Velocity ($\mu\text{m/s}$)	Fully Sampled Pixels	2.4X Pixels	4.8X Pixels
1.7	8	6	7
3.5	18	18	10
5.2	25	20	14
7.0	27	17	23
8.7	15	11	19

Contrast of the slow-moving cells was generally enhanced in the undersampled reconstructions. The implicit denoising also contributed to the CNR increase. This was true for low-contrast cells like the one highlighted in red in Fig. 3-3c, d, which had a CNR of 4.2 and 4.0 with 2.4X and 4.8X acceleration, respectively, as well as for high-contrast cells, especially with high acceleration ratio such as the cell in blue in Fig. 3-3c, d (CNR 4.9 in 2.4X and 10.2 in 4.8X). The constrained reconstruction effectively reduced the temporal blurring in the proximity of high contrast features. The fast-moving cells with sufficient contrast benefitted from the higher acceleration, including the relatively low contrast green cell in Fig. 3-3c, d (CNR 2.1 in 2.4X, CNR 3.1 in 4.8X). The 9.6X acceleration reconstruction results did not have a sufficient quality to allow for further analysis and is therefore not shown.

3.4.2 Phantom Experiments

The sampling scheme was applied in phantom measurements. Hypointensities over a few pixels similar to iron-labeled cells originated from micron sized iron particles suspended in agar gel. An image of the stationary phantom was acquired as reference, shown in Fig. 3-4a. The phantom was mechanically rotated at constant angular velocity,

and images were acquired with the variable density Cartesian scheme. Reconstructions were performed with the fully sampled k space data and varying undersampling ratio data. An acceleration factor of 9.6X led to poor quality reconstructions (results not shown). The fully sampled, and 2.4X and 4.8X undersampled images are shown in Fig. 3-4b, c, and d, respectively. The nonlinear CS reconstruction implicitly denoised the images, as a result of the regularization. High pass filters were applied to the undersampled images to compensate the blurring from DL patch averaging.

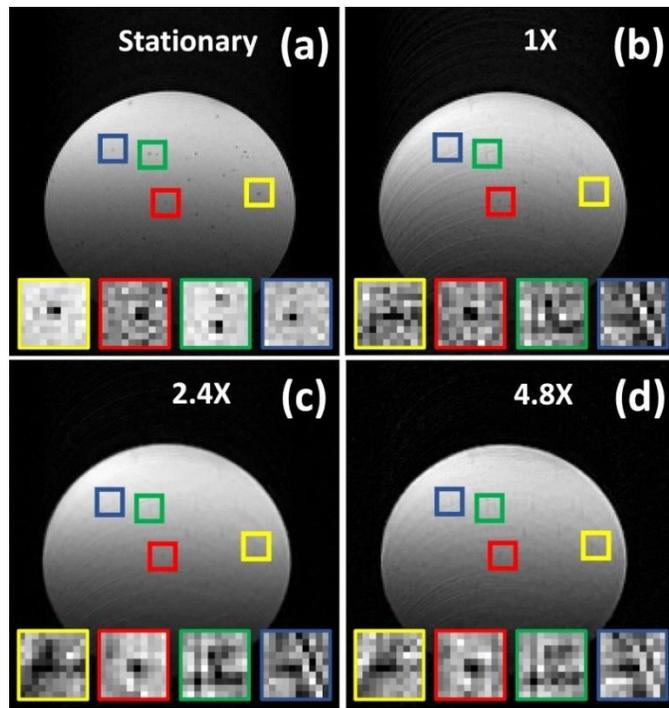


Fig. 3-4. Rotating phantom experiments with the variable density sampling scheme. An agarose phantom with embedded MPIOs was scanned in a stationary position (a) as a reference and for a rotation of $2.93 \cdot 10^{-4}$ rotations/second mimicking cells which speeds of up to $9.2 \mu\text{m/s}$. In the rotational case, the k-space data was processed to reconstruct the fully sampled (1X) image (b), 2.4X (c) and 4.8X (d) undersampled images. Zoomed image details with adjusted window/level settings and the coil sensitivity corrected show examples of particles with various velocities and contrasts. They demonstrate temporal blurring of fast-moving particles and improvement in contrast using undersampled reconstructions.

Some banding artifacts were observed while the phantom was rotating. The “T” artifacts, as observed in the simulations, were apparent in the fully sampled image. They were visually reduced in the undersampled images, similarly to the simulation results shown in Fig. 3-3.

A few areas are highlighted in the enlarged view with different window/level displays. The void in blue in Fig. 3-4 is one example of a relatively low-contrast high-linear-velocity void which was invisible in the 1X image (Fig. 3-4b, CNR 1.9, velocity 5.3 $\mu\text{m/s}$). It was improved in 2.4X (Fig. 3-4c, CNR 3.0) and visible in 4.8X (Fig. 3-4d, CNR 3.5). The void in yellow in Fig. 3-4 is an example of a high-contrast high-velocity cell which was visible in the fully sampled image (CNR 2.0, velocity 5.3 $\mu\text{m/s}$), although the shape was severely distorted. The CNR showed a slight improvement in 2.4X (CNR 2.2) and a large improvement in 4.8X (CNR 3.1). In red in Fig. 3-4, a low-velocity low-contrast void is highlighted. This was visible in 1X (CNR 2.5, velocity 0.4 $\mu\text{m/s}$) and had a CNR increase in 2.4X (CNR 3.3) due to the denoising. The void also had some CNR increase in 4.8X (CNR 3.1). In green in Fig. 3-4, two cells are observed in the stationary image (right velocity 3.8 $\mu\text{m/s}$, left velocity 3.9 $\mu\text{m/s}$). However, the two features cannot be distinguished in the fully sampled rotating image due to the severe “T” motion artifact (resulting void CNR 3.6). The motion artifact was reduced in 2.4X and two separate cells were discernable (right CNR 1.6, left CNR 3.4). The artifact was further reduced in 4.8X and the two separate voids could be identified more easily (right CNR 2.1, left CNR 5.5).

The CNR of cell features in different reconstructions were evaluated. Comparisons of CNR in 1X and 4.8X images are shown in Fig. 3-5. The intensity of individual data points indicates the cell velocity in Fig. 3-5a and CNR of stationary cells in Fig. 3-5b.

Almost all cells, except for one with very low contrast and velocity, had higher CNR in 4.8X than in 1X. Generally larger improvements were observed for the high velocity cells, agreeing with the severe temporal blurring in 1X. Low-contrast cells benefitted to a lesser degree in the CS reconstruction, as revealed in Fig. 3-5b. The fastest cell observed was at $7.0 \mu\text{m/s}$ with a CNR of 10.0 in the stationary image and a CNR of 3.2 in 4.8X, which was not visible in the fully sampled image.

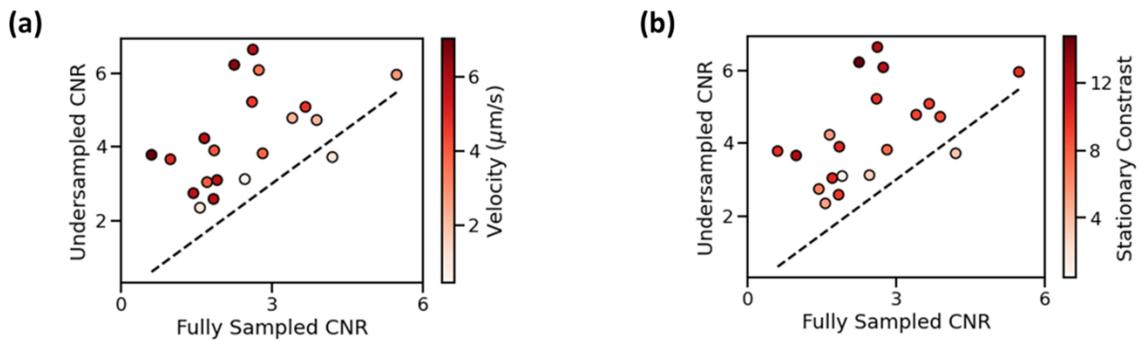


Fig. 3-5. Comparison of particle CNR in the fully sampled (1X) and 4.8X undersampled reconstructions in the rotating phantom experiment. Single data points represent individual particles with the intensity being particle velocity ($\mu\text{m/s}$) in (a) and CNR in the stationary reference image in (b). Almost all cells are above the 45° dashed line, indicating increased CNRs in the undersampled image. Fast-moving particles and those with high initial contrast in the stationary phantom benefitted to a greater extent from undersampling.

A significant percentage increase in CNR was observed for the 4.8X undersampled reconstruction at each velocity (Fig. 5, $p = 0.0021$, $R^2 = 0.42$). There were substantial variations in the degree of improvement for cells at similar velocities, which explained the low R^2 . Generally, more significant improvements were achieved for particles with higher stationary CNR. Yet, an average improvement of $6.1\% / (\mu\text{m/s})$ was obtained.

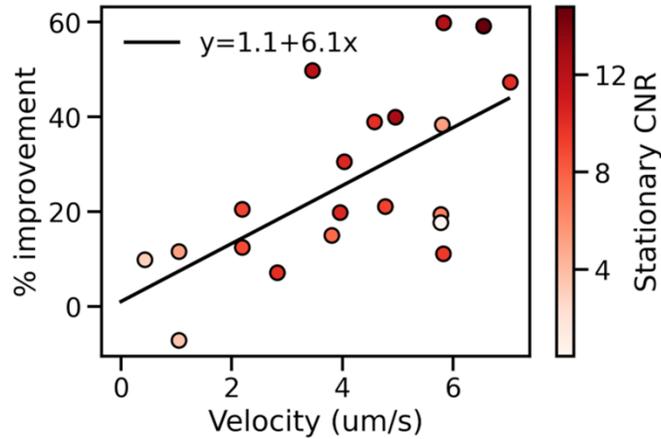


Fig. 3-6. Velocity dependent improvement of particle CNR from the rotational phantom experiment. The percentage improvement is calculated as the difference in CNR, between the 4.8X undersampled and fully sampled images, divided by the average CNR of the two methods. Individual data points represent single particles, and the solid line is a linear regression. The positive slope shows an average of 6.1% increase in contrast per $\mu\text{m/s}$.

3.4.3 In Vivo Experiments

The method was applied to in vivo cell tracking experiments. Fully sampled images as well as 2.4X and 4.8X accelerated reconstructions clearly showed individual cells with different motion behavior as hypointense spots. Overall, in one mouse brain a total of 245 cells were detected in the fully sampled images, out of which 92 (37.6 %) were short-term, 102 (41.6 %) long-term short-range, and 51 (20.8 %) long-term long-range cells. In both accelerated reconstructions the total number of cells remained nearly constant (244 in 2.4X; 242 in 4.8X). However, fewer long-term short-range and long-term long-range cells were observed (85 and 25 in 2.4X, 88 and 23 in 4.8X, respectively). Mainly small low-contrast cells could not be detected in the undersampled images anymore. On the other hand, the number of short-term cells increased (134 in 2.4X; 131 in 4.8X). Examples of these cells which were only visible using the undersampled reconstructions are shown in Fig. 3-7. Here, the fully sampled images appeared noisy, with no obvious cell features as shown in

the zoomed view in the 2nd column in Fig. 3-7. In the retrospectively 2.4X accelerated reconstructions, 3rd column, cells were discernible in the top two images but virtually absent in the bottom two images. SNR was improved in the undersampled reconstructions due to the regularization. In the 4.8X reconstructions (columns 4 and 5 in Fig. 3-7) that correspond to the same k-space data of column 3, a cell was visible in one of the frames for each image. The line profiles are shown to demonstrate the contrast differences. Improved contrast in one of the 4.8X frames indicated the cells were present in the slice plane for a short duration, leading to their absence when the temporal blurring was severe. The additional cells detected could be a result of cells moving quickly in the through-plane direction, or cells moving in-plane with variable velocity that were only sufficiently slow to be detected in one subframe.

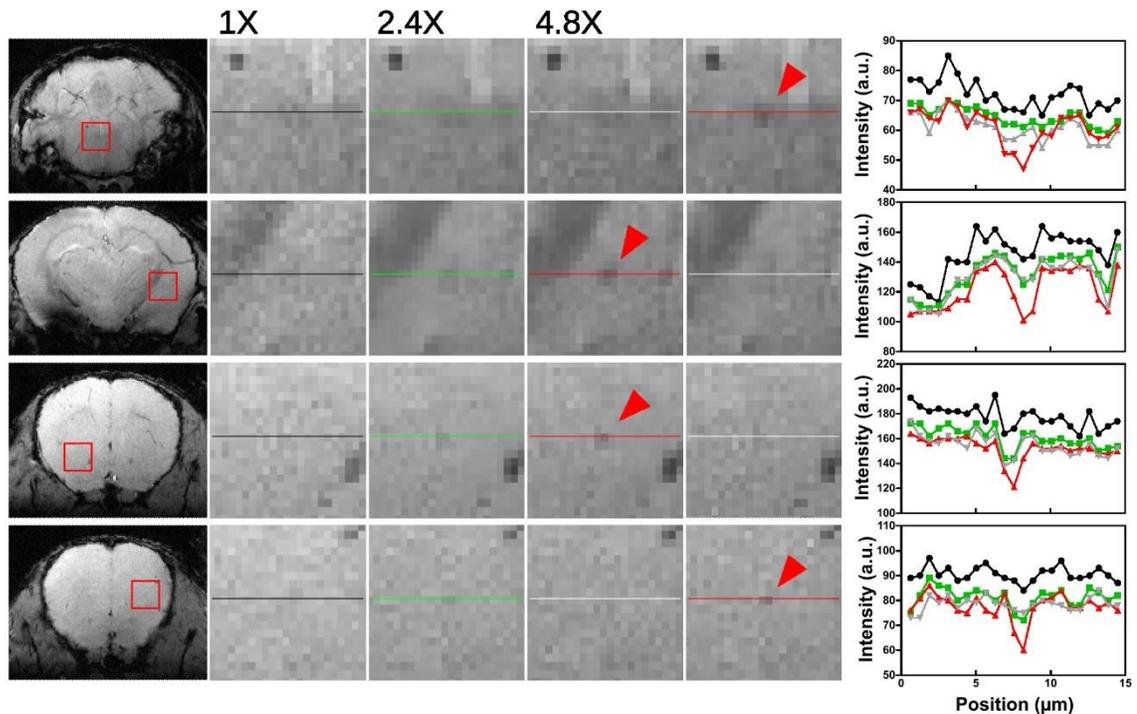


Fig. 3-7. In vivo single cell tracking with the variable density sampling scheme. In exemplary slices at different positions in the mouse brain (1st column), image details, indicated by the red rectangle, show representative examples of cells which were only detectable using the undersampled reconstruction (marked with red arrowhead). Fully sampled images (1X; 2nd column) are compared with the undersampled reconstructions. The k-space data of 2.4X reconstructions (3rd column) corresponded to one of the frames in Fig. 1b, which were further accelerated to reconstruct two 4.8X images (4th and 5th columns). Line profiles corresponding to the marked lines in the image details are shown in the 6th column. SNR were improved in the undersampled reconstructions. Improved contrast in one of the 4.8X frames indicated the cells were present for a short duration leading to their absence when the temporal blurring was severe.

The same data set was processed with spatial wavelet and spatial-temporal TV as the sparsifying transform, as shown in Fig 3-8. The cells highlighted in Fig. 3-7 were not detected, confirming the significance of a proper constraint in the CS reconstruction.

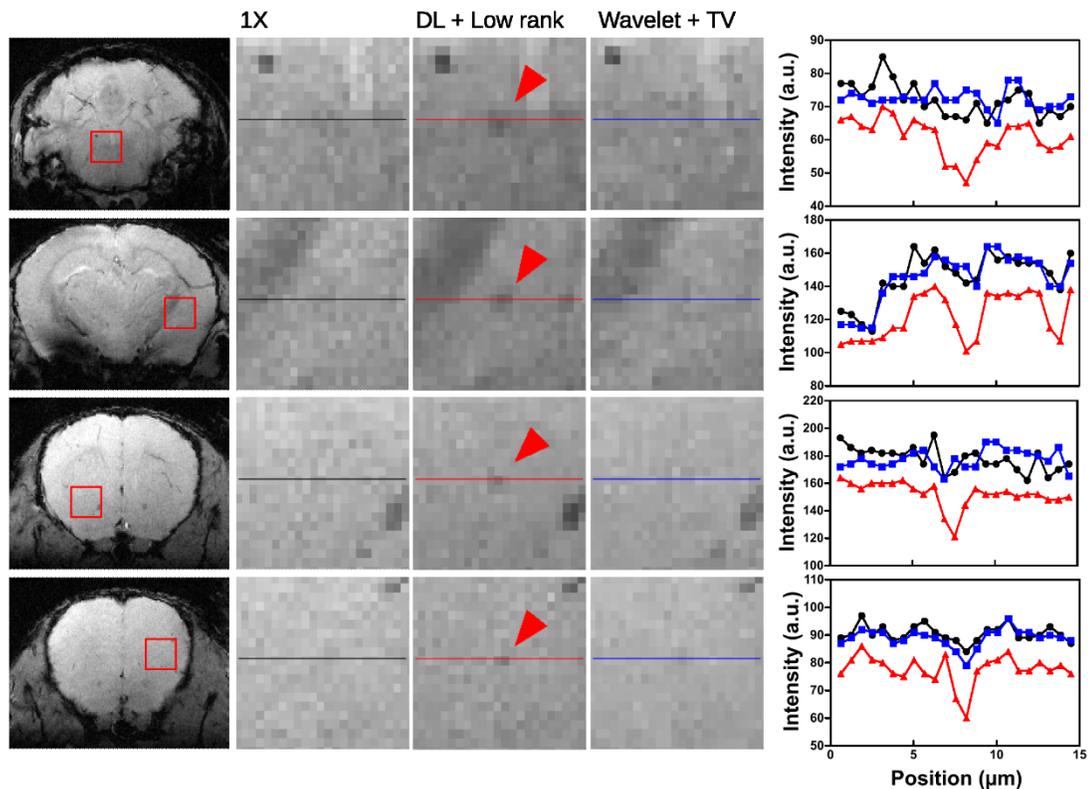


Fig. 3-8. In exemplary slices at different positions in the mouse brain (1st column, same as in Fig. 3-7), image details, indicated by the red rectangle, show representative examples of cells which were only detectable using the DLLR undersampled reconstruction. Cells are not visible in the fully sampled images (2nd column, same as in Fig. 3-7). However, the cells are easily visible in the 4.8X DL + Low Rank reconstruction (3rd column, same as in Fig. 3-7), while the 4.8X Wavelet + TV reconstruction (4th column) did not enhance the cell contrast sufficiently for it to be detectable. Line profiles corresponding to the marked lines in the image details are shown in the 5th column.

3.5 Discussion

A variable temporal resolution Cartesian sampling scheme was developed for the single cell tracking experiments. An improvement in the cell detection has been achieved compared to the conventional data sampling.

In the simulation and phantom experiments, most fast-moving cells appeared as a “T” shape in the fully sampled image due to motion artifacts. This did not significantly reduce the discernibility of cell features in the phantom images. However, severe negative

effects are expected in brain images, where the “T” artifact is likely to overlap with or be mistaken for other brain features. These distortions in the fully sampled image may bring challenges in the potential development of an automated detection algorithm as more cell shapes, depending on the motion, should be considered. In the CS images, artifacts were reduced, and the contrast of most cells was improved.

The phantom experiments showed an additional banding artifact caused by high-frequency translational motion when the motor was on. This was not present in the in vivo images, and hence not considered in the design of the sampling scheme. If motion of the subject is severe, the $k=0$ line can be acquired in each of the highest acceleration frames to estimate and correct the motion.

Effects of the CS reconstruction were substantially different for the four classes of cells. The low-velocity high-contrast cells were visible in both the undersampled and fully sampled images, as demonstrated in the simulation and phantom experiments. These generally had a slight increase in CNR in the undersampled images due to denoising in the regularized reconstruction. The high-velocity high-contrast cells, which were challenging to detect in the fully sampled image due to severe temporal blurring, were more easily identified in the undersampled images. On the contrary, the low-velocity low-contrast cells were likely to be detected in the fully sampled images. Simulated results showed only minor CNR improvements for these cells. This is because little benefit was expected from the improved temporal resolution for the near stationary features. Further, CS tends to remove low-contrast features, as for example in the in vivo experiments, where the relatively complex brain structure increased the challenge of reconstructing cells. In preliminary attempts with a density compensated IFFT for the initial guess, these low-

contrast low-velocity cells were more likely to be removed. The flexible undersampling ratio was important for providing the CS reconstruction with a high-quality initial guess. This mitigated the loss of slow-moving features in the undersampled reconstruction. The CNR improvements helped identify fast-moving high-contrast features, although the high-velocity low-contrast cells remained challenging in both the fully sampled and undersampled images.

Single cell tracking is a very challenging problem that is different from most CS applications targeting large scale structures. The cells appear as small signal voids similar to impulse noise, which is removed by many conventional CS regularizers. It is therefore very important to choose an appropriate constraint. Additional cells were detected in the undersampled images using the DLLR-CS method in both simulation and the experiments. However, the wavelet plus TV regularized method was not effective at detecting additional cells and some cells present in the fully sampled and DLLR were removed as shown in Fig. 3-8. In DL, atoms with cell features can be manually added to the dictionary to ensure that the feature of interest is well represented by the regularizer. In the temporal dimension, a low rank model was chosen, instead of TV. Sparse features are low rank so it is less likely to remove cells visible in only a few time frames, which may be removed by TV since they are similar to impulse noise.

False positives rarely occurred in the CS reconstruction in the simulation and phantom experiments. It is impossible to evaluate all false positives in the in vivo experiments, since the ground truth was not available. We have observed empirically that the DL regularizer could compress features into a smaller size. For example, a section of

vein that should be a thin line might result in a void. These can be easily identified by comparing with the fully sampled reconstruction.

The golden angle radial sampling also provides flexible undersampling ratios [29]. One major advantage of the golden angle radial scheme is robustness against motion. In our study, the mouse brain is approximately stationary leading to minor motion artifacts. The periphery of k-space contains information on the cells, and therefore should not be assigned a lower density as in the radial sampling. The golden angle radial scheme has a greater flexibility than the proposed Cartesian undersampling scheme, as the number of lines in the reconstruction is not restricted. However, with a high undersampling ratio, sufficient flexibility can be provided by the Cartesian scheme. Additionally, the Cartesian method is very easy to implement as it simply requires modification of a gradient table.

The recent development of deep learning-based image reconstruction and feature detection may be applied to the single cell tracking MRI. A large data set is generally required to train a neural network. The challenges of small cell features that are similar to noise remain. No matter how powerful a data processing method is, it can only recover the information that is contained within the acquired data. The k-space sampling scheme to optimally capture the cell features and the effectiveness of deep learning will be investigated in future work.

3.6 Conclusion

A flexible Cartesian sampling scheme has been proposed to simultaneously acquire the fully sampled and accelerated images and applied to the challenging problem of real-time single cell tracking MRI. The DLLR-CS reconstruction effectively improved the cell

CNR and recovered the high velocity cells with sufficient contrast, at virtually no cost. The T-shaped distortion of fast cells has been visibly reduced with the improved temporal resolution. Improved cell detection has been achieved in simulation, phantom and in vivo experiments. Based on the phantom experiments, the detection limit for cells moving in-plane has been increased and cells with velocities of up to 7.0 $\mu\text{m/s}$ could be detected. The method is easy to implement as long as the gradient table is accessible.

CHAPTER 4

Optimized Phase Cycling for Coherence Pathway Selection in PEPI Experiments

4.1 Introduction

The Pi Echo-Planar Imaging (PEPI) sequence [3] employs multiple 180° refocusing RF pulses for an efficient readout and rapid data acquisition. The long echo train is similar to that in Fast Spin Echo (FSE) [4], but the spatial encoding gradients do not satisfy the CPMG condition, i.e., a net phase occurred within each echo cycle. The phase accumulates throughout the echo train, so that minimal phase encoding gradient is required on each echo. The reduction in gradient amplitude and absence of rewind gradient significantly reduced gradient duty cycle, eddy currents and concomitant magnetic fields. High spatial resolution could be achieved beyond the limit permitted by the gradient hardware with conventional sequences, such as FSE [3,9]. This is also advantageous for low field imaging where the concomitant field is more significant [10] and the low SAR permits numerous high flip angle RF pulses [36].

The major limitation of PEPI is the requirement of a sufficiently high-quality 180° refocusing pulse, which should reverse all the phase introduced by spatial encoding gradients. Otherwise, coherence pathway artifacts occur. BLIPPED [9], the pure phase encoding version of PEPI, is subject to the same constraint. Composite pulses have been used to achieve the highest possible quality of refocusing pulse [9,37]. The XY-16 phase cycling scheme, instead of CPMG phase, has been employed to prevent cumulative pulse errors since the CPMG condition is not satisfied without rewind gradients. Even with composite pulses and XY-16 phase cycling, the sample must be confined to the

homogeneous region of the RF coil [9,37]. Artifact-free, slice-selective experiments have not been possible as imperfect slice profiles result in a range of flip angles.

Coherence pathway selection has been a research topic in MR experiments. When the desired pathway is the FID signal, relatively simple solutions, such as spoiling gradients and RF spoiling can be applied [38]. Crusher gradients [39] can be applied to select the echo signal. Phase cycling schemes have also been used in NMR experiments to eliminate all unwanted coherence pathways [40], which requires many repetitions and may be too time consuming for some imaging experiments.

The extended phase graph (EPG) formalism has been used previously for calculating echo amplitudes in FSE experiments [41]. Several works have also used the partitioning method to keep track of the phase and amplitude of each coherence pathway independently [42,43]. Kaffanke et al used the partitioning method to design a phase-cycled averaging to eliminate the residual magnetization for a SPRITE experiment [43]. The phase-cycled average allowed the selection of FID signal pathways for low flip angle excitations.

In this work, the partitioning method will be used to develop a phase-cycling scheme selecting the main spin-echo pathway with flip angles near 180° in PEPI experiments.

4.2 Theory

4.2.1 PEPI Pulse Sequence

The pulse sequence diagram for the 2D PEPI is shown in Fig. 4-1. The net phase within each echo, from either the phase encoding or the frequency encoding gradient, is nonzero. In the phase encoding dimension, gradient is applied on every second echo. The refocusing pulse reverses the phase, causing a flip across the origin in k-space, so that another line of k-space is covered without applying additional gradient. A 3D PEPI was performed with the slice selection gradient replaced with a second phase encoding gradient. The second phase gradient was applied only after the excitation pulse so that k_{phase_2} is constant along the echo train, except for a change of sign between even and odd echoes.

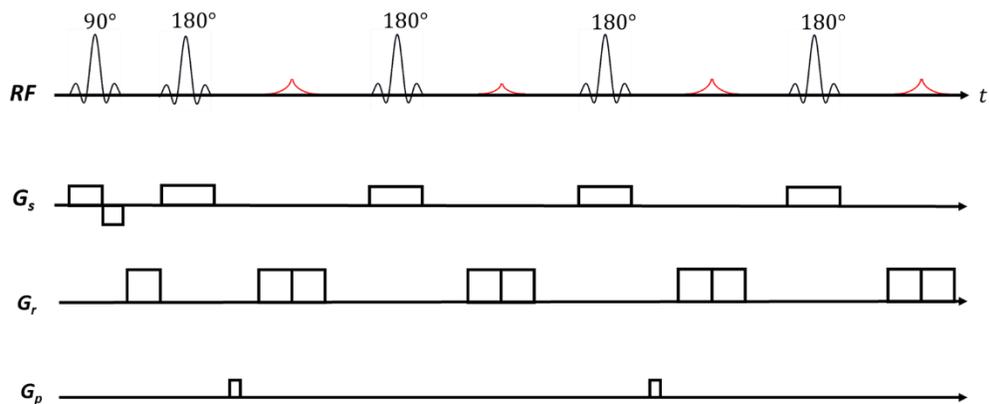


Fig. 4-1. The pulse sequence diagram for 2D PEPI. The phase encoding gradient is applied after every other refocusing pulse. The frequency encoding gradient is also unbalanced.

4.2.2 The Partitioning Method

The coherence pathways which arise during an echo train can be efficiently described using the partitioning method originally described by Kaiser et al [42]. The partitioning method is closely related to the EPG method [44], except that each coherence

pathway is considered separately rather than combining states with the same dephasing. In this work, the RF phase is included in the RF pulse operators. [15]

First, consider a change of basis from the M_x , M_y , M_z components of the net magnetization vector of a single isochromat at position \mathbf{r} :

$$M_+(\mathbf{r}) = \frac{1}{\sqrt{2}}(M_x + iM_y) = \frac{1}{\sqrt{2}}Me^{i\theta} = (M_-)^* \quad (4.1a)$$

$$M_-(\mathbf{r}) = \frac{1}{\sqrt{2}}(M_x - iM_y) = \frac{1}{\sqrt{2}}Me^{-i\theta} = (M_+)^* \quad (4.1b)$$

$$M'_z = M_z \quad (4.1c)$$

where $M = |M_{xy}|$, $\cos\theta = \frac{M_x}{M}$ and $\sin\theta = \frac{M_y}{M}$. The change of basis can be described by the

following unitary operators:

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & +i & 0 \\ 1 & -i & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \quad (4.2a)$$

$$U^{-1} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 \\ -i & +i & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \quad (4.2b)$$

To reach the so called ‘‘configuration state’’ basis in the EPG formalism, we Fourier transform the M_+ , M_- , M'_z basis.

$$F_+(\mathbf{k}) = \int_V M_+(\mathbf{r}) e^{-i\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{r} = \int_V \frac{1}{\sqrt{2}}(M_x(\mathbf{r}) + iM_y(\mathbf{r})) e^{-i\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{r} \quad (4.3a)$$

$$F_-(\mathbf{k}) = \int_V M_-(\mathbf{r}) e^{-i\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{r} = \int_V \frac{1}{\sqrt{2}}(M_x(\mathbf{r}) - iM_y(\mathbf{r})) e^{-i\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{r} \quad (4.3b)$$

$$Z(\mathbf{k}) = \int_V M'_z(\mathbf{r}) e^{-i\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{r} = \int_V M_z(\mathbf{r}) e^{-i\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{r} \quad (4.3c)$$

The three real valued parameters (M_x, M_y, M_z) have been transformed into three complex parameters (F_+, F_-, Z), so there are some redundancies. Since The $Z(\mathbf{k})$ states are the Fourier transform of the real valued $M'_z(\mathbf{r})$ component, we have the relationship $Z(\mathbf{k})^* = Z(-\mathbf{k})$. Similarly, $F_+(\mathbf{k})^* = F_-(-\mathbf{k})$ can be easily derived based on Eqn. 4.3a & 4.3b. As a result, one only needs to store positive \mathbf{k} states with no loss of information.

The effects of gradients and RF pulses can now be described in this configuration state basis. Consider a gradient which creates a phase $\mathbf{k}' \cdot \mathbf{r}$ in the M_x, M_y, M_z basis:

$$\begin{bmatrix} M_x(\mathbf{r}) \\ M_y(\mathbf{r}) \\ M_z(\mathbf{r}) \end{bmatrix} \rightarrow \begin{bmatrix} M(\mathbf{r}) \cos(\mathbf{k}' \cdot \mathbf{r}) \\ -M(\mathbf{r}) \sin(\mathbf{k}' \cdot \mathbf{r}) \\ M_z(\mathbf{r}) \end{bmatrix} \quad (4.4)$$

Using U to convert to the M_+, M_-, M_z basis, the effect of the gradient is

$$U \begin{bmatrix} M_x(\mathbf{r}) \\ M_y(\mathbf{r}) \\ M_z(\mathbf{r}) \end{bmatrix} \rightarrow U \begin{bmatrix} M(\mathbf{r}) \cos(\mathbf{k}' \cdot \mathbf{r}) \\ -M(\mathbf{r}) \sin(\mathbf{k}' \cdot \mathbf{r}) \\ M_z(\mathbf{r}) \end{bmatrix} \quad (4.5)$$

$$\begin{bmatrix} M_+(\mathbf{r}) \\ M_-(\mathbf{r}) \\ M'_z(\mathbf{r}) \end{bmatrix} \rightarrow \begin{bmatrix} M_+(\mathbf{r}) e^{-ik' \cdot \mathbf{r}} \\ M_-(\mathbf{r}) e^{ik' \cdot \mathbf{r}} \\ M'_z(\mathbf{r}) \end{bmatrix}$$

The Fourier transform is then applied to change to the configuration state basis

$$\begin{bmatrix} F_+(\mathbf{k}) \\ F_-(\mathbf{k}) \\ Z(\mathbf{k}) \end{bmatrix} \rightarrow \begin{bmatrix} \int_V M_+ e^{-ik' \cdot \mathbf{r}} e^{-ik \cdot \mathbf{r}} d^3 \mathbf{r} \\ \int_V M_- e^{ik' \cdot \mathbf{r}} e^{-ik \cdot \mathbf{r}} d^3 \mathbf{r} \\ \int_V M'_z(\mathbf{r}) e^{-ik \cdot \mathbf{r}} d^3 \mathbf{r} \end{bmatrix} = \begin{bmatrix} F_+(\mathbf{k} + \mathbf{k}') \\ F_-(\mathbf{k} - \mathbf{k}') \\ Z(\mathbf{k}) \end{bmatrix}. \quad (4.6)$$

A gradient simply changes \mathbf{k} . Since \mathbf{k} is a measure of dephasing, the Z state has phase storage. The F_+ state undergoes dephasing, and the F_- state undergoes rephasing.

The effect of an RF pulse with flip angle α and phase φ in the M_x, M_y, M_z basis can be described by the rotation matrix $R_{xy}(\varphi, \alpha) = R_z(\varphi)R_x(\alpha)R_z(-\varphi)$. This can be transformed into the M_+, M_-, M'_z basis as $T(\varphi, \alpha) = UR_{xy}(\varphi, \alpha)U^{-1} = UR_z(\varphi)R_x(\alpha)R_z(-\varphi)U^{-1}$. It is unchanged when taking the Fourier Transform assuming there is no \mathbf{r} dependence. The effect of an RF pulse in the configuration state basis is

$$\begin{bmatrix} F_+(\mathbf{k}) \\ F_-(\mathbf{k}) \\ Z(\mathbf{k}) \end{bmatrix} \rightarrow UR_z(\varphi)R_x(\alpha)R_z(-\varphi)U^{-1} \begin{bmatrix} F_+(\mathbf{k}) \\ F_-(\mathbf{k}) \\ Z(\mathbf{k}) \end{bmatrix} \quad (4.7)$$

$$= \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \begin{bmatrix} F_+(\mathbf{k}) \\ F_-(\mathbf{k}) \\ Z(\mathbf{k}) \end{bmatrix}$$

$$\begin{bmatrix} F_+(\mathbf{k}) \\ F_-(\mathbf{k}) \\ Z(\mathbf{k}) \end{bmatrix} \rightarrow \begin{bmatrix} \cos^2\left(\frac{\alpha}{2}\right) & e^{2i\varphi}\sin^2\left(\frac{\alpha}{2}\right) & -\frac{i}{\sqrt{2}}e^{i\varphi}\sin(\alpha) \\ e^{-2i\varphi}\sin^2\left(\frac{\alpha}{2}\right) & \cos^2\left(\frac{\alpha}{2}\right) & \frac{i}{\sqrt{2}}e^{-i\varphi}\sin(\alpha) \\ -\frac{i}{\sqrt{2}}e^{-i\varphi}\sin(\alpha) & \frac{i}{\sqrt{2}}e^{i\varphi}\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} F_+(\mathbf{k}) \\ F_-(\mathbf{k}) \\ Z(\mathbf{k}) \end{bmatrix} \quad (4.8)$$

RF pulses in the configuration state basis cause a mixing of states with the same \mathbf{k} . Under the EPG formalism, the system is described by a state matrix with elements representing the three states F_+, F_-, Z at different dephasing order \mathbf{k} . The $F_+(\mathbf{k} = 0)$ state represents the in phase, transverse magnetization corresponding to the sum of all echoes and recently excited signal. The partition method is closely related to the EPG formalism. Instead of applying the matrix multiplication as in Eqn 4.8, the states are split as:

$$F_+(\mathbf{k}) \rightarrow \begin{bmatrix} T_{11}F_+(\mathbf{k}) \\ T_{21}F_-(\mathbf{k}) \\ T_{31}Z(\mathbf{k}) \end{bmatrix}, F_-(\mathbf{k}) \rightarrow \begin{bmatrix} T_{12}F_+(\mathbf{k}) \\ T_{22}F_-(\mathbf{k}) \\ T_{32}Z(\mathbf{k}) \end{bmatrix}, Z(\mathbf{k}) \rightarrow \begin{bmatrix} T_{13}F_+(\mathbf{k}) \\ T_{23}F_-(\mathbf{k}) \\ T_{33}Z(\mathbf{k}) \end{bmatrix} \quad (4.9)$$

This is more memory intensive compared to the EPG method, but it allows coherence pathways to be described independently. An example of the partition method for a three RF pulse sequence is given in Appendix 1.

Based on Eqn 4.7-4.9, the phase of each coherence pathway is a function of the phases of the RF pulses multiplied by some integer. Since the RF phases change during the optimization of the phase cycling scheme, only the amplitude and integer multiples can be precalculated for each pathway. The constant terms in the phase are not essential for the cost function proposed in this work. Only the relative phase amongst averages is relevant when determining if the signals add constructively or destructively.

4.2.3 Cost Function

The goal of this work is to minimize artifacts from unwanted coherence pathways in the PEPI experiments. A phase cycling scheme is developed where different RF phases are employed in multiple repetitions. The complex signal of each unwanted coherence pathway has different phases in the repetitions leading to a diminished sum. The cost function in Eqn. 4.10 penalizes any residual contributions from unwanted coherence pathways in all the echoes after averaging.

$$Cost = \sum_l \sum_j |A_j \sum_k e^{i(\varphi_j \cdot \Phi_k - \theta_{receiver,l}(\Phi_k))}|, \quad (4.10)$$

where l , j , and k are indices of the echo number, coherence pathway, and repetition of scan, respectively. Φ_k is a vector containing the phases of all the RF pulses in the k th repetition. φ_j is a vector storing the phase dependence of the j th coherence pathway on the RF phase and A_j is the amplitude of the corresponding pathway. $\theta_{receiver,l}(\Phi_k)$ is the

receiver phase at the l th echo time as determined by the phase of the desired coherence pathway.

4.3 Methods

4.3.1 Cost Function

A tree structure was used to pre-calculate and store the phase dependence φ_j and amplitude A_j for each coherence pathway. The root node representing the thermal equilibrium magnetization had attributes amplitude = 1, phase accumulation $\mathbf{k} = 0$ and a vector of zeros for the phase dependence φ_j . Three children were created for each node to represent the splitting of a state by the RF pulse into the F_+ , F_- , Z states with the same \mathbf{k} . The amplitude and phase dependence were calculated by Eqn. 4.9. The variable \mathbf{k} was updated to account for the frequency encoding gradient. Each child created three additional children for each RF pulse until the end of the pulse train. A flag was assigned to each node based on its \mathbf{k} value indicating if an echo occurred and another flag indicating if it was a desired or unwanted signal contribution. Lists of all unwanted signal pathway amplitudes A , phase dependence vectors φ and receiver phases $\theta_{receiver}$ were obtained based on the flags.

Programming of the optimization and simulation was performed in Python. The `scipy.optimize.minimize` function with the Nelder-Mead method [45] was used for the optimization with a flip angle of 150° . The optimization was repeated 10 times with different randomized initial guess for the phases ranging from 0 to 2π . The result with the lowest cost function value was used. A laptop with an Intel (R) Core (TM) i7-9750H CPU @ 2.60GHz and 16GB of RAM was used. Calculating the phase coefficients and

amplitudes of coherence pathways for a 16 Echo Train Length (ETL) scan took 0.05s. Optimization of the RF phases for a 16 ETL, four averages scan took 12min 1.2s.

4.3.2 Simulations

A Bloch equation simulation on a 2D phantom with a matrix size of 64x64 and 21 isochromats per voxel was used to test the coherence pathway cancelation method. A homogeneous T_1 and T_2 of 100ms was used across the phantom. The simulation phantom structure was chosen to match a 3D printed resolution phantom. A TE and echo spacing of 2.5 ms and TR of 300 ms were used in the simulation. A 160° refocusing flip angle was used to simulate PEPI with the optimized phase cycling, CPMG phase scheme, CPMG with composite pulses, and XY-16 phase scheme. An ETL of 16 was used with four phase cycling averages. For the simulation and all experiments, the k-space center was acquired first and the modulation transfer function was a lowpass filter.

4.3.3 Phantom Experiments

A 1T permanent magnet, small animal scanner from Aspect (M2, Aspect Imaging Ltd., Israel) was used for experimental evaluation of the method. A 35mm diameter solenoidal RF coil was used for the resolution phantom experiments and a mouse head coil was used for the mouse images.

For the resolution phantom, 3D/2D PEPI experiments had field of view (FOV) of 25mm x 25mm x 25mm/25mm x 25mm with a 0.4mm isotropic/in-plane resolution. The TR, TE and dwell time were 300ms/710ms, 2.5ms/4ms and 15 μ s, respectively. The 2D slice thickness was 1mm. The ETL was 16 with four phase cycling averages. A composite

$90_{90}180_090_{90}$ refocussing pulse was used for the 3D PEPI to compensate for pulse imperfections [3,46].

4.3.4 Mouse Head Experiments

The 2D PEPI and FSE mouse images had a 20mm x 20mm FOV with 1mm slice thickness and in-plane resolution of 0.16mm x 0.16mm. The dwell time was 15 μ s, with a TE of 5ms and TR of 1775ms. An ETL of 32 was used with 112 averages for both experiments. A 4-average phase cycling scheme was repeated 28 times for PEPI. The total scan time was 13 minutes and 15 seconds. The max phase encoding gradient strengths were 2.4G/cm and 25.2G/cm for the 2D PEPI and FSE, respectively. The phase gradient duty cycling was 223 times higher for FSE than PEPI.

4.4 Results

4.4.1 Cost Function

An example result of the optimization is shown in Fig. 4-2 for a 16 ETL experiment with four averages. The four coloured lines correspond to different repetitions of the experiment, while the conventional XY-16 phase cycling [47] is shown in black. The output of the optimization has random-like variation along the echo train direction and across averages. For the 3D PEPI, the FIDs of the near 180° pulses and the echoes of these signal pathways were included in the cost function. For a 160° refocusing pulse, the optimization resulted in a 62% reduction of the cost function value compared to no phase cycling. For the 2D PEPI the cost function did not include the FID or its echoes since the slice select gradient acts as a crusher for these signal pathways. For a 160° refocusing pulse,

the optimization resulted in a 71% reduction of the cost function value compared to without phase cycling.

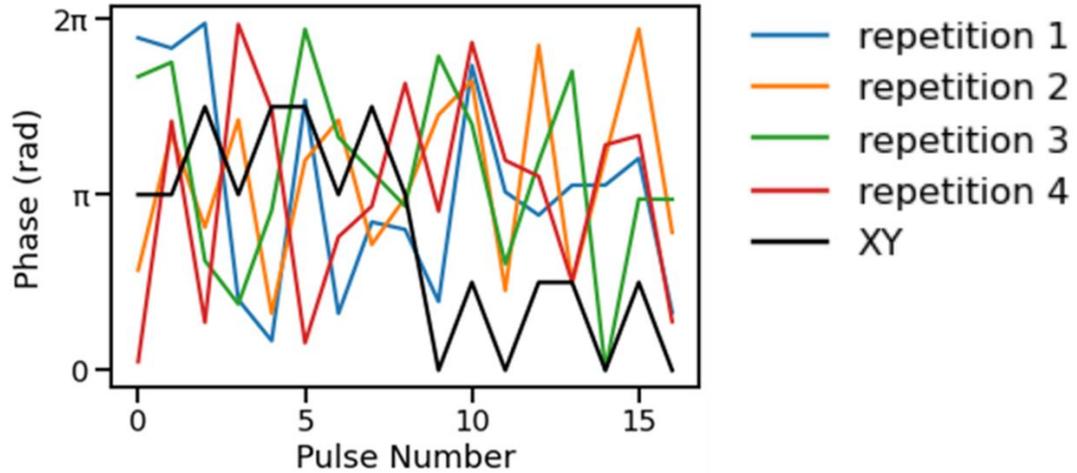


Fig. 4-2. An example output of the optimized phase cycling scheme. Each coloured line shows the RF phases of one average. The proposed method has random like oscillations compared to the XY-16 phase cycling in black.

4.4.2 Simulations

In the simulated images using CPMG phase scheme for PEPI, shown in Fig. 4-3b, high intensity coherence pathway artifacts are present along the phase encoding direction. A CPMG with composite refocusing pulses as in [4] was simulated in Fig. 4-3c. The artifacts were reduced, which were still significant with this flip angle. XY-16 phase scheme is shown in Fig. 4-3d which had several minor banding artifacts. Each of the unaveraged images from the optimized phase scheme had substantial artifacts as shown in Fig. 4-3e. In the averaged image Fig. 4-3f, the artifacts were largely removed, leaving only low intensity variations absent in the ground truth phantom Fig. 4-3a.

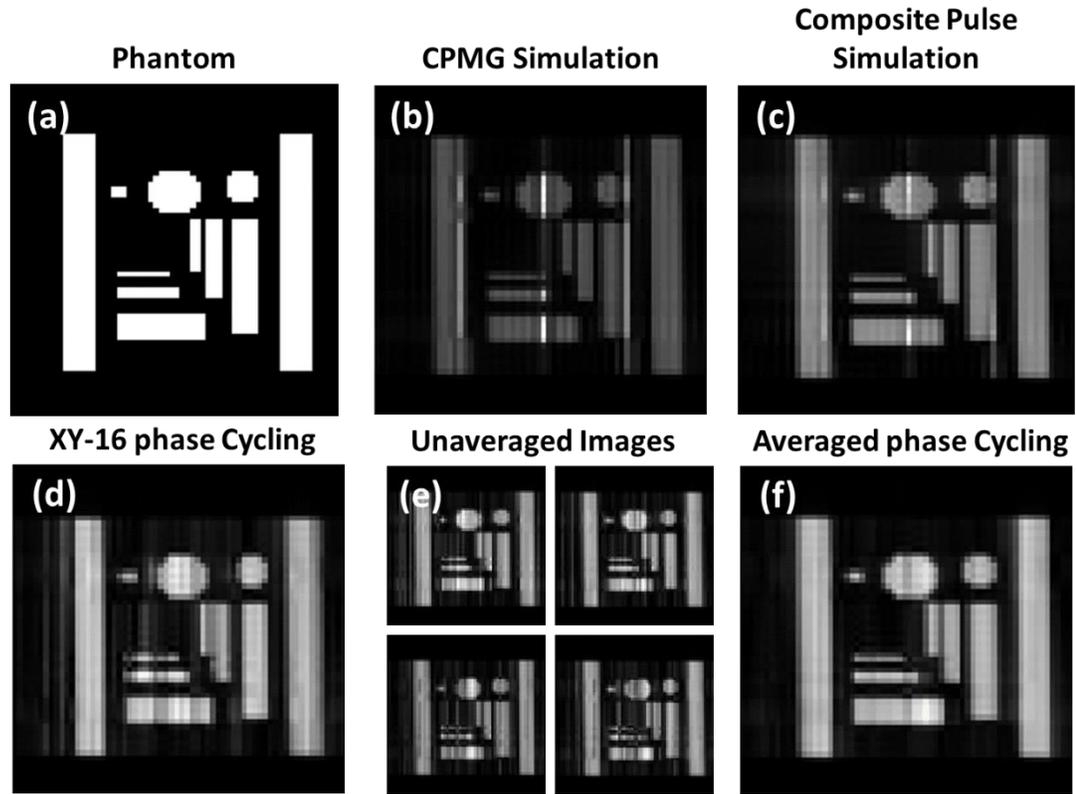


Fig. 4-3. The PEPI experiments on a phantom (a) were simulated with Bloch equation with 160° refocusing pulses and 16 ETL. The CPMG phase scheme (b) resulted in substantial banding artifacts along the phase encoding direction. The CPMG with composite refocusing pulses (c) slightly reduced the artifact. An XY-16 phase scheme (d) led to low intensity artifacts in the phase encoding direction. Each repetition from the optimized phase cycling scheme is shown in (e), where the artifacts varied with different RF phases. The average (f) of the complex images in (e) is comparable to (a) with the artifacts mostly removed.

4.4.3 Phantom Experiments

3D PEPI images were acquired experimentally on a resolution phantom with the proposed method and XY-16 phase cycling, as shown in Fig. 4-4. The phase cycling method produced an artifact free image with nominal 180° refocusing pulses and a very similar quality image at a nominal 170° , as shown in the top row of Fig. 4-4. Minor artifact was observed at 160° in Fig. 4-4c, which is comparable to the simulation results Fig. 4-3f. With XY-16 phase, low intensity artifacts appeared with 180° and 170° , while the

coherence pathway resulted in severe image quality degradation at 160° , as shown in the bottom row of Fig. 4-4. Considering the nominal 180° image with optimized phase cycling Fig. 4-4a as the reference, the XY-16 images had an RMSE of 0.0018, 0.0024, and 0.0052 for nominal flip angles of 180° , 170° and 160° , respectively. The optimized phase cycling images had an RMSE of 0.0004 and 0.0019 for the 170° and 160° , respectively.

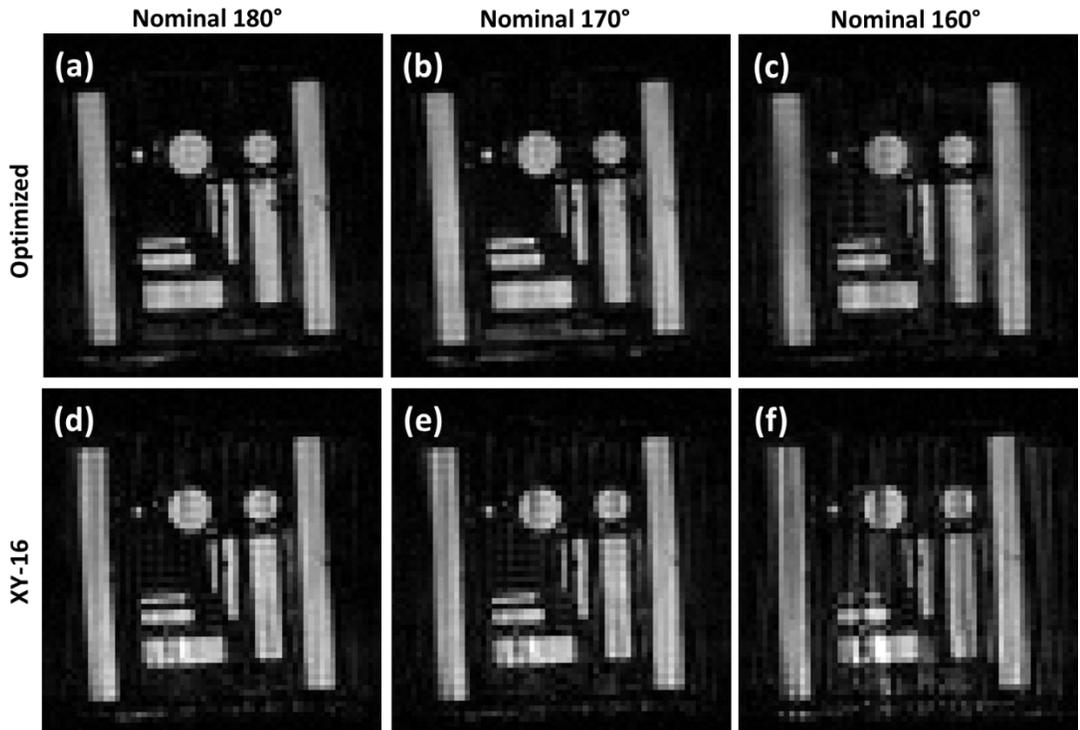


Fig. 4-4. 3D PEPI experiments (16 ETL and 4 averages) with varying refocusing flip angles were performed with the proposed phase cycling scheme (1st row) and XY-16 (2nd row). The first phase encoding direction with blips applied every second echo is horizontal while the second phase encoding direction is vertical. The image quality was good at a nominal 180° for both methods (1st column). With a 170° nominal flip angle, coherence pathway artifacts were apparent in the XY-16 image (e), while the optimized method (b) remained relatively artifact free. A 160° refocusing pulse resulted in significant artifacts in both images (3rd column). The XY-16 image (f) quality was extremely degraded.

2D PEPI experiments were performed on the phantom at a range of nominal flip angles, as shown in Fig. 4-5. The optimized phase cycling scheme produced good quality images at nominal flip angles of 180° and 170° , as shown in Fig. 4-5a and b. The coherence

pathway artifacts became more significant at a nominal flip angle of 160° as shown in Fig. 4-5c, where most features were still resolvable. The XY-16 method was unable to achieve an artifact free image for 2D PEPI even with a nominal flip angle of 180° as shown in Fig. 4-5d. Low intensity coherence pathway artifacts were present along the horizontal phase encoding direction. Fig. 4-5e and Fig. 4-5f show the artifacts increasing in intensity and further degrading image quality with nominal 170° and 160° refocussing pulses. Taking the nominal 180° image with optimized phase cycling as the reference, the XY-16 images had an RMSE of 0.0031, 0.0050, and 0.0101 for nominal flip angles of 180° , 170° and 160° , respectively. The optimized phase cycling images had an RMSE of 0.0023 and 0.0063 for the 170° and 160° , respectively. These results qualitatively agreed with the 3D phantom experiments, but the slice selection led to lower SNRs.

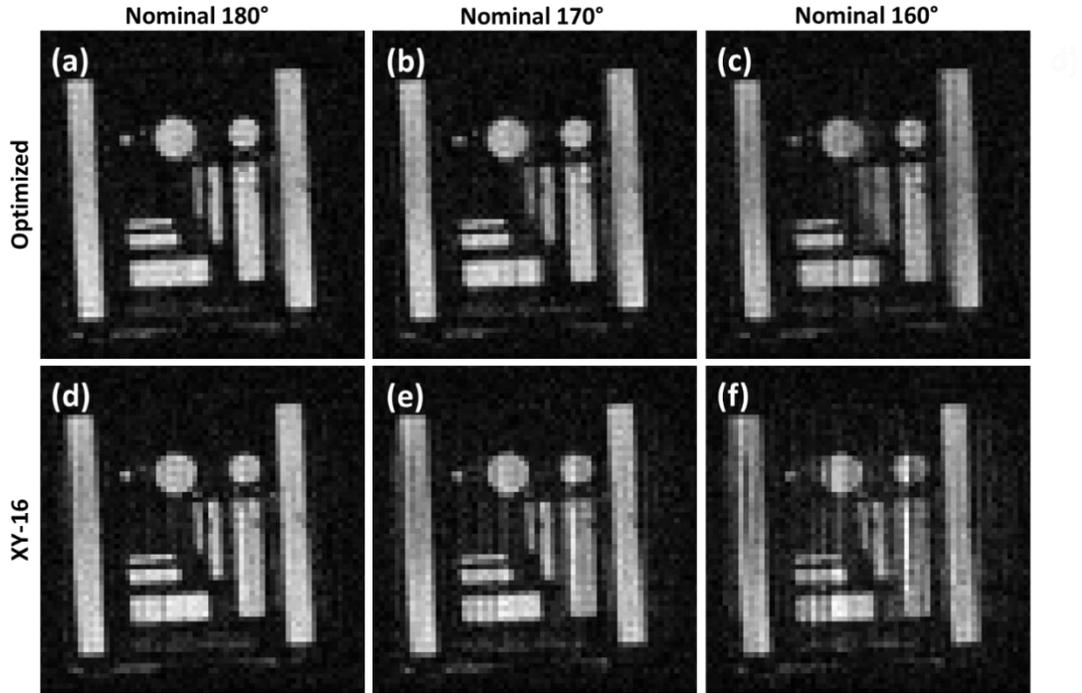


Fig. 4-5. 2D PEPI experiments (16 ETL and 4 averages) with varying refocusing flip angles were performed with the proposed phase cycling scheme (1st row) and XY-16 (2nd row). The phase encoding direction is horizontal while the second phase encoding direction is vertical. The image quality was good at a nominal 180° for the optimized method (a), while low intensity artifacts are observed with XY-16 (d). With a 170° nominal flip angle, coherence pathway artifacts were significant in the XY-16 image (e), while the optimized method (b) shows only minor artifacts. A 160° refocusing pulse resulted in artifacts in both images (3rd column) where the artifacts were of much higher intensity in the XY-16 (f).

4.4.3 Mouse Head Experiments

Images of a mouse head were acquired with 2D PEPI and compared to 2D FSE. A slice at the gradient isocenter (1st row) and a 2mm off center slice (2nd row) are shown in Fig. 4-6. The 2D PEPI using the optimized phase cycling produced high quality artifact-free images (1st column). Both images with XY-16 (2nd column) had substantial coherence pathway artifacts. 2D FSE images with the same timing parameters as the PEPI images are shown in Fig. 4-6c and Fig. 4-6f. The SNR and overall image quality is lower in the FSE compared to the PEPI with phase cycling. In FSE (3rd column), more blurring was observed in the slice away from isocenter, Fig. 4-6f, compared to the slice at isocenter Fig.

4-6c, due to the concomitant magnetic fields. The blurring appears to be more significant further from the center. For example, there is more blurring in the region highlighted in Fig. 4-6f compared to that highlighted in Fig. 4-6c. This increase in blurring away from the gradient isocenter confirmed the concomitant field effects. On the contrary, the two PEPI slices had comparable quality as the phase encoding gradient amplitudes were largely reduced.

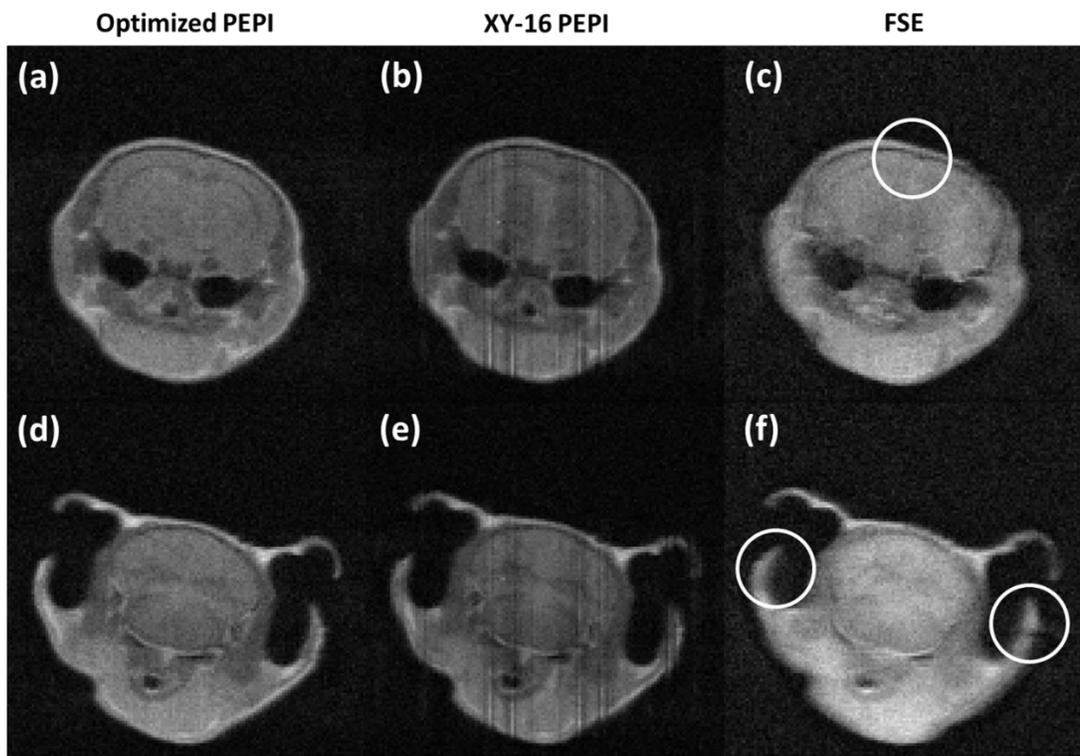


Fig. 4-6. 2D PEPI image of a mouse brain using the optimized phase cycling scheme (1st column) and XY-16 (2nd column). The 1st row is the slice at the gradient isocenter and the 2nd row is offset 2mm from the isocenter. Significant coherence pathway artifacts are present in XY-16. 2D FSE images (3rd column) had reduced quality. Additional blurring away from the gradient isocenter was observed, as highlighted in (f), due to the concomitant magnetic field, while features close to the gradient isocenter were less degraded, such as highlighted in (c).

4.4 Discussion

A PEPI experiment with coherence pathway cancellation has been designed and verified. The method requires multiple averages with different RF phases, which has been demonstrated to be less sensitive to imperfect refocusing than conventional methods. Additionally, the proposed method permits artifact-free, slice-selective PEPI which was not previously possible due to imperfect slice profile. This greatly expands the potential applications of PEPI since the sample can extend outside the homogeneous region of the B_1 field along the slice direction as in the mouse head images.

For low-field applications it is often necessary to perform averages to achieve sufficient SNR, especially for high resolution imaging, so there is no extra cost to incorporate the phase cycling scheme. Additionally, flip angles of the refocusing pulse do not need to be reduced to maintain SAR limit, so that the deviations from 180° are only due to B_1 field inhomogeneity, error in RF calibration, and imperfect slice selection profile. Compared to FSE, PEPI reduces the phase gradient duty cycle by a large factor. In this work, a reduction of 223X was achieved. The signal was reduced by a small percentage in the coherence pathways cancelation as a compromise. This trade-off diminished the concomitant field effects.

Concomitant field artifacts can be significant for high resolution imaging such as the mouse head images in Fig. 4-6. Additional blurring appeared in the FSE image away from the gradient isocenter, consistent with concomitant field effects. This was not present in the PEPI images because the gradient amplitude was much lower for the identical spatial resolution.

The optimized phase cycling method can be applicable to other experiments, such as BLIPPED [9] and TRASE [48]. BLIPPED is a pure phase encoding version of PEPI thus it suffers from similar coherence pathway artifacts with imperfect refocusing [9]. TRASE encodes spatial information with the RF phase variation, so different coherence pathways also have different spatial encoding similar to PEPI.

It has been assumed that the RF pulse $T(\varphi, \alpha)$ was not dependent on position when the M_+, M_-, M'_z basis was transformed to the configuration state basis. In practice, there is usually some variation in B_1 field across the sample, where the proportion of mixing between the F_+, F_-, Z states and the phase introduced are the same, with an additional convolution causing a mixing between states of different \mathbf{k} . The B_1 variation is generally small in the spatial domain, so the convolution kernel is narrow leading to a mixing over a very small range of \mathbf{k} . This should not create significant errors for volume coils, especially at low field.

In this work, the phase of each coherence pathway is recorded separately. The cost function is the sum squared of the residual of each coherence pathway after averaging. A flip angle must be chosen to calculate the amplitudes of the coherence pathways for weighting the terms. The optimization attempts to set the phase such that each coherence pathway cancels over the averages. Since the amplitudes of unwanted pathways have factors which tend towards zero as the flip angle approaches 180° , it is expected that a phase cycling scheme which produces an acceptable quality at a low flip angle will remain acceptable or perform even better for flip angles closer to 180° .

Coefficients for the phase dependence of each RF pulse as well as the amplitude of each pathway are precalculated and stored to speed up the optimization. Since the number of coherence pathways grows exponentially with the number of pulses, it is necessary to discard low amplitude coherence pathways when the number of RF pulses is large.

A modification to the cost function of potential interest is to sum over all unwanted coherence pathways at a given acquisition time before taking the magnitude as:

$$Cost = \sum_l \left| \sum_j A_j \sum_k e^{i(\varphi_j \cdot \Phi_k - \theta_{receiver,l}(\Phi_k))} \right|, \quad (7)$$

where different coherence pathways from the same average may cancel each other which could result in a better cancelation of unwanted signal. It is, however, expected to increase sensitivity to the flip angle as the amplitudes of these different pathways depend on the flip angle.

From Eqn. 4.8, the amplitude of the desired coherence pathway decays as a function of $\sin^{2n} \left(\frac{\alpha}{2} \right)$ where n is the number of refocusing pulses. This extra decay in addition to the T_2 decay is not present in FSE with CPMG phase cycling. This contributes to a more significant point spread function (PSF). As a result, low flip angle refocusing used in FSE to reduce SAR are not practical for PEPI experiments even with good artifact elimination. PEPI is more advantageous in low field MRI where SAR is not a limiting factor for nominal 180° pulses and only imperfect slice selection and B_1 inhomogeneity need to be accounted for. It is therefore unnecessary to perform the optimization at very low flip angles. It should also be noted that the reduced spatial encoding gradients in PEPI can permit a shorter echo spacing, reducing the T_2 decay contribution to the PSF. PEPI has a less severe PSF when studying short T_2 systems with the flip angle close to 180° .

4.5 Conclusion

An optimized phase cycling scheme has been proposed to minimize the signal contribution from unwanted coherence pathways in PEPI experiments. The reduced sensitivity to imperfect refocusing flip angle has been demonstrated with simulation, phantom and mouse head experiments. This method has also enabled high quality 2D imaging with slice selection, so that PEPI can be applied to study a broader range of systems, especially high-resolution imaging at low field.

CHAPTER 5

Conclusions and Future Work

5.1 Conclusions

Two optimization problems in MRI have been examined. The use of CS for the tracking of individual, ION tagged cells to improve temporal resolution was explored. An optimized phase cycling scheme was investigated for artifact reduction in PEPI experiments.

One of the major challenges for MRI is the long acquisition time which can result in motion artifacts including blurring. CS aims to address this, but conventional regularizers are not well suited to cell tracking experiments where the features of interest are noise like signal voids. In this work it was shown that DL and low rank regularizers can be used in CS to improve the temporal resolution in cell tracking experiments provided a high-quality initial guess. A variable undersampling ratio Cartesian sampling scheme was introduced so that a fully sampled reconstruction was acquired simultaneously and can be used as the initial guess. The phantom experiments demonstrated contrast improvements for fast moving cells while the static cells are preserved. The in vivo measurements confirmed that additional cells could be detected which were not visible in the fully sampled images.

PEPI has great potential for low field MRI as it provides images at a similar speed as the widely used FSE, while requiring significantly reduced gradient strength and duty cycle. The serious limitation of a near perfect refocusing pulse is addressed in this thesis. An optimization of the RF pulse phase has been performed to minimize the signal

contribution of the unwanted coherence pathways. Simulations and experimental results have confirmed that the proposed phase cycling scheme is less sensitive to flip angle deviations compared to conventional methods. Additionally, a slice selective PEPI experiment was achieved.

5.2 Future Work

Current cell tracking experiment utilizes T_2^* contrast to image ION labelled cells. The distorted magnetic field causes additional dephasing around the cells, resulting in a reduced intensity. It is expected that the phase of the image also carries information about the presence of tagged cells. In the proposed method this phase information is simply discarded by taking the absolute value. Future work could incorporate this phase information into the contrast of the final image to assist identifying cells, as in susceptibility weighted imaging.

The optimized phase cycling scheme can be useful for a variety of other MRI pulse sequences in addition to PEPI, such as TRASE. The use of this optimization function should be investigated for other pulse sequences susceptible to coherence pathway artifacts. Another possible direction is to modify the cost function to generate new pulse sequences where multiple coherence pathways with different spatial encoding are measured in each acquisition interval for accelerated image acquisition.

REFERENCES

- [1] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging,” *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, Dec. 2007, doi: 10.1002/mrm.21391.
- [2] C. Faber, “MRI of Single Cells Labeled with Superparamagnetic Iron Oxide Nanoparticles,” in *Magnetic Resonance Microscopy*, Wiley, 2022, pp. 263–281. doi: 10.1002/9783527827244.ch12.
- [3] D. Xiao and B. J. Balcom, “ π Echo-Planar Imaging with Concomitant Field Compensation for Porous Media MRI,” *Journal of Magnetic Resonance*, vol. 260, pp. 38–45, Nov. 2015, doi: 10.1016/j.jmr.2015.08.023.
- [4] J. Hennig, A. Nauerth, and H. Friedburg, “RARE imaging: A Fast Imaging Method for Clinical MR,” *Magnetic Resonance in Medicine*, vol. 3, no. 6, pp. 823–833, Dec. 1986, doi: 10.1002/mrm.1910030602.
- [5] Z. P. Liang and P. C. Lauterbur, *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*, 1st ed. New York, N.Y.: Wiley-IEEE Press, 2000.
- [6] M. Shinnar and J. S. Leigh, “The Application of Spinors to Pulse Synthesis and Analysis,” *Magnetic Resonance in Medicine*, vol. 12, no. 1, pp. 93–98, Oct. 1989, doi: 10.1002/mrm.1910120112.
- [7] A. D. Elster, “Gradient-Echo MR Imaging: Techniques and Acronyms,” *Radiology*, vol. 186, no. 1, pp. 1–8, Jan. 1993, doi: 10.1148/radiology.186.1.8416546.
- [8] E. L. Hahn, “Spin Echoes,” *Physical Review*, vol. 80, no. 4, pp. 580–594, Nov. 1950, doi: 10.1103/PhysRev.80.580.
- [9] D. Xiao and B. J. Balcom, “BLIPPED (BLipped Pure Phase EncoDing) High Resolution MRI with Low Amplitude Gradients,” *Journal of Magnetic Resonance*, vol. 285, pp. 61–67, Dec. 2017, doi: 10.1016/j.jmr.2017.10.013.
- [10] P. L. Volegov, J. C. Mosher, M. A. Espy, and R. H. Kraus, “On Concomitant Gradients in Low-Field MRI,” *Journal of Magnetic Resonance*, vol. 175, no. 1, pp. 103–113, Jul. 2005, doi: 10.1016/j.jmr.2005.03.015.
- [11] M. F. Kircher, S. S. Gambhir, and J. Grimm, “Noninvasive Cell-Tracking Methods,” *Nature Reviews Clinical Oncology*, vol. 8, no. 11, pp. 677–688, Nov. 2011, doi: 10.1038/nrclinonc.2011.141.

- [12] E. T. Ahrens and J. W. M. Bulte, “Tracking Immune Cells in Vivo Using Magnetic Resonance Imaging,” *Nature Reviews Immunology*, vol. 13, no. 10, pp. 755–763, Oct. 2013. doi: 10.1038/nri3531.
- [13] M. Masthoff et al., “Introducing Specificity to Iron Oxide Nanoparticle Imaging by Combining ^{57}Fe -Based MRI and Mass Spectrometry,” *Nano Letters*, vol. 19, no. 11, pp. 7908–7917, Nov. 2019, doi: 10.1021/acs.nanolett.9b03016.
- [14] K. Andreas et al., “Highly Efficient Magnetic Stem Cell Labeling with Citrate-Coated Superparamagnetic Iron Oxide Nanoparticles for MRI Tracking,” *Biomaterials*, vol. 33, no. 18, pp. 4515–4525, Jun. 2012, doi: 10.1016/j.biomaterials.2012.02.064.
- [15] V. Hoerr et al., “Bacteria Tracking by in Vivo Magnetic Resonance Imaging,” *BMC Biology*, vol. 11, no. 1, p. 63, Dec. 2013, doi: 10.1186/1741-7007-11-63.
- [16] S. J. Dodd, M. Williams, J. P. Suhan, D. S. Williams, A. P. Koretsky, and C. Ho, “Detection of Single Mammalian Cells by High-Resolution Magnetic Resonance Imaging,” *Biophysical Journal*, vol. 76, no. 1, pp. 103–109, Jan. 1999, doi: 10.1016/S0006-3495(99)77182-1.
- [17] C. Heyn et al., “In Vivo Magnetic Resonance Imaging of Single Cells in Mouse Brain with Optical Validation,” *Magnetic Resonance in Medicine*, vol. 55, no. 1, pp. 23–29, Jan. 2006, doi: 10.1002/mrm.20747.
- [18] E. M. Shapiro, K. Sharer, S. Skrtic, and A. P. Koretsky, “In Vivo Detection of Single Cells by MRI,” *Magnetic Resonance in Medicine*, vol. 55, no. 2, pp. 242–249, Feb. 2006, doi: 10.1002/mrm.20718.
- [19] Y. Mori et al., “From Cartoon to Real Time MRI: In Vivo Monitoring of Phagocyte Migration in Mouse Brain,” *Scientific Reports*, vol. 4, no. 1, p. 6997, May 2015, doi: 10.1038/srep06997.
- [20] M. Masthoff et al., “Temporal Window for Detection of Inflammatory Disease using Dynamic Cell Tracking with Time-Lapse MRI,” *Scientific Reports*, vol. 8, no. 1, Dec. 2018, doi: 10.1038/s41598-018-27879-z.
- [21] M. Masthoff et al., “Resolving Immune Cells with Patrolling Behaviour by Magnetic Resonance Time-Lapse Single Cell Tracking,” *EBioMedicine*, vol. 73, Nov. 2021, doi: 10.1016/j.ebiom.2021.103670.
- [22] V. Estellers, S. Soatto, and X. Bresson, “Adaptive Regularization with the Structure Tensor,” *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1777–1790, Jun. 2015, doi: 10.1109/TIP.2015.2409562.

- [23] J. Caballero, A. N. Price, D. Rueckert, and J. v. Hajnal, “Dictionary Learning and Time Sparsity for Dynamic MR Data Reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 33, no. 4, pp. 979–994, Apr. 2014, doi: 10.1109/TMI.2014.2301271.
- [24] S. Ravishankar and Y. Bresler, “MR Image Reconstruction from Highly Undersampled k-Space Data by Dictionary Learning,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 5, pp. 1028–1041, May 2011, doi: 10.1109/TMI.2010.2090538.
- [25] M. Doneva, P. Börnert, H. Eggers, C. Stehning, J. S negas, and A. Mertins, “Compressed Sensing Reconstruction for Magnetic Resonance Parameter Mapping,” *Magnetic Resonance in Medicine*, vol. 64, no. 4, pp. 1114–1120, Oct. 2010, doi: 10.1002/mrm.22483.
- [26] B. Dumitrescu and P. Irofti, *Dictionary Learning Algorithms and Applications*. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-78674-2.
- [27] R. Rubinstein, A. M. Bruckstein, and M. Elad, “Dictionaries for Sparse Representation Modeling,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010, doi: 10.1109/JPROC.2010.2040551.
- [28] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, “Compressive Sensing via Nonlocal Low-Rank Regularization,” *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3618–3632, Aug. 2014, doi: 10.1109/TIP.2014.2329449.
- [29] L. Feng et al., “Golden-Angle Radial Sparse Parallel MRI: Combination of Compressed Sensing, Parallel Imaging, and Golden-Angle Radial Sampling for Fast and Flexible Dynamic Volumetric MRI,” *Magnetic Resonance in Medicine*, vol. 72, no. 3, pp. 707–717, Sep. 2014, doi: 10.1002/mrm.24980.
- [30] M. Mani, M. Jacob, A. Guidon, V. Magnotta, and J. Zhong, “Acceleration of High Angular and Spatial Resolution Diffusion Imaging Using Compressed Sensing with Multichannel Spiral Data,” *Magnetic Resonance in Medicine*, vol. 73, no. 1, pp. 126–138, Jan. 2015, doi: 10.1002/mrm.25119.
- [31] Haifeng Wang, Xiaoyan Wang, Yihang Zhou, Yuchou Chang, and Yong Wang, “Smoothed Random-Like Trajectory for Compressed Sensing MRI,” in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2012, pp. 404–407. doi: 10.1109/EMBC.2012.6345953.
- [32] S. Winkelmann, T. Schaeffter, T. Koehler, H. Eggers, and O. Doessel, “An Optimal Radial Profile Order Based on the Golden Ratio for Time-Resolved

- MRI,” *IEEE Transactions on Medical Imaging*, vol. 26, no. 1, pp. 68–76, Jan. 2007, doi: 10.1109/TMI.2006.885337.
- [33] M. Armstrong, F. Felix, W. Enrica, M. Max, F. Cornelius, and X. Dan, “Variable Temporal Resolution Cartesian Sampling for Cell Tracking MRI Variable Temporal Resolution Cartesian Sampling for Cell Tracking MR [conference abstract], *ISMRM*, 1815, (2021). [Online]. Available: <https://www.ismrm.org/21/program-files/D-127.htm>.”
- [34] P. Virtanen et al., “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020, doi: 10.1038/s41592-019-0686-2.
- [35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, and O. Grisel, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [36] J. P. Marques, F. F. J. Simonis, and A. G. Webb, “Low-Field MRI: An MR Physics Perspective,” *Journal of Magnetic Resonance Imaging*, vol. 49, no. 6, pp. 1528–1542, Jun. 2019, doi: 10.1002/jmri.26637.
- [37] B. Manz, P. S. Chow, and L. F. Gladden, “Echo-Planar Imaging of Porous Media with Spatial Resolution Below 100 μm ,” *Journal of Magnetic Resonance*, vol. 136, no. 2, pp. 226–230, Feb. 1999, doi: 10.1006/jmre.1998.1619.
- [38] V. Roeloffs, D. Voit, and J. Frahm, “Spoiling Without Additional Gradients: Radial FLASH MRI with Randomized Radiofrequency Phases,” *Magnetic Resonance in Medicine*, vol. 75, no. 5, pp. 2094–2099, May 2016, doi: 10.1002/mrm.25809.
- [39] K. Landheer and C. Juchem, “Dephasing Optimization Through Coherence Order Pathway Selection (DOTCOPS) for Improved Crusher Schemes in MR Spectroscopy,” *Magnetic Resonance in Medicine*, vol. 81, no. 4, pp. 2209–2222, Apr. 2019, doi: 10.1002/mrm.27587.
- [40] M. H. Levitt, P. K. Madhu, and C. E. Hughes, “Cogwheel Phase Cycling,” *Journal of Magnetic Resonance*, vol. 155, no. 2, pp. 300–306, Apr. 2002, doi: 10.1006/jmre.2002.2520.
- [41] J. Hennig, M. Weigel, and K. Scheffler, “Calculation of Flip Angles for Echo Trains with Predefined Amplitudes with the Extended Phase Graph (EPG)-Algorithm: Principles and Applications to Hyperecho and TRAPS Sequences,” *Magnetic Resonance in Medicine*, vol. 51, no. 1, pp. 68–80, Jan. 2004, doi: 10.1002/mrm.10658.

- [42] R. Kaiser, E. Bartholdi, and R. R. Ernst, "Diffusion and Field-Gradient Effects in NMR Fourier Spectroscopy," *The Journal of Chemical Physics*, vol. 60, no. 8, pp. 2966–2979, Apr. 1974, doi: 10.1063/1.1681477.
- [43] J. B. Kuffanek, T. Stöcker, S. Romanzetti, T. Dierkes, M. O. Leach, and N. J. Shah, "Phase-Cycled Averaging for the Suppression of Residual Magnetization in SPI Sequences," *Journal of Magnetic Resonance*, vol. 199, no. 2, pp. 117–125, Aug. 2009, doi: 10.1016/j.jmr.2008.11.018.
- [44] M. Weigel, "Extended Phase Graphs: Dephasing, RF Pulses, and Echoes - Pure and Simple," *Journal of Magnetic Resonance Imaging*, vol. 41, no. 2, pp. 266–295, Feb. 2015, doi: 10.1002/jmri.24619.
- [45] F. Gao and L. Han, "Implementing the Nelder-Mead Simplex Algorithm with Adaptive Parameters," *Computational Optimization and Applications*, vol. 51, no. 1, pp. 259–277, Jan. 2012, doi: 10.1007/s10589-010-9329-3.
- [46] M. H. Levitt, "Composite Pulses," *Magnetic Resonance Spectroscopy*, vol. 18, no. 2, pp. 61–122, 1986.
- [47] T. Gullion, D. B. Baker, and M. S. Conradi, "New, Compensated Carr-Purcell Sequences," *Journal of Magnetic Resonance (1969)*, vol. 89, no. 3, pp. 479–484, Oct. 1990, doi: 10.1016/0022-2364(90)90331-3.
- [48] J. C. Sharp and S. B. King, "MRI using Radiofrequency Magnetic Field Phase Gradients," *Magnetic Resonance in Medicine*, vol. 63, no. 1, pp. 151–161, Jan. 2010, doi: 10.1002/mrm.22188.

APPENDICES

Appendix 1

A1.1 Partition Method for 3 Pulses

The coherence pathways generated from a 3 RF pulse 2D PEPI experiment will be considered. \mathbf{k} is a vector which describes independent dephasing effects with its components. For example, if B_0 inhomogeneity, frequency encoding gradient, phase gradient, and slice gradients are the independent phase effects considered, \mathbf{k} is a vector in a 4-dimensional space ($\mathbf{k} = [k_{\Delta B_0}, k_f, k_p, k_s]$). The coherence pathways will be calculated at locations (1)-(10) shown in Fig. A1-1. For simplicity, the integrated area of each gradient pulse is taken to be 1 and the B_0 inhomogeneity during $TE/2$ is taken to increment $k_{\Delta B_0}$ by 1 (more rigorously one could write $1 \cdot \Delta k_{\Delta B_0}$ for example). The duration of the RF pulse and acquisition window are assumed to be negligible for simplicity.

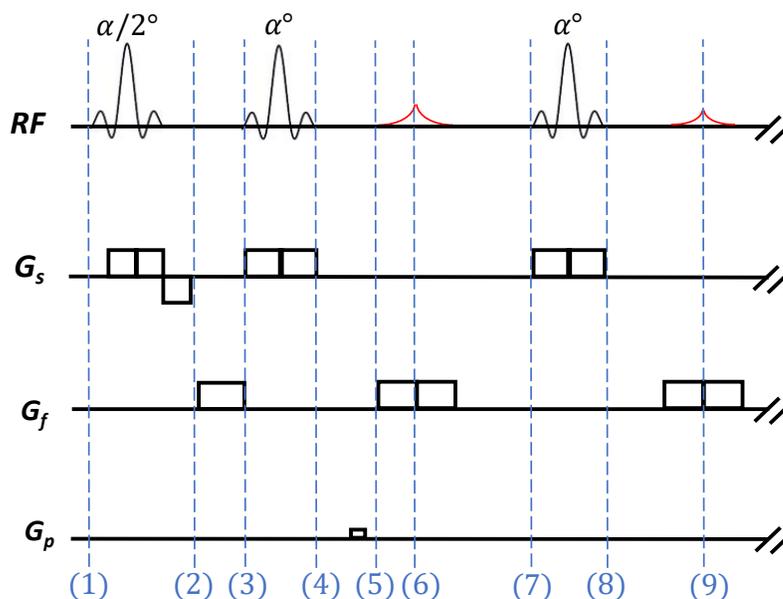


Fig. A1-1. A pulse sequence diagram for the first three pulses in a PEPI experiment with non-ideal refocusing pulses. Numbers (1-9) indicate time points of interest at which the coherence pathways are calculated with the partition method.

At (1) the magnetization is at thermal equilibrium with no phase accumulation. This corresponds to the state $Z(\mathbf{k} = [0,0,0,0])$ with amplitude 1 and no phase. The first RF pulse splits this state based on Eqn. 4.9. Note for $\mathbf{k} = [0,0,0,0]$ only one of the F_+ and F_- states needs to be stored as they are redundant by the relationship $(F_+(\mathbf{k}))^* = F_-(-\mathbf{k})$.

(1)-(2) is described by:

$$Z(\mathbf{k} = [0,0,0,0]) \begin{array}{l} \longrightarrow \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \\ \searrow -\frac{i}{\sqrt{2}} e^{i\varphi_1} \sin\left(\frac{\alpha}{2}\right) F_+(\mathbf{k} = [0,0,0,0]) \end{array} \quad (\text{A2a})$$

$$-\frac{i}{\sqrt{2}} e^{i\varphi_1} \sin\left(\frac{\alpha}{2}\right) F_+(\mathbf{k} = [0,0,0,0]) \quad (\text{A2b})$$

From (2)-(3) there is gradient applied along the frequency encoding direction, and phase accumulation occurs due to B_0 inhomogeneity.

$$\cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \longrightarrow \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \quad (\text{A3a})$$

$$-\frac{i}{\sqrt{2}} e^{i\varphi_1} \sin\left(\frac{\alpha}{2}\right) F_+(\mathbf{k} = [0,0,0,0]) \longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_1} \sin\left(\frac{\alpha}{2}\right) F_+(\mathbf{k} = [1,1,0,0]) \quad (\text{A3b})$$

The refocusing pulse splits these two states. The $(F_+(\mathbf{k}))^* = F_-(-\mathbf{k})$ redundancy is once again used for the splitting of the $Z(\mathbf{k} = [0,0,0,0])$ state. The evolution of the system from (3)-(4) is described by:

$$\cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \begin{array}{l} \longrightarrow \cos(\alpha) \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \\ \searrow -\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos\left(\frac{\alpha}{2}\right) F_+(\mathbf{k} = [0,0,0,1]) \end{array} \quad (\text{A4a})$$

$$-\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos\left(\frac{\alpha}{2}\right) F_+(\mathbf{k} = [0,0,0,1]) \quad (\text{A4b})$$

$$\begin{aligned}
& -\frac{i}{\sqrt{2}} e^{i\varphi_1} \sin\left(\frac{\alpha}{2}\right) \longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_1} \cos^2\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right) & \text{(A4c)} \\
& F_+(\mathbf{k} = [1,1,0,2]) \\
& -\frac{i}{\sqrt{2}} e^{i\varphi_1} \sin\left(\frac{\alpha}{2}\right) \longrightarrow -\frac{i}{\sqrt{2}} e^{i(\varphi_1-2\varphi_2)} \sin^3\left(\frac{\alpha}{2}\right) & \text{(A4d)} \\
& F_+(\mathbf{k} = [1,1,0,0]) \longrightarrow F_-(\mathbf{k} = [1,1,0,0]) \\
& -\frac{i}{\sqrt{2}} e^{i\varphi_1} \sin\left(\frac{\alpha}{2}\right) \longrightarrow -\frac{1}{2} e^{i(\varphi_1-\varphi_2)} \sin\left(\frac{\alpha}{2}\right) \sin(\alpha) & \text{(A4e)} \\
& F_+(\mathbf{k} = [1,1,0,0]) \longrightarrow Z(\mathbf{k} = [1,1,0,1])
\end{aligned}$$

The phase encoding gradient is applied from (4)-(5). Additionally, dephasing from B_0 inhomogeneity occurs. The system at the start of the acquisition window at (5) is given by:

$$\cos(\alpha) \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \longrightarrow \cos(\alpha) \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \quad \text{(A5a)}$$

$$-\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos\left(\frac{\alpha}{2}\right) \longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos\left(\frac{\alpha}{2}\right) \quad \text{(A5b)}$$

$$F_+(\mathbf{k} = [0,0,0,1]) \longrightarrow F_+(\mathbf{k} = [1,0,1,1])$$

$$-\frac{i}{\sqrt{2}} e^{i\varphi_1} \cos^2\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right) \longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_1} \cos^2\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right) \quad \text{(A5c)}$$

$$F_+(\mathbf{k} = [1,1,0,2]) \longrightarrow F_+(\mathbf{k} = [2,1,1,2])$$

$$-\frac{i}{\sqrt{2}} e^{i(\varphi_1-2\varphi_2)} \sin^3\left(\frac{\alpha}{2}\right) \longrightarrow -\frac{i}{\sqrt{2}} e^{i(\varphi_1-2\varphi_2)} \sin^3\left(\frac{\alpha}{2}\right) \quad \text{(A5d)}$$

$$F_-(\mathbf{k} = [1,1,0,0]) \longrightarrow F_-(\mathbf{k} = [0,1,-1,0])$$

$$-\frac{1}{2} e^{i(\varphi_1-\varphi_2)} \sin\left(\frac{\alpha}{2}\right) \sin(\alpha) \longrightarrow -\frac{1}{2} e^{i(\varphi_1-\varphi_2)} \sin\left(\frac{\alpha}{2}\right) \sin(\alpha) \quad \text{(A5e)}$$

$$Z(\mathbf{k} = [1,1,0,1]) \longrightarrow Z(\mathbf{k} = [1,1,0,1])$$

The main signal contributions come from the FID of (A5b) and the spin echo pathway (A5d). The FID pathway has some dephasing due to the slice select gradient, so it is less significant in the 2D PEPI compared to the 3D. The Z pathways, (A5a) and (A5e), do not contribute to signal as they represent the longitudinal component of the magnetization. The remaining pathway (A5c) is a negligible signal contribution due to the high dephasing from the slice select gradient, B_0 inhomogeneity and frequency encoding gradient.

The frequency encoding gradient is applied from (5)-(6). Conventionally the relationship $(F_+(\mathbf{k}))^* = F_-(-\mathbf{k})$ is used such that F_+ states represent dephasing states (magnitude of \mathbf{k} is increasing) and F_- states represent rephasing states (magnitude of \mathbf{k} is decreasing) thus the evolution is written as:

$$\cos(\alpha) \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \longrightarrow \cos(\alpha) \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \quad (\text{A6a})$$

$$-\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos\left(\frac{\alpha}{2}\right) \longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos\left(\frac{\alpha}{2}\right) \quad (\text{A6b})$$

$$F_+(\mathbf{k} = [1,0,1,1]) \longrightarrow F_+(\mathbf{k} = [1,1,1,1])$$

$$-\frac{i}{\sqrt{2}} e^{i\varphi_1} \cos^2\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right) \longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_1} \cos^2\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right) \quad (\text{A6c})$$

$$F_+(\mathbf{k} = [2,1,1,2]) \longrightarrow F_+(\mathbf{k} = [2,2,1,2])$$

$$-\frac{i}{\sqrt{2}} e^{i(\varphi_1-2\varphi_2)} \sin^3\left(\frac{\alpha}{2}\right) \longrightarrow \frac{i}{\sqrt{2}} e^{i(-\varphi_1+2\varphi_2)} \sin^3\left(\frac{\alpha}{2}\right) \quad (\text{A6d})$$

$$F_-(\mathbf{k} = [0,1,-1,0]) \longrightarrow F_+(\mathbf{k} = [0,0,1,0])$$

$$-\frac{1}{2} e^{i(\varphi_1-\varphi_2)} \sin\left(\frac{\alpha}{2}\right) \sin(\alpha) \longrightarrow -\frac{1}{2} e^{i(\varphi_1-\varphi_2)} \sin\left(\frac{\alpha}{2}\right) \sin(\alpha) \quad (\text{A6e})$$

$$Z(\mathbf{k} = [1,1,0,1]) \longrightarrow Z(\mathbf{k} = [1,1,0,1])$$

The FID pathway, (A6b), becomes progressively dephased as the frequency encoding gradient is applied, thus its contribution to the signal decreases along the acquisition window. The primary echo pathway, (A6d), is rephased as the frequency encoding gradient moves the signal from $-k_{max}$ to 0 in k-space.

Right before the next refocusing pulse (6)-(7), the frequency encoding gradient is applied to finish reading out the line, and B_0 inhomogeneity causes dephasing.

$$\cos(\alpha) \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \longrightarrow \cos(\alpha) \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \quad (\text{A7a})$$

$$-\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos\left(\frac{\alpha}{2}\right) \longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos\left(\frac{\alpha}{2}\right) \quad (\text{A7b})$$

$$F_+(\mathbf{k} = [1,1,1,1]) \longrightarrow F_+(\mathbf{k} = [2,2,1,1])$$

$$\begin{aligned} -\frac{i}{\sqrt{2}} e^{i\varphi_1} \cos^2\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right) &\longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_1} \cos^2\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right) & (A7c) \\ F_+(\mathbf{k} = [2,2,1,2]) & & F_+(\mathbf{k} = [3,3,1,2]) \end{aligned}$$

$$\begin{aligned} \frac{i}{\sqrt{2}} e^{i(-\varphi_1+2\varphi_2)} \sin^3\left(\frac{\alpha}{2}\right) &\longrightarrow \frac{i}{\sqrt{2}} e^{i(-\varphi_1+2\varphi_2)} \sin^3\left(\frac{\alpha}{2}\right) & (A7d) \\ F_+(\mathbf{k} = [0,0,1,0]) & & F_+(\mathbf{k} = [1,1,1,0]) \end{aligned}$$

$$\begin{aligned} -\frac{1}{2} e^{i(\varphi_1-\varphi_2)} \sin\left(\frac{\alpha}{2}\right) \sin(\alpha) &\longrightarrow -\frac{1}{2} e^{i(\varphi_1-\varphi_2)} \sin\left(\frac{\alpha}{2}\right) \sin(\alpha) & (A7e) \\ Z(\mathbf{k} = [1,1,0,1]) & & Z(\mathbf{k} = [1,1,0,1]) \end{aligned}$$

The refocusing pulse, (7)-(8), creates additional coherence pathways:

$$\begin{aligned} \cos(\alpha) \cos\left(\frac{\alpha}{2}\right) &\longrightarrow \cos^2(\alpha) \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) & (A8a) \\ Z(\mathbf{k} = [0,0,0,0]) & & \end{aligned}$$

$$\begin{aligned} &\longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_3} \sin(\alpha) \cos(\alpha) \cos\left(\frac{\alpha}{2}\right) & (A8b) \\ & & F_+(\mathbf{k} = [0,0,0,1]) \end{aligned}$$

$$\begin{aligned} &\longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos^3\left(\frac{\alpha}{2}\right) & (A8c) \\ & & F_+(\mathbf{k} = [2,2,1,3]) \end{aligned}$$

$$\begin{aligned} -\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos\left(\frac{\alpha}{2}\right) &\longrightarrow -\frac{i}{\sqrt{2}} e^{i(\varphi_2-2\varphi_3)} \sin(\alpha) \sin^2\left(\frac{\alpha}{2}\right) \cos\left(\frac{\alpha}{2}\right) & (A8d) \\ F_+(\mathbf{k} = [2,2,1,1]) & & F_-(\mathbf{k} = [2,2,1,1]) \end{aligned}$$

$$\begin{aligned} &\longrightarrow -\frac{1}{2} e^{i(\varphi_2-\varphi_3)} \sin^2(\alpha) \cos\left(\frac{\alpha}{2}\right) & (A8e) \\ & & Z(\mathbf{k} = [2,2,1,2]) \end{aligned}$$

$$\begin{aligned} &\longrightarrow -\frac{i}{\sqrt{2}} e^{i\varphi_1} \cos^4\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right) & (A8f) \\ & & F_+(\mathbf{k} = [3,3,1,4]) \end{aligned}$$

$$\begin{aligned} -\frac{i}{\sqrt{2}} e^{i\varphi_1} \cos^2\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right) &\longrightarrow -\frac{i}{\sqrt{2}} e^{i(\varphi_1-2\varphi_3)} \cos^2\left(\frac{\alpha}{2}\right) \sin^3\left(\frac{\alpha}{2}\right) & (A8g) \\ F_+(\mathbf{k} = [3,3,1,2]) & & F_-(\mathbf{k} = [3,3,1,2]) \end{aligned}$$

$$\begin{aligned} &\longrightarrow -\frac{1}{2} e^{i(\varphi_1-\varphi_3)} \cos^2\left(\frac{\alpha}{2}\right) \sin\left(\frac{\alpha}{2}\right) \sin(\alpha) & (A8h) \\ & & Z(\mathbf{k} = [3,3,1,3]) \end{aligned}$$

$$\begin{aligned} & \frac{i}{\sqrt{2}} e^{i(-\varphi_1+2\varphi_2)} \sin^3\left(\frac{\alpha}{2}\right) \\ & F_+(\mathbf{k} = [1,1,1,0]) \end{aligned} \begin{cases} \nearrow \frac{i}{\sqrt{2}} e^{i(-\varphi_1+2\varphi_2)} \sin^3\left(\frac{\alpha}{2}\right) \cos^2\left(\frac{\alpha}{2}\right) & \text{(A8i)} \\ \rightarrow \frac{i}{\sqrt{2}} e^{i(-\varphi_1+2\varphi_2-2\varphi_3)} \sin^5\left(\frac{\alpha}{2}\right) & \text{(A8j)} \\ \searrow \frac{1}{2} e^{i(-\varphi_1+2\varphi_2-\varphi_3)} \sin^3\left(\frac{\alpha}{2}\right) \sin(\alpha) & \text{(A8k)} \\ & Z(\mathbf{k} = [1,1,1,1]) \end{cases}$$

$$\begin{aligned} & -\frac{1}{2} e^{i(\varphi_1-\varphi_2)} \sin\left(\frac{\alpha}{2}\right) \\ & \sin(\alpha) Z(\mathbf{k} = [1,1,0,1]) \end{aligned} \begin{cases} \nearrow \frac{i}{2\sqrt{2}} e^{i(\varphi_1-\varphi_2+\varphi_3)} \sin\left(\frac{\alpha}{2}\right) \sin^2(\alpha) & \text{(A8l)} \\ \rightarrow -\frac{i}{2\sqrt{2}} e^{i(\varphi_1-\varphi_2-\varphi_3)} \sin\left(\frac{\alpha}{2}\right) \sin^2(\alpha) & \text{(A8m)} \\ \searrow -\frac{1}{2} e^{i(\varphi_1-\varphi_2)} \sin\left(\frac{\alpha}{2}\right) \sin(\alpha) \cos(\alpha) & \text{(A8n)} \\ & Z(\mathbf{k} = [1,1,0,1]) \end{cases}$$

There is once again an FID, (A8b), which experiences some crushing effect from the slice select gradient. Its signal contribution will decline over the acquisition as dephasing from the frequency encoding gradient and B_0 inhomogeneity accumulates:

$$\cos^2(\alpha) \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \rightarrow \cos^2(\alpha) \cos\left(\frac{\alpha}{2}\right) Z(\mathbf{k} = [0,0,0,0]) \quad \text{(A9a)}$$

$$\begin{aligned} & -\frac{i}{\sqrt{2}} e^{i\varphi_3} \sin(\alpha) \cos(\alpha) \cos\left(\frac{\alpha}{2}\right) \\ & F_+(\mathbf{k} = [0,0,0,1]) \end{aligned} \rightarrow \begin{aligned} & -\frac{i}{\sqrt{2}} e^{i\varphi_3} \sin(\alpha) \cos(\alpha) \cos\left(\frac{\alpha}{2}\right) \\ & F_+(\mathbf{k} = [1,1,0,1]) \end{aligned} \quad \text{(A9b)}$$

$$\begin{aligned} & -\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos^3\left(\frac{\alpha}{2}\right) \\ & F_+(\mathbf{k} = [2,2,1,3]) \end{aligned} \longrightarrow \begin{aligned} & -\frac{i}{\sqrt{2}} e^{i\varphi_2} \sin(\alpha) \cos^3\left(\frac{\alpha}{2}\right) \\ & F_+(\mathbf{k} = [3,3,1,3]) \end{aligned} \quad \text{(A9c)}$$

$$\begin{aligned} & -\frac{i}{\sqrt{2}} e^{i(\varphi_2-2\varphi_3)} \sin(\alpha) \sin^2\left(\frac{\alpha}{2}\right) \\ & \cos\left(\frac{\alpha}{2}\right) F_-(\mathbf{k} = [2,2,1,1]) \end{aligned} \longrightarrow \begin{aligned} & -\frac{i}{\sqrt{2}} e^{i(\varphi_2-2\varphi_3)} \sin(\alpha) \sin^2\left(\frac{\alpha}{2}\right) \cos\left(\frac{\alpha}{2}\right) \\ & F_-(\mathbf{k} = [1,1,1,1]) \end{aligned} \quad \text{(A9d)}$$

$$-\frac{1}{2}e^{i(\varphi_2-\varphi_3)}\sin^2(\alpha)\cos\left(\frac{\alpha}{2}\right) \xrightarrow{\quad} -\frac{1}{2}e^{i(\varphi_2-\varphi_3)}\sin^2(\alpha)\cos\left(\frac{\alpha}{2}\right) \quad (\text{A9e})$$

$$Z(\mathbf{k} = [2,2,1,2]) \xrightarrow{\quad} Z(\mathbf{k} = [2,2,1,2])$$

$$-\frac{i}{\sqrt{2}}e^{i\varphi_1}\cos^4\left(\frac{\alpha}{2}\right)\sin\left(\frac{\alpha}{2}\right) \xrightarrow{\quad} -\frac{i}{\sqrt{2}}e^{i\varphi_1}\cos^4\left(\frac{\alpha}{2}\right)\sin\left(\frac{\alpha}{2}\right) \quad (\text{A9f})$$

$$F_+(\mathbf{k} = [3,3,1,4]) \xrightarrow{\quad} F_+(\mathbf{k} = [4,4,1,4])$$

$$-\frac{i}{\sqrt{2}}e^{i(\varphi_1-2\varphi_3)}\cos^2\left(\frac{\alpha}{2}\right)\sin^3\left(\frac{\alpha}{2}\right) \xrightarrow{\quad} -\frac{i}{\sqrt{2}}e^{i(\varphi_1-2\varphi_3)}\cos^2\left(\frac{\alpha}{2}\right)\sin^3\left(\frac{\alpha}{2}\right) \quad (\text{A9g})$$

$$F_-(\mathbf{k} = [3,3,1,2]) \xrightarrow{\quad} F_-(\mathbf{k} = [2,2,1,2])$$

$$-\frac{1}{2}e^{i(\varphi_1-\varphi_3)}\cos^2\left(\frac{\alpha}{2}\right)\sin\left(\frac{\alpha}{2}\right) \xrightarrow{\quad} -\frac{1}{2}e^{i(\varphi_1-\varphi_3)}\cos^2\left(\frac{\alpha}{2}\right)\sin\left(\frac{\alpha}{2}\right)\sin(\alpha) \quad (\text{A9h})$$

$$\sin(\alpha)Z(\mathbf{k} = [3,3,1,3]) \xrightarrow{\quad} Z(\mathbf{k} = [3,3,1,3])$$

$$\frac{i}{\sqrt{2}}e^{i(-\varphi_1+2\varphi_2)}\sin^3\left(\frac{\alpha}{2}\right) \xrightarrow{\quad} \frac{i}{\sqrt{2}}e^{i(-\varphi_1+2\varphi_2)}\sin^3\left(\frac{\alpha}{2}\right)\cos^2\left(\frac{\alpha}{2}\right) \quad (\text{A9i})$$

$$\cos^2\left(\frac{\alpha}{2}\right)F_+(\mathbf{k} = [1,1,1,2]) \xrightarrow{\quad} F_+(\mathbf{k} = [2,2,1,2])$$

$$\frac{i}{\sqrt{2}}e^{i(-\varphi_1+2\varphi_2-2\varphi_3)} \xrightarrow{\quad} -\frac{i}{\sqrt{2}}e^{i(\varphi_1-2\varphi_2+2\varphi_3)}\sin^5\left(\frac{\alpha}{2}\right) \quad (\text{A9j})$$

$$\sin^5\left(\frac{\alpha}{2}\right)F_-(\mathbf{k} = [1,1,1,0]) \xrightarrow{\quad} F_+(\mathbf{k} = [0,0,-1,0])$$

$$\frac{1}{2}e^{i(-\varphi_1+2\varphi_2-\varphi_3)}\sin^3\left(\frac{\alpha}{2}\right) \xrightarrow{\quad} \frac{1}{2}e^{i(-\varphi_1+2\varphi_2-\varphi_3)}\sin^3\left(\frac{\alpha}{2}\right)\sin(\alpha) \quad (\text{A9k})$$

$$\sin(\alpha)Z(\mathbf{k} = [1,1,1,1]) \xrightarrow{\quad} Z(\mathbf{k} = [1,1,1,1])$$

$$\frac{i}{2\sqrt{2}}e^{i(\varphi_1-\varphi_2+\varphi_3)}\sin\left(\frac{\alpha}{2}\right) \xrightarrow{\quad} \frac{i}{2\sqrt{2}}e^{i(\varphi_1-\varphi_2+\varphi_3)}\sin\left(\frac{\alpha}{2}\right)\sin^2(\alpha) \quad (\text{A9l})$$

$$\sin^2(\alpha)F_+(\mathbf{k} = [1,1,0,2]) \xrightarrow{\quad} F_+(\mathbf{k} = [2,2,0,2])$$

$$-\frac{i}{2\sqrt{2}}e^{i(\varphi_1-\varphi_2-\varphi_3)}\sin\left(\frac{\alpha}{2}\right) \xrightarrow{\quad} \frac{i}{2\sqrt{2}}e^{i(-\varphi_1+\varphi_2+\varphi_3)}\sin\left(\frac{\alpha}{2}\right)\sin^2(\alpha) \quad (\text{A9m})$$

$$\sin^2(\alpha)F_-(\mathbf{k} = [1,1,0,0]) \xrightarrow{\quad} F_+(\mathbf{k} = [0,0,0,0])$$

$$-\frac{1}{2}e^{i(\varphi_1-\varphi_2)}\sin\left(\frac{\alpha}{2}\right)\sin(\alpha) \xrightarrow{\quad} -\frac{1}{2}e^{i(\varphi_1-\varphi_2)}\sin\left(\frac{\alpha}{2}\right)\sin(\alpha)\cos(\alpha) \quad (\text{A9n})$$

$$\cos(\alpha)Z(\mathbf{k} = [1,1,0,1]) \xrightarrow{\quad} Z(\mathbf{k} = [1,1,0,1])$$

The FID from the first α pulse is being refocused in (A9d). This pathway has its largest signal contribution at the end of the acquisition period when the frequency encoding gradient effects are fully refocused. It is however partially spoiled by the slice select gradient. There are two echoes, (A9j) and (A9m), at the center of the acquisition. The first,

(A9j), is the desired echo pathway. The receiver phase is set to the phase of this pathway for each average so that this contribution adds constructively. Therefore, a receiver phase of $\theta_{receiver,2} = \varphi_1 - 2\varphi_2 + 2\varphi_3 - \pi$ is used for the second acquisition in the echo train. The (A9m) pathway is a stimulated echo, where the transverse magnetization generated by the 1st RF pulse is flipped back to the longitudinal direction by the 2nd RF pulse and excited by the 3rd RF pulse to rephase in the transverse plane. Since the stimulated echo is in the Z-state during phase encoding, its k_{phase} value is not the same as the desired pathway. It must be eliminated to prevent artifacts resulting from the signal contribution with the incorrect spatial encoding. This pathway is included as a term in the cost function with weighting $A_j = \frac{1}{2\sqrt{2}} \sin\left(\frac{\alpha}{2}\right) \sin^2(\alpha)$ and phase vector $\varphi_j = [-1,1,1]$. The two FID contributions can be included in the cost function similarly.

A1.2 Implementation of Pathway Calculation

A base class, “coherence_pathway” is created which includes parameters such as the refocusing flip angle and echo train length. Relevant information about the pathway such as the amplitude and phase dependence is stored. The elements of the dephasing vector $k_{\Delta B_0}$, k_f and k_s do not all need to be stored to determine if an echo has occurred. In the case of $k_{\Delta B_0}$ and k_f , both have a similar effect of double the Δk increment between α pulses compared to between the first and second RF pulses. Both are also symmetric about the center of the acquisition window. k_s differs only in that the slice select gradient for the $\alpha/2$ pulse is balanced such that both coherence pathways have $k_s = 0$ right before the slice select gradient of the α pulse. As a result, the coherence pathways split from the state in Z during this period have reduced amplitude from the crushing effect. This can be

accounted for with the FID_weight parameter, although in this work a value of 1 was used for the 3D experiments to ignore crushing, and a value of 0 was used to ignore these contributions.

```

1. class coherence_pathway():
2.     def __init__(self, FA, n_pulses, amp=1, min_amp=1e-4, phase_coeffs=None,
3.                 const_phase=0,
4.                 pulses=0, k=0, primary=True, signal=False, TE=9, T1=80, T2=40,
5.                 FID_weight=1):
6.         # default phase_coeffs set to none but should be zeros length n_pulses
7.         if(phase_coeffs is None):
8.             phase_coeffs = np.zeros(n_pulses)
9.         # determine how much relaxation occurs between pulse and echo
10.        relax = self.relaxation(TE/2, T1,T2)
11.
12.        # store information about coherence pathway in attributes
13.        self.pulses = pulses
14.        self.k = k
15.        self.amp = amp * relax
16.        self.phase_coeffs = phase_coeffs
17.        self.const_phase = const_phase
18.        self.primary = primary
19.        self.signal = signal
20.        self.n_pulses = n_pulses

```

The amplitude and phase coefficients that each child node should receive depends on the state of the parent node. Three derived classes are made for the F_+ , F_- and Z states. Each defines its own function for calculating the amplitude and phase coefficients for its child nodes.

```

1. class F(coherence_pathway):
2.
3.     def relaxation(self, t, T1, T2):
4.         return np.exp(-t/T2)
5.
6.     def F_calc(self, FA):
7.         amp = np.cos(FA/2)**2 * self.amp
8.         phase_coeffs = np.copy(self.phase_coeffs)
9.         phase_coeffs[self.pulses] = 0
10.        const_phase = self.const_phase
11.
12.        return amp, phase_coeffs, const_phase
13.
14.     def F_star_calc(self, FA):
15.         amp = np.sin(FA/2)**2 * self.amp
16.         phase_coeffs = np.copy(self.phase_coeffs)
17.         phase_coeffs[self.pulses] = -2

```

```

18.         const_phase = self.const_phase
19.
20.         return amp, phase_coeffs, const_phase
21.
22.     def Z_calc(self, FA):
23.         amp = 0.5 * np.sin(FA) * self.amp
24.         phase_coeffs = np.copy(self.phase_coeffs)
25.         phase_coeffs[self.pulses] = -1
26.         const_phase = (self.const_phase + 3*np.pi/2) % (2*np.pi)
27.
28.         return amp, phase_coeffs, const_phase
29.
30.     def is_Z(self):
31.         return False
32.
33. class F_star(coherence_pathway):
34.
35.     def relaxation(self, t, T1, T2):
36.         return np.exp(-t/T2)
37.
38.     def F_calc(self, FA):
39.         amp = np.sin(FA/2)**2 * self.amp
40.         phase_coeffs = np.copy(self.phase_coeffs)
41.         phase_coeffs[self.pulses] = 2
42.         const_phase = self.const_phase
43.
44.         return amp, phase_coeffs, const_phase
45.
46.     def F_star_calc(self, FA):
47.         amp = np.cos(FA/2)**2 * self.amp
48.         phase_coeffs = np.copy(self.phase_coeffs)
49.         phase_coeffs[self.pulses] = 0
50.         const_phase = self.const_phase
51.
52.         return amp, phase_coeffs, const_phase
53.
54.     def Z_calc(self, FA):
55.         amp = 0.5 * np.sin(FA) * self.amp
56.         phase_coeffs = np.copy(self.phase_coeffs)
57.         phase_coeffs[self.pulses] = 1
58.         const_phase = (self.const_phase + np.pi/2) % (2*np.pi)
59.
60.         return amp, phase_coeffs, const_phase
61.
62.     def is_Z(self):
63.         return False
64.
65.
66. class Z(coherence_pathway):
67.
68.     def relaxation(self, t, T1, T2):
69.         return np.exp(-t/T1)
70.
71.     def F_calc(self, FA):
72.         amp = np.sin(FA) * self.amp
73.         phase_coeffs = np.copy(self.phase_coeffs)
74.         phase_coeffs[self.pulses] = 1
75.         const_phase = (self.const_phase + 3*np.pi/2) % (2*np.pi)
76.
77.         return amp, phase_coeffs, const_phase
78.

```

```

79.     def F_star_calc(self, FA):
80.         amp = np.sin(FA) * self.amp
81.         phase_coeffs = np.copy(self.phase_coeffs)
82.         phase_coeffs[self.pulses] = -1
83.         const_phase = (self.const_phase + np.pi/2) % (2*np.pi)
84.
85.         return amp, phase_coeffs, const_phase
86.
87.     def Z_calc(self, FA):
88.         # amplitude is negative for 180, currently don't account for this
89.         amp = abs(np.cos(FA)) * self.amp
90.         phase_coeffs = np.copy(self.phase_coeffs)
91.         phase_coeffs[self.pulses] = 0
92.         const_phase = self.const_phase
93.
94.         return amp, phase_coeffs, const_phase
95.
96.     def is_Z(self):
97.         return True

```

An initial node of the derived Z class represents the system at thermal equilibrium. This node generates additional nodes that represent the states just before the first refocusing pulse. These nodes branch to represent the states at the center of the acquisition window which branch for the next window and so on. A logical structure was included in the constructor of the base class to determine if the node should branch further depending on its amplitude and whether there are additional pulses left in the echo train. Logical statements are also included to determine if measurable signal is generated by the child node on creation. Flag are passed to the children indicating if they are the desired signal pathway (primary) or unwanted signal.

```

1. # if there is an additional pulse after, create children
2. if(pulses<n_pulses):
3.     if(self.amp>min_amp):
4.
5.         # excitation pulse is special case (half flip angle)
6.         if(pulses==0):
7.
8.             # only need one child (F* redundant, Z has different echo locations)
9.             amp, phase_coeffs, const_phase = self.F_calc(FA/2)
10.            self.F_child = F(FA, n_pulses, amp, min_amp, phase_coeffs,
11.                            const_phase, pulses=pulses+1, k=1,
12.                            primary=True, signal=False, TE=TE, T1=T1, T2=T2)
13.

```

```

14.         # redundant
15.         self.F_star_child = None
16.
17.         # gives rise to FIDs
18.         amp, phase_coeffs, const_phase = self.Z_calc(FA/2)
19.         # apply FID weight to account for T2* and Spoiling effects
20.         self.Z_child = Z(FA, n_pulses, amp*relax*FID_weight, min_amp,
21.                         phase_coeffs, const_phase,
22.                         pulses=pulses+1, k=abs(k), primary=False,
23.                         signal=False, TE=TE, T1=T1, T2=T2)
24.
25.     else:
26.         # child in the F state
27.         amp, phase_coeffs, const_phase = self.F_calc(FA)
28.
29.         if(k==0):
30.             # FID
31.             flag = True
32.         else:
33.             # not FID
34.             flag = False
35.
36.         # abs of k to ensure it is positive +2 for additional dephasing
37.         self.F_child = F(FA, n_pulses, amp*relax, min_amp,
38.                         phase_coeffs, const_phase, pulses=pulses+1,
39.                         k=abs(k)+2, primary=False, signal=flag,
40.                         TE=TE, T1=T1, T2=T2)
41.
42.         # child in the F* states
43.         amp, phase_coeffs, const_phase = self.F_star_calc(FA)
44.
45.         if(k==1):
46.             # special case (goes to F*(-1) then to F(1))
47.             # this creates an echo
48.             phase_coeffs = -phase_coeffs
49.             const_phase = -const_phase
50.             self.F_star_child = F(FA, n_pulses, amp*relax, min_amp,
51.                                 phase_coeffs, const_phase,
52.                                 pulses=pulses+1, k=1,
53.                                 primary=primary, signal=True,
54.                                 TE=TE, T1=T1, T2=T2)
55.         elif(k==-1):
56.             # special case F*(-1) to F*(-1) after RF pulse then to F(1)
57.             # this also creates an echo
58.             phase_coeffs = -phase_coeffs
59.             const_phase = -const_phase
60.             self.F_star_child = F(FA, n_pulses, amp*relax, min_amp,
61.                                 phase_coeffs, const_phase,
62.                                 pulses=pulses+1, k=1,
63.                                 primary=primary, signal=True,
64.                                 TE=TE, T1=T1, T2=T2)
65.         elif(k==0):
66.             # FID or echo at start of acquisition
67.             if(self.is_Z()):
68.                 # FID
69.                 self.F_star_child = None
70.             else:
71.                 # echo at end
72.                 phase_coeffs = -phase_coeffs
73.                 const_phase = -const_phase
74.                 self.F_star_child = F(FA, n_pulses, amp*relax, min_amp,

```

```

75.             phase_coeffs, const_phase,
76.             pulses=pulses+1, k=1,
77.             primary=primary, signal=True,
78.             TE=TE, T1=T1, T2=T2)
79.         elif(abs(k)==2):
80.             # echo at end of acquisition
81.             phase_coeffs = -phase_coeffs
82.             const_phase = -const_phase
83.             self.F_star_child = F(FA, n_pulses, amp*relax, min_amp,
84.             phase_coeffs, const_phase,
85.             pulses=pulses+1, k=0,
86.             primary=primary, signal=True,
87.             TE=TE, T1=T1, T2=T2)
88.         else:
89.             # still in F*(-k) state
90.             self.F_star_child = F_star(FA, n_pulses, amp*relax, min_amp,
91.             phase_coeffs, const_phase,
92.             pulses=pulses+1, k=-abs(k)+2,
93.             primary=primary, signal=False,
94.             TE=TE, T1=T1, T2=T2)
95.
96.
97.             # child in the Z states
98.             amp, phase_coeffs, const_phase = self.Z_calc(FA)
99.             self.Z_child = Z(FA, n_pulses, amp*relax, min_amp,
100.             phase_coeffs, const_phase,
101.             pulses=pulses+1, k=abs(k), primary=False,
102.             signal=False, TE=TE, T1=T1, T2=T2)
103.
104.         else:
105.             # don't need any further calculations because small contribution
106.             self.F_child = None
107.
108.             self.F_star_child = None
109.
110.             self.Z_child = None
111.     else:
112.         # don't need any further calculations after last pulse
113.         self.F_child = None
114.
115.         self.F_star_child = None
116.
117.         self.Z_child = None

```

The base class contained a function which returned the amplitude and phase information of all primary echoes as the root node determined the primary pathway child and queried it. Each child then checked if it had a child which was primary and so on until the end of the branch was reached. This child returned its amplitude and phase information to its parent which appended its own amplitude and phase and returned the list. This continued until a full list was generated by the root node. A similar function was created to

retrieve the amplitude and phase information of all signal contributions at a particular echo location.

```
1. def get_primary_pathways(self):
2.     """
3.     Gets the phase_coeffs, amplitudes, and phase constants of each primary
4.     echo for all children.
5.
6.     Returns
7.     -----
8.     phase_coeffs : list
9.         List of arrays of the coefficients (multiple with the phases of
10.        the RF pulses) for each primary echo.
11.     amp : list
12.        List of amplitudes of primary echoes.
13.     const_phase : list
14.        List of the constant part (not dependent on RF phase) for each
15.        primary echoes.
16.
17.     """
18.
19.     if(self.F_child is None):
20.
21.         # then there are no children and we need to initialize the list if
22.         primary
23.
24.         if(self.primary and self.signal):
25.             phase_coeffs = [self.phase_coeffs]
26.             amp = [self.amp]
27.             const_phase = [self.const_phase]
28.         else:
29.             return
30.
31.     else:
32.
33.         # if excitation just directly return without appending (no echo)
34.         # also exclude the initial object which is prior to the excitation
35.
36.         # else combine current echo with child echoes
37.         if(self.F_child.primary):
38.             phase_coeffs, amp, const_phase = self.F_child.get_primary_pathways()
39.
40.         if(self.pulses>1):
41.             phase_coeffs.append(self.phase_coeffs)
42.             amp.append(self.amp)
43.             const_phase.append(self.const_phase)
44.
45.         elif(self.F_star_child.primary):
46.             phase_coeffs, amp, const_phase =
47.             self.F_star_child.get_primary_pathways()
48.
49.         if(self.pulses>1):
50.             phase_coeffs.append(self.phase_coeffs)
51.             amp.append(self.amp)
52.             const_phase.append(self.const_phase)
53.
54.     return phase_coeffs, amp, const_phase
```

```

53.
54.
55. def get_echoes(self, pulse, primary=False):
56.     """
57.     Get all coherence pathways giving an echo imidiately after the nth pulse
58.
59.     Parameters
60.     -----
61.     pulse : int
62.         Index of the refocussing pulse immediately before the echo
63.         (index starting at zero).
64.     primary : Boolean, optional
65.         Whether or not to include the primary echo pathway. The default is
66.         False.
67.     Returns
68.     -----
69.     phase_coeffs : list
70.         List of arrays of the coefficients (multiple with the phases of
71.         the RF pulses) for each primary echo.
72.     amp : list
73.         List of amplitudes of primary echoes.
74.     const_phase : list
75.         List of the constant part (not dependent on RF phase) for each
76.         primary echoes.
77.
78.     """
79.
80.     # +1 for excitation, +1 for initial object which is prior to the excitation
81.     if(self.pulses==pulse+2):
82.
83.         # the next pulse is the one of interest (the echo after it really)
84.
85.         if(self.signal):
86.             if(not(primary) and self.primary):
87.
88.                 # then we don't want this included
89.                 phase_coeffs = []
90.                 amp = []
91.                 const_phase = []
92.
93.             else:
94.                 # then we should begin the list here
95.                 phase_coeffs = [self.phase_coeffs]
96.                 amp = [self.amp]
97.                 const_phase = [self.const_phase]
98.         else:
99.             # then we don't want this included
100.            phase_coeffs = []
101.            amp = []
102.            const_phase = []
103.
104.         else:
105.
106.             if not(self.F_child is None):
107.                 phase_coeffs, amp, const_phase =
108.                 self.F_child.get_echoes(pulse,primary)
109.
110.            # F* contributions
111.            if not(self.F_star_child is None):

```

```

111.         phase_coeffs_, amp_, const_phase_ =
self.F_star_child.get_echoes(pulse,primary)
112.
113.         # append to other pathways
114.         phase_coeffs.extend(phase_coeffs_)
115.         amp.extend(amp_)
116.         const_phase.extend(const_phase_)
117.
118.         # Z contributions
119.         phase_coeffs_, amp_, const_phase_ =
self.Z_child.get_echoes(pulse,primary)
120.
121.         # append to other pathways
122.         phase_coeffs.extend(phase_coeffs_)
123.         amp.extend(amp_)
124.         const_phase.extend(const_phase_)
125.
126.     else:
127.         # nothing from this branch
128.         phase_coeffs = []
129.         amp = []
130.         const_phase = []
131.
132.
133.     return phase_coeffs, amp, const_phase

```

Appendix 2

Code for the implementation of the DLLR-CS method described in Chapter 3:

```

1. from scipy import io
2. import numpy as np
3. import matplotlib.pyplot as plt
4. import pickle
5. from scipy import stats
6. from scipy.signal import correlate2d
7.
8. from sklearn.decomposition import MiniBatchDictionaryLearning
9. from sklearn.feature_extraction.image import extract_patches_2d
10. from sklearn.feature_extraction.image import reconstruct_from_patches_2d
11.
12. """user defined parameters"""
13. # at most set to twice the number of cpu cores but depending on memory might
14. # have to go lower
15. threads = 1
16. # filename for loading data
17. #k_space_filename = "phantom_20210822.mat"
18. #k_space_filename = "phantom_simulated_spokes_80.mat"
19. k_space_filename = "full_dataset_40_lines_9859_2022.mat"
20.
21. # good ones 3 (0.05,0.04,0.5), 10 (0.001,0.001,0.5)
22. # 0.05, 0.05,0.5,10
23.
24. #weighting of dictionary
25. lam3 = 0.05
26. # weight of low rank approximation
27. lam4 = 0.01

```

```

28. # weight of central data
29. lam2 = 1
30. # soft threshold for svd
31. lam1 = 0.1
32. # list of noise levels
33. noise = 15
34.
35. # these default settings are generally good
36. # rank of svd
37. r = 5
38. # number of nonzero DL coefficeints to use
39. coefs = 4
40.
41. # how many frames reconstructed per fully sampled image (fully sampled included)
42. frames = 5
43. # whether or not to perform motion correction
44. correction = False
45.
46. def high_pass_filter(image, coefficient, var):
47.     """
48.     1 - guassian filter. Enhances edges and amplifies noise.
49.
50.     Parameters
51.     -----
52.     image : array
53.         2D array.
54.     coefficient : float
55.         Coefficientout front the gaussian.
56.     var : float
57.         Determines width of gaussian.
58.
59.     Returns
60.     -----
61.     array
62.         High-pass filtered image.
63.
64.     """
65.
66.     x = np.linspace(-1,1,image.shape[0])
67.     y = np.linspace(-1,1,image.shape[1])
68.
69.     X, Y = np.meshgrid(y, x)
70.
71.     filt = 1 - coefficient * np.exp(-(X**2 + Y**2)/var)
72.
73.     return abs(cifft2(cfft2(image) * filt))
74.
75. def roll(x):
76.     """
77.     For circularly shifting brain into center
78.
79.     Parameters
80.     -----
81.     x : array
82.         image to circularly shift.
83.
84.     Returns
85.     -----
86.     array
87.         circularly shifted image.
88.

```

```

89.     """
90.     return np.roll(x,-40,1)
91.
92. def cfft(x):
93.     """
94.     Performs Fourier transform along first axis
95.
96.     Parameters
97.     -----
98.     x : array
99.         Array to perform 1D Fourier transform on.
100.
101.     Returns
102.     -----
103.     array
104.         The result of the 1D Fourier transform.
105.
106.     """
107.
108.     return np.fft.fftshift(np.fft.fft(np.fft.ifftshift(x)))
109.
110.     def ciff2(x):
111.         """
112.         Performs inverse Fourier transform along first axis
113.
114.         Parameters
115.         -----
116.         x : array
117.             Array to perform 1D inverse Fourier transform on.
118.
119.         Returns
120.         -----
121.         array
122.             The result of the 1D inverse Fourier transform.
123.
124.         """
125.
126.         return np.fft.fftshift(np.fft.ifft(np.fft.ifftshift(x)))
127.
128.     def cfft2(x):
129.         """
130.         Performs Fourier transform along first two axis
131.
132.         Parameters
133.         -----
134.         x : array
135.             Array to perform 2D Fourier transform on.
136.
137.         Returns
138.         -----
139.         array
140.             The result of the 2D Fourier transform.
141.
142.         """
143.
144.         return
145.         np.fft.fftshift(np.fft.fftn(np.fft.ifftshift(x,axes=[0,1]),axes=[0,1]),axes=[0,1
146.         ])
147.
148.     def ciff2(x):
149.         """

```

```

148.         Performs inverse Fourier transform along first two axis
149.
150.         Parameters
151.         -----
152.         x : array
153.             Array to perform 2D inverse Fourier transform on.
154.
155.         Returns
156.         -----
157.         array
158.             The result of the inverse 2D Fourier transform.
159.
160.         """
161.
162.         return
163.         np.fft.fftshift(np.fft.ifftn(np.fft.ifftshift(x, axes=[0,1]), axes=[0,1]), axes=[0,
164.         1])
165.
166.     def soft_threshold(x,alpha):
167.         """
168.         Returns an array with same shape as x with all values with smaller
169.         magnitude than alpha set to zero, and all others having their
170.         magnitude
171.         reduced by alpha
172.
173.         Parameters
174.         -----
175.         x : array
176.             An array of any shape.
177.         alpha : float
178.             Thresholding parameter.
179.
180.         Returns
181.         -----
182.         array
183.             The result of the soft thresholding.
184.
185.         """
186.         return np.maximum(np.abs(x)-alpha,0)*np.sign(x)
187.
188.     def get_shift(k_sam):
189.         """
190.
191.         Parameters
192.         -----
193.         k_sam : array
194.             K-space measurements.
195.
196.         Returns
197.         -----
198.         v_shift : array
199.             Shift in vertical direction.
200.         h_shift : array
201.             Shift in horizontal direction.
202.
203.         """
204.         v_shift = np.zeros(k_sam.shape[-1])
205.         h_shift = np.zeros(k_sam.shape[-1])

```

```

206.         for i in range(2*frames-1,k_sam.shape[-1],frames):
207.             # find shift of fully sampled image from cross-correlation
208.             cor = correlate2d(abs(cifft2(k_sam[:, :, frames-1])),
209.                             abs(cifft2(k_sam[:, :, i])))
210.             temp = np.unravel_index(cor.argmax(), cor.shape)
211.
212.             v_shift[i] = temp[0] - cor.shape[0]/2
213.             h_shift[i] = temp[1] - cor.shape[1]/2
214.
215.             # linear interpolation for other frames
216.             v_shift[i-frames:i] = np.linspace(v_shift[i-frames],
217.                                               v_shift[i],frames)
218.             h_shift[i-frames:i] = np.linspace(h_shift[i-
frames],h_shift[i],frames)
219.
220.         return (v_shift,h_shift)
221.
222.
223.     def corrections(k1, k_sam, v_shift, h_shift):
224.         """
225.         Performs horizontal and verticle shift corrections as well as
226.         zeroth order phase correction
227.
228.         Parameters
229.         -----
230.         k1 : array
231.             The frame being corrected.
232.         k_sam : array
233.             The whole set of k-space data (realized not needed but don't
want to
234.             make big changes before I send it
and
235.             passed by reference so not a big
deal).
236.         v_shift : float
237.             How far the current frame is shifted vertically.
238.         h_shift : float
239.             How far the current frame is shifted horizontally.
240.
241.         Returns
242.         -----
243.         k1 : array
244.             k1 frame with corrections applied (note unsampled points not
exactly
245.             zero due to floating point
error).
246.
247.         """
248.
249.         # horizontal shift correction
250.         #x = np.linspace(-0.5,0.5,k1.shape[1])
251.         sz = k1.shape[1]
252.         x = np.linspace(-sz/2+1,sz/2,sz)/sz
253.         correction = np.exp(-1j * 2 * np.pi * x * (h_shift))
254.         k1 = k1 * correction[None,:]
255.         # vertical phase correction
256.         #x = np.linspace(-0.5,0.5,k1.shape[0])
257.         sz = k1.shape[0]
258.         x = np.linspace(-sz/2+1,sz/2,sz)/sz
259.         correction = np.exp(-1j * 2 * np.pi * x * (v_shift))
260.         k1 = k1 * correction[:,None]

```

```

261.         # 0th order phase correction
262.         #correction = np.angle(k1[k_sam.shape[0]//2,k1.shape[1]//2])
263.         #k1 = cfft2(cifft2(k1) * np.exp(-1j * correction))
264.         return k1
265.
266.     def compute_density(sampling_scheme):
267.         """
268.         Computes sampling density for density compensation
269.
270.         Parameters
271.         -----
272.         sampling_scheme : array
273.             boolean array of sampled vs unsampled points.
274.
275.         Returns
276.         -----
277.         Z : array
278.             Array of sampling density at each point.
279.
280.         """
281.
282.         mat = np.array(sampling_scheme,dtype=np.bool)
283.         X, Y = np.mgrid[0:mat.shape[0], 0:mat.shape[1]]
284.         positions = np.vstack([X.ravel(), Y.ravel()])
285.         values = np.vstack([X[mat].ravel(), Y[mat].ravel()])
286.         kernel = stats.gaussian_kde(values, bw_method=5e-1)
287.         Z = np.reshape(kernel(positions).T, X.shape)
288.
289.         Z = np.tile(Z[mat.shape[0]//2,:][None,...],(mat.shape[0],1))
290.         Z /= np.mean(Z[mat])
291.
292.         return Z
293.
294.     def dl_reconstruction(image, dico, patch_size):
295.         """
296.         Calculate dictionary learning (DL) coefficients along first two
297.         axes.
298.         Calculate reconstruction for that slice, iterate over third axis
299.
300.         Parameters
301.         -----
302.         image : array
303.             Image to calculate DL transform on.
304.         dico : MiniBatchDictionaryLearning object
305.             Dictionary of atoms and transform parameters.
306.         patch_size : tuple
307.             Dimensions of patch.
308.
309.         Returns
310.         -----
311.         reco : array
312.             DL reconstruction.
313.
314.         """
315.         V = dico.components_
316.         reco = np.zeros(image.shape)
317.
318.         # roll so there isn't a cutoff in the brain
319.         image = np.roll(image,-20,axis=1)
320.

```

```

321.         for i in range(image.shape[-1]):
322.             # extract patches of noisy image
323.             patches = extract_patches_2d(abs(image[:, :, i]), patch_size)
324.             patches = patches.reshape(patches.shape[0], -1)
325.             intercept = np.mean(patches, axis=0)
326.             patches -= intercept
327.             length = len(patches)
328.
329.             # sparse representation of data
330.             code = dico.transform(patches)
331.
332.             # return to image domain
333.             patches = np.dot(code, V)
334.             patches += intercept
335.             patches = patches.reshape(length, *patch_size)
336.
337.             reco[:, :, i] = reconstruct_from_patches_2d(patches,
reco.shape[0:2])
338.
339.             # roll back to original
340.             reco = np.roll(reco, 20, axis=1)
341.             return reco
342.
343.     def low_rank_approx(SVD=None, A=None, r=1):
344.         """
345.         Computes an r-rank approximation of a matrix
346.         given the component u, s, and v of it's SVD
347.         """
348.         if not SVD:
349.             SVD = np.linalg.svd(A, full_matrices=False)
350.         u, s, v = SVD
351.         u = u[:, :r]
352.         s = s[:r]
353.         v = v[:r, :]
354.         reco = (u * s) @ v
355.         return reco
356.
357.
358.     def svd_threshold(SVD=None, A=None, lam1=0.1):
359.         """
360.         Computes approximation of a matrix cutting off singular values below
361.         lam0 given the component u, s, and v of it's SVD. (Not in use)
362.         """
363.         if not SVD:
364.             SVD = np.linalg.svd(A, full_matrices=False)
365.         u, s, v = SVD
366.         s = soft_threshold(s, lam1)
367.         reco = (u * s) @ v
368.         return reco
369.
370.     def low_rank(image, lam1):
371.         """
372.         Performs SVD approximation to r singular values along each phase
encoding/
373.         temporal slice of the image
374.
375.         Parameters
376.         -----
377.         image : array
378.             3D array where the first dimension is iterated along and SVD
379.             approximation calculated along second and third.

```

```

380.         r : int
381.             Number of singular values to keep.
382.
383.         Returns
384.         -----
385.         reco : array
386.             Result of approximation.
387.
388.         """
389.
390.         reco = np.zeros(image.shape)
391.
392.         for j in range(image.shape[0]):
393.             reco[j,:,:] = svd_threshold(A=abs(image[j,:,:]), lam1=lam1)
394.
395.         return reco
396.
397.     def x_step(X, dic, lowr, k_un, k_full, phase, mat, c_mask, lam2, lam3,
lam4):
398.         """
399.         Update the guess for the reconstruction X
400.
401.         Parameters
402.         -----
403.         X : array
404.             The current guess of the image.
405.         dic : array
406.             Result of dictionary learning reconstruction .
407.         lowr : array
408.             Result of low rank approximation.
409.         k_sam : array
410.             Sampled k-space.
411.         phase : array
412.             Phase of image.
413.         mat : array
414.             Boolean array of sampled points.
415.         c_mask : array
416.             mask for the center of k-space.
417.         lam2 : float
418.             weight of central data.
419.         lam3 : float
420.             weigthing of dictionary.
421.         lam4 : float
422.             weight of low rank approximation.
423.
424.         Returns
425.         -----
426.         array
427.             New guess of the image.
428.
429.         """
430.
431.         temp = cfft2(np.exp(1j*phase) * (lam3 * dic + lam4 * lowr))
432.
433.         # list to relate undersampled and fully sampled frames
434.         ful_idx = (frames-1)//2*[0]
435.         for i in range(1,k_full.shape[-1]):
436.             ful_idx += (frames-1)*[i]
437.         ful_idx+=(frames-1)//2*[k_full.shape[-1]-1]
438.
439.         for i in range(k_un.shape[-1]):

```

```

440.
441.         # sampled
442.         k_reco = mat[...,i] * (k_un[...,i] + temp[...,i]) / (1 + lam3 +
lam4)
443.         # unsampled center
444.         k_reco += ((1 - mat[...,i]) * c_mask[...,i] *
445.                   (temp[...,i] + lam2*k_full[...,ful_idx[i]]) / (lam3 +
lam4 + lam2))
446.         # unsampled exterior
447.         k_reco += (1 - mat[...,i]) * (1 - c_mask[...,i]) * temp[...,i] /
(lam3 + lam4)
448.
449.         X[...,i] = np.abs(cifft2(k_reco))
450.
451.         return abs(X)
452.
453.     def fit_transform(dico, image, patch_size):
454.         """
455.         Fit dictionary and transform coefficients given an image
456.
457.         Parameters
458.         -----
459.         dico : MiniBatchDictionaryLearning object
460.         Sklearn MiniBatchDictionaryLearning object with desired parameters
set.
461.         image : array
462.         Image on which to perform the fit transform.
463.         patch_size : tuple
464.         Size of patches and atoms.
465.
466.         Returns
467.         -----
468.         reco : array
469.         image resulting from fit transform.
470.
471.         """
472.         # extract patches of noisy image
473.         patches = extract_patches_2d(image, patch_size)
474.         patches = patches.reshape(patches.shape[0], -1)
475.         intercept = np.mean(patches, axis=0)
476.         patches -= intercept
477.         length = len(patches)
478.         # sparse representation of data
479.         code = dico.fit_transform(patches)
480.         V = dico.components_
481.         # return to image domain
482.         patches = np.dot(code, V)
483.         patches += intercept
484.         patches = patches.reshape(length, *patch_size)
485.         reco = reconstruct_from_patches_2d(patches, image.shape)
486.         return reco
487.
488.     def reconstruction(idx_slice):
489.         print(idx_slice)
490.         # load all fully and undersampled frames
491.         k_sam = io.loadmat(k_space_filename)["k_all"][:,:,idx_slice,:]
492.
493.         # set fully sampled frames
494.         k_full = k_sam[:, :, frames-1::frames]
495.         # set undersampled frames

```

```

496.         k_un =
np.zeros((k_sam.shape[0],k_sam.shape[1],k_full.shape[2]*(frames-
1)),dtype='complex128')
497.         un_idx = 0
498.         for i in range(k_sam.shape[-1]):
499.             if((i+1)%frames!=0):
500.                 k_un[:, :, un_idx] = k_sam[:, :, i]
501.                 un_idx += 1
502.         del k_sam, un_idx
503.
504.         center = k_full.shape[1]//2
505.
506.         # mask
507.
508.         # mask
509.         mat = 1 - 1*(k_un==0)
510.         '''
511.         if(correction):
512.             # determine shift for motion correction
513.             v_shift, h_shift = get_shift(k_sam)
514.
515.             for i in range( k_sam.shape[-1]):
516.                 # apply correction
517.                 k_sam[:, :, i] = corrections(np.copy(k_sam[:, :, i]), k_sam,
518.                                                 v_shift[0] - v_shift[i],
519.                                                 h_shift[0] - h_shift[i])
520.             '''
521.
522.         # create mask for center k_space points (given some weight in
reconstruction)
523.         c_mask = np.zeros(k_un.shape)
524.
525.         x, y= np.meshgrid(np.linspace(-
k_un.shape[1]//2,k_un.shape[1]//2,k_un.shape[1]),
526.                           np.linspace(-
k_un.shape[0]//2,k_un.shape[0]//2,k_un.shape[0]))
527.
528.         c_mask[:,center-15:center+15,:] = 1
529.
530.         # make list of nearest fully sampled lines for each undersampled
frame
531.         ful_idx = (frames-1)//2*[0]
532.         for i in range(1,k_full.shape[-1]):
533.             ful_idx += (frames-1)*[i]
534.             ful_idx+=(frames-1)//2*[k_full.shape[-1]-1]
535.
536.         # allocate memory for initial guess and phase
537.         image = np.zeros(k_un.shape,dtype='complex128')
538.         phase = np.zeros(k_un.shape)
539.         # create initial guess for each slice
540.         for i in range(k_un.shape[-1]):
541.
542.             # phase is equal to phase of nearest fully sampled frame
543.             temp = (1 - mat[... ,i]) * k_full[:, :, ful_idx[i]] + mat[... ,i] *
k_un[... ,i]
544.             temp = ciff2(temp)
545.             phase[:, :, i] = np.angle(temp)
546.
547.             # initial guess from the nearest fully sampled frame + noise
548.             temp = abs(temp)
549.             np.random.seed(seed=(i+1)*4321)

```

```

550.         image[:, :, i] = temp +
np.random.normal(scale=np.max(abs(temp)/noise),
551.                                                         size=k_full.shape[
:-1])
552.
553.
554.
555.         # loading dictionary
556.         patch_sz = 7
557.         patch_size = (patch_sz, patch_sz)
558.
559.         ...
560.         dico_dot = pickle.load(open("dot_dictionary.txt", 'rb'))
561.         dico_brain = pickle.load(open("brain_dictionary_phantom.txt", 'rb'))
562.         dico_edge = pickle.load(open("mouse_edges_dictionary.txt", 'rb'))
563.         ...
564.
565.         dico = pickle.load(open("combine_dictionary_with_2x2.txt", 'rb'))
566.
567.         dico.transform_algorithm = 'omp'
568.         dico.transform_n_nonzero_coefs = coefs
569.
570.         X = np.abs(image)
571.         del image
572.
573.         # dictionary learning compressed sensing
574.
575.         for i in range(4):
576.
577.             print(i)
578.             dic = dl_reconstruction(X, dico, patch_size)
579.             lowr = low_rank(X, r)
580.
581.             X = x_step(X, dic, lowr, k_un, k_full, phase, mat, c_mask, lam2,
lam3, lam4)
582.
583.             dic = dl_reconstruction(X, dico, patch_size)
584.             lowr = low_rank(X, r)
585.
586.             X = x_step(X, dic, lowr, k_un, k_full, phase, mat, c_mask, 0, lam3,
0)
587.
588.             return dic, X, abs(cifft2(k_full))
589.
590.         %% running reconstruction on all frames
591.
592.         # for multithreading
593.         from multiprocessing import Pool
594.
595.         if __name__ == '__main__':
596.             # start timer
597.             import time
598.             start = time.time()
599.
600.             # calls reconstruction for threads number of slices at once
601.             with Pool(threads) as p:
602.                 data = p.map(reconstruction, [0])
603.
604.             # end timer
605.             end = time.time()
606.         print(end - start)

```

VITA AUCTORIS

NAME: Mark Armstrong

PLACE OF BIRTH: Windsor, ON

YEAR OF BIRTH: 1998

EDUCATION: Essex District High School, Essex, ON, 2016

University of Windsor, B.Sc., Windsor, ON, 2020