

# An implementation of an open-source MRI sequencing framework

Supervisor: Frank Simonis

Sander Kiewiet (s2177250)

Bachelor Thesis

MD&I - Techmed Centre - University of Twente

July 10, 2023

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Objective . . . . .	4
<b>2</b>	<b>Theory</b>	<b>5</b>
2.1	Clinical TSE sequence modeling . . . . .	5
2.2	k-space and data acquisition . . . . .	6
2.2.1	k-space undersampling . . . . .	6
2.3	The Pulseq framework . . . . .	7
<b>3</b>	<b>Method</b>	<b>8</b>
3.1	Sequence variants . . . . .	9
3.1.1	TSE sequence (A) . . . . .	10
3.1.2	Modified TSE sequence (A.1-3) . . . . .	11
3.1.3	Modeled TSE (B) . . . . .	12
3.1.4	Modeled Modified TSE sequence (B.1-3) . . . . .	12
3.2	Executing the generated sequences . . . . .	13
3.3	Image Reconstruction . . . . .	13
<b>4</b>	<b>Results</b>	<b>14</b>
4.1	Basic TSE sequence . . . . .	14
4.2	Clinical TSE sequence . . . . .	15
4.3	TSE sequence (A) . . . . .	15
4.4	Modified TSE sequences . . . . .	16
4.4.1	Modified TSE sequence (A.1) . . . . .	16
4.4.2	Modified TSE sequence (A.2) . . . . .	16
4.4.3	Modified TSE sequence (A.3) . . . . .	17
4.5	Modeled TSE sequence (B) . . . . .	18
4.6	Modeled Modified TSE sequences . . . . .	18
4.6.1	Modeled Modified TSE sequence (B.1) . . . . .	18
4.6.2	Modeled Modified TSE sequence (B.2) . . . . .	19
4.6.3	Modeled Modified TSE sequence (B.3) . . . . .	19
<b>5</b>	<b>Conclusion</b>	<b>20</b>
<b>6</b>	<b>Discussions</b>	<b>20</b>
6.1	Installation . . . . .	20
6.2	Effective echo time . . . . .	20
6.3	k-space dimensions and resolution . . . . .	21
6.4	Modifications to the sequences . . . . .	21
6.5	Improvements to Pulseq sequence design . . . . .	21
6.6	Future works . . . . .	21
	<b>References</b>	<b>22</b>

<b>A Appendix</b>	<b>23</b>
A.1 Manual	23
A.1.1 MATLAB Step 1	23
A.1.2 Scanner	24
A.1.3 MATLAB Step 2	24
A.2 Global simulated TSE values	25
A.3 The Pulseq sequence file format	26
A.4 Code Modifications	27

## Abstract

The primary objective of this project was to integrate the Pulseq Magnetic Resonance Imaging (MRI) sequence design platform into a Siemens Aera AG clinical MRI scanner (1.5T). This was accomplished by modeling and modifying a clinical Turbo Spin Echo (TSE) sequence commonly used for prostate imaging, with the assistance of the Pulseq platform. The sequences were generated and analyzed using MATLAB, and the modified variants were uploaded to the MRI scanner. An MRI phantom served as the imaging object, and the acquired data were thoroughly analyzed to draw conclusions regarding the efficacy of the modifications, the sequence modeling process, and the resulting image quality. The successful implementation of the Pulseq platform on the scanner enabled the desired sequence modifications. However, the image quality of the modified sequences fell short compared to the existing clinical TSE sequence.

## 1 Introduction

Magnetic Resonance Imaging (MRI) is a widely used non-invasive imaging technique in diagnostic medicine. MRI has now become an essential tool for visualizing and analyzing internal structures of the human body. MRI offers high spatial resolution and good tissue contrast, making this technique suitable for detecting and characterizing a wide range of pathological conditions.

Magnetic Resonance (MR) sequence design is a critical aspect of magnetic resonance imaging that governs the acquisition of data and plays a fundamental role in shaping the resulting image contrast and resolution. MR sequences are protocols that determine the timing and parameters of radiofrequency (RF) pulses and gradient fields applied during the scan. By manipulating these parameters, sequence design allows for the optimization of image acquisition to highlight specific tissue properties and enhance image quality.

While clinically approved sequences are typically pre-installed on most scanners, designing experimental sequences requires custom software. Currently, experimental sequence design on clinical scanners often relies on manufacturer-specific closed-source software. However, this approach generates sequences incompatible with software from other manufacturers, making it challenging to share sequence files among researchers at different institutions using different hardware and software. Moreover, manufacturer-specific software for MR sequence design is often a low-level programming language, in which it is difficult to visualize newly built MR sequences.

In 2017, researchers at the University of Freiburg in Germany released an open-source MR framework known as ‘Pulseq’. The primary objective of Pulseq is to provide a universal and manufacturer-independent file format for MR sequences, eliminating the reliance on proprietary hardware and programming languages specific to different scanner manufacturers. [1]

The advantages of having an open-source sequence design platform are mainly academic validation, accessibility, and collaboration. The core of the Pulseq platform is an open file format for MR sequences. An open file format ensures that the specifications and details of the sequence design are openly available and can be easily accessed and shared by researchers, scientists, and clinicians. This

openness brings several benefits to academic research in the field of MRI. Researchers can accurately reproduce and validate previously published sequences, which is crucial for verifying and building upon existing knowledge. Reproducibility enhances scientific validation in MRI research and acknowledges the credibility of the research. This also enables researchers to compare results directly without reverse engineering the sequence.

The Pulseq platform makes it possible to design, analyze, and export sequences using a custom Pulseq toolbox in MATLAB. Pulseq also makes it possible to execute these sequences on supported scanners. [1] This is one of the key advantages of Pulseq because traditionally, MRI scanners require sequences to be written in manufacturer-specific programming languages and stored in proprietary file formats. This limitation can hinder collaboration, research reproducibility, and the ability to execute sequences on different scanners. Using Pulseq, researchers can export their designed sequences in the Pulseq file format and execute them on supported scanners.

The developers also promote the use of Pulseq for educational purposes. [1]. The MATLAB toolbox is compatible with Python, and an external graphical interface has also been made to support the framework. [2] [3]

However, Pulseq is still under development, and full scanner integration regardless of manufacturer or model is not yet complete. Therefore, hardware-specific installation manuals are needed to utilize Pulseq.

## 1.1 Objective

This project primarily aims to implement and test the Pulseq sequence design platform by modeling and modifying a clinical Turbo Spin Echo (TSE) sequence for prostate examination using the Pulseq toolbox in MATLAB. The objective of the modification is to undersample k-space data acquisition in order to reduce the overall scan time for prostate cancer diagnosis. [4] This project focuses on modeling and modifying the clinical TSE sequence. Various variants of the TSE sequence will be executed on a Siemens Aera AG clinical MRI scanner with a field strength of 1.5 Tesla. The outcomes of these experiments will provide valuable insights into the effectiveness and feasibility of the Pulseq sequence design platform for rapid testing and prototyping purposes.

## 2 Theory

This section provides an overview of the clinical Turbo Spin Echo (TSE) sequence modeling and the Pulseseq platform.

### 2.1 Clinical TSE sequence modeling

A TSE sequence is formed by a series of RF pulses and spin echoes applied in a specific pattern. The sequence begins with an initial RF pulse that tips the magnetization vector by 90 degrees, after which the spins dephase. This is followed by a series of 180-degree refocusing RF pulses, which reverse the dephasing effects caused by inhomogeneities in the magnetic field. The initial RF pulse combined with the 180-degree refocusing pulses form a spin echo, and the series of acquired spin echoes in rapid succession is called an "echo train". The refocusing pulses are interleaved with gradient fields, which encode spatial information.

The TSE sequence for prostate examination, which is the focus of this project, is modeled in MATLAB using the Pulseseq toolbox with various sequence parameters. The resulting sequences are then executed on the scanner for further evaluation.

By varying parameters such as repetition time (TR), echo time (TE), flip angle, and the number of echoes per echo train, the modeled TSE sequence can be optimized to achieve the desired contrast and resolution. The TSE sequence used in this project is derived from an established clinical sequence specifically designed for prostate (cancer) imaging [4]. The specific parameters of this sequence are detailed in Section 3 and can be found in Table 1.

A TSE sequence forms an image based on multiple T2-contrasts since multiple TEs are used. [5] The TE in a TSE sequence is an "effective TE" since the TE of the echo nearest to  $k_0$  in k-space in an echo train accounts for the highest contrast of the reconstructed image. [6] Further review of k-space and data acquisition in this report is addressed in section 2.2.

Parameter	Value
FoV	180x180 mm
Slice thickness	3 mm
TR	8990 ms
TE	112 ms
Echo spacing	11.2 ms
Flip angle	150 deg
Base resolution	320
Bandwidth	200 Hz/Px
Echo trains per slice	26
Echoes per train	25

Table 1: Clinical TSE prostate sequence parameters

## 2.2 k-space and data acquisition

Different types of sequences determine the path through k-space and how MRI data is collected. k-space is a mathematical representation of the spatial frequency domain, often depicted as a grid or matrix. In k-space, each point represents a unique frequency component of the image. By acquiring data in k-space, the MRI scanner captures the frequency content of the image, which is then transformed and reconstructed to generate the final image.

Various methods exist for traversing through k-space, and this can be effectively visualized by plotting gradients and their corresponding k-space trajectory for a single echo train and multiple echoes in MATLAB. Figure 1 illustrates such visualization. The script utilized to generate the gradients and RF pulses in Figure 1 serves as the basis for generating the TSE sequences discussed in this report. It is worth noting that the x and y-values of the plotted k-spaces throughout this report are dimensionless and do not possess specific labels

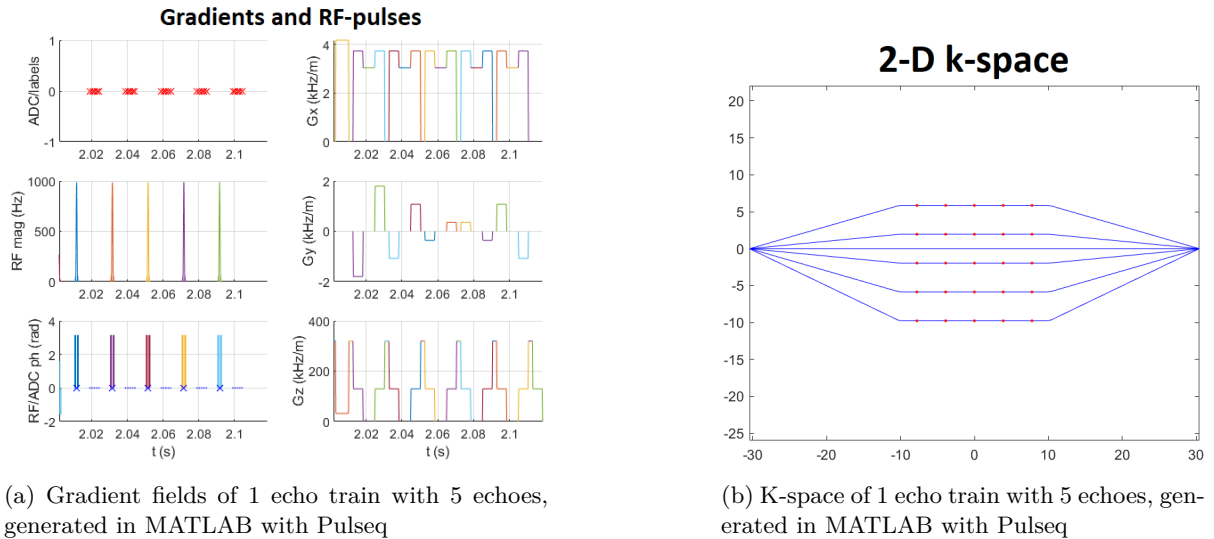


Figure 1: Gradients and k-space of 1 echo train with 5 echoes

In Figure 1a the gradients and the excitations of the 5 180-degree pulses can be seen. The resulting k-space can be seen in Figure 1b. This will give a rough idea of how a TSE sequence is formed with the script used in this report. The k-space in this example has 5x5 data points. When adding a second echo train, extra lines are drawn through k-space with an offset to the first echo train.

### 2.2.1 k-space undersampling

One objective of this project is to model a TSE sequence with undersampled k-space in support of faster scan times for prostate cancer diagnostics. The undersampling of k-space can be achieved by omitting one or more echo trains from the sequence. In Section 3 the details of the omitted echo trains will be explained.

## 2.3 The Pulseq framework

This section gives an overview of the two components of the Pulseq framework: the Pulseq toolbox for MATLAB and a Pulseq-modified scanner interpreter.

The Pulseq toolbox, integrated with MATLAB, enables the generation and analysis of sequences. Researchers can utilize this toolbox to create sequences and export them as open file format (.seq) files, which are compatible with the modified scanner interpreter. During and after the generation of the sequence, the generated sequences can be analyzed and visualized. The sequence (.seq) files can be uploaded to the scanners and executed as regular MR sequences.

On the scanner side, the scanner interpreter is modified to read out the Pulseq file format. It is a modified hardware-specific dynamic link library (.dll) responsible for converting the (.seq) files generated by the Pulseq toolbox into the appropriate sequence file format specific to each manufacturer. This .dll ensures compatibility between the Pulseq file format and the scanner. By modifying the scanner interpreter, the scanner can read and interpret (.seq) files, enabling the execution of the designed sequences on the scanner. A general overview of the data flow can be seen in Figure 2.

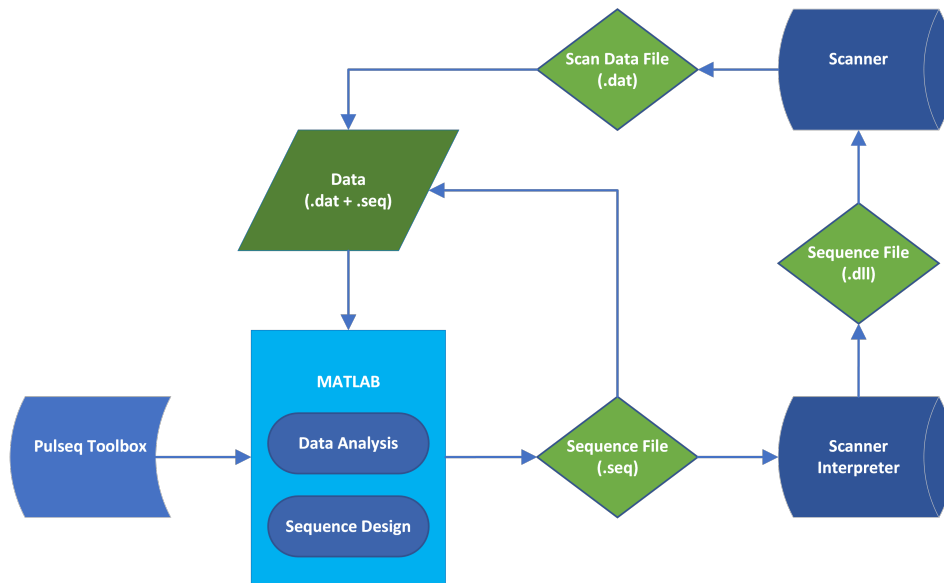


Figure 2: General overview of dataflow in the Pulseq framework

The detailed installation of the Pulseq framework in MATLAB and on the interpreter will be discussed in the appendix [A.1](#).



### 3 Method

Different variants of the clinical TSE sequence and its parameters can be modeled in MATLAB (v2021a, The MathWorks, Natick, MA) with the support of the Pulseseq toolbox. The developers of Pulseseq have made basic MATLAB scripts for various types of MRI sequences. A script already exists for the generation (and analysis, see Section 3.3) of a basic TSE sequence file [7]. This script can be used as a base to model the clinical TSE sequence. Among other simulation options, the k-space trajectory can be plotted in MATLAB using the Pulseseq toolbox and can be seen in Figure 3.

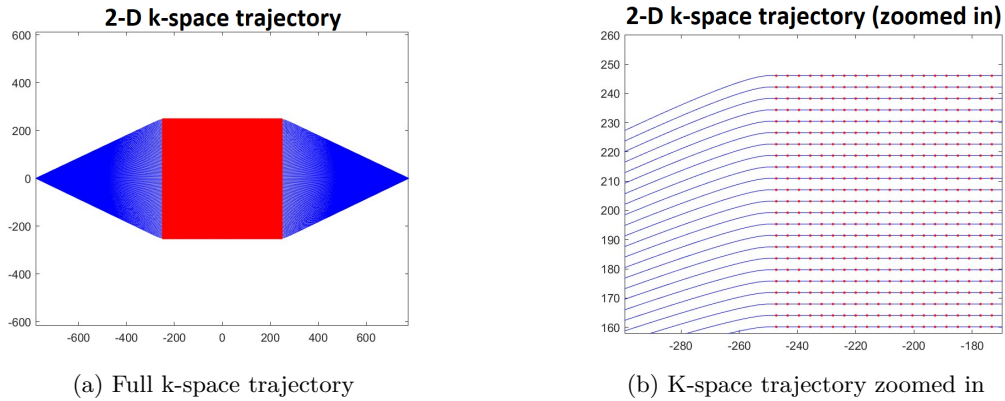


Figure 3: K-spaces plotted of the basic TSE sequence file

In Figure 3b, the zoomed-in k-space trajectory can be seen. The data acquisition is shown as red dots overlaid on the blue trajectory. In this standard sequence, there are 128x128 k-space data points. Using a Pulseseq function called "testReport", important system parameters such as maximum gradients, maximum slew rates, and many other important parameters can be outputted, as can be seen in Appendix A.2. This can be helpful to quickly verify the parameters of the generated sequence.

The desired parameters of the sequence were discussed in Section 2 and are shown in Table 1. Most of these parameters can be adjusted in the base file and can be executed to generate and simulate a new sequence file. k-space visualization, such as in Figure 3, can be analyzed to verify the k-space trajectory and the sequence parameters that were changed. The undersampling of the k-space will be done by omitting echo trains from the modeled sequences. Variants of sequences with the following echo trains removed will be made:

- Echo train no. 1
- Echo train no. 5
- Echo train no 2, 5, 13

These different variants are made to observe the effect of the removal of echo trains in a random singular order (omitting echo train no. 1 and 5) and the effect of removing multiple echo trains (echo train no. 2, 5 and 13). The different variants with different parameters and different modifications can be found in Figure 4.

### 3.1 Sequence variants

This section will describe the variants of the TSE sequence that were created. The variants of the sequence will be modeled step-by-step starting with the basic TSE sequence. The variants of the modeled sequence are shown in Figure 4.

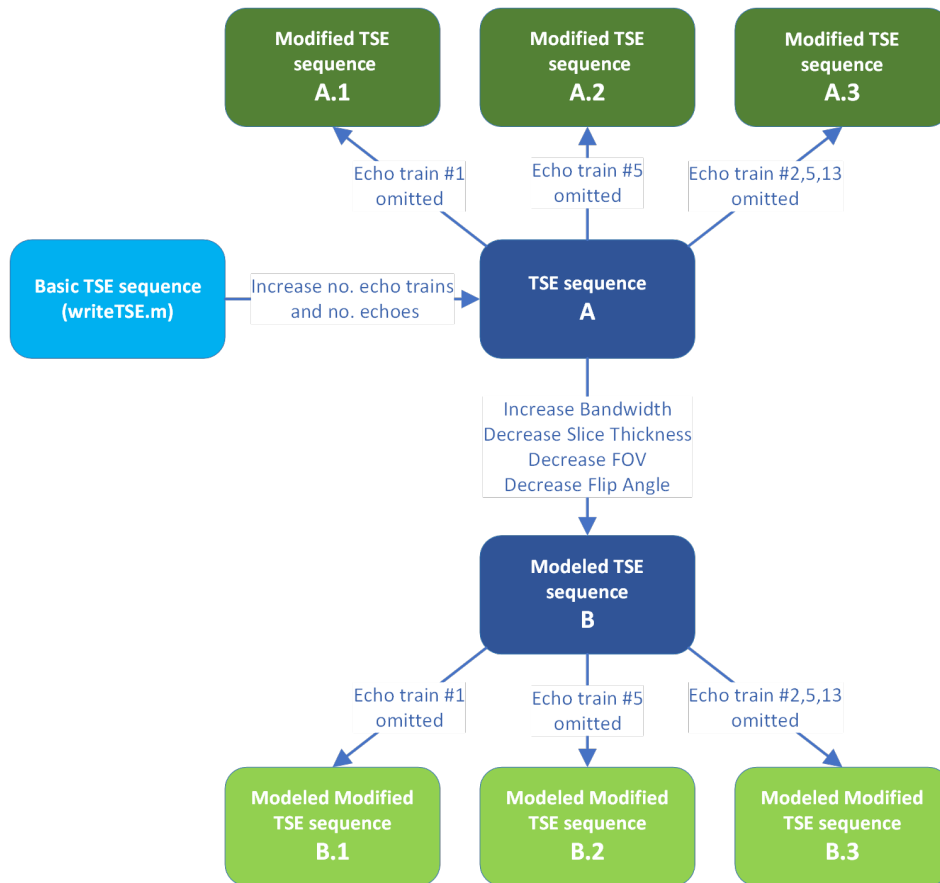


Figure 4: Variants of the TSE sequence originating from the basic TSE sequence

The variants referenced in this report will refer to the variants in Figure 4. The sequence variants are modeled by changing parameters in the basic TSE script and can be found in Table 2. The variants in this report will be referenced by (A,B,A.1 etc.).

Parameter	Basic TSE sequence	TSE sequence (A)	Modified TSE (A.1-3)	Modified Modeled TSE (B.1-3)
FOV	256 $mm^2$	180mm $mm^2$	256 mm $mm^2$	180 $mm^2$
Slice thickness	5 mm	3 mm	5 mm	3 mm
TE	60 ms	168 ms	260 ms	260 ms
TR	2000 ms	2000 ms	2000 ms	2000 ms
Echo spacing	12 ms	12 ms	20 ms	20 ms
Flip angle	180 deg	150 deg	180 deg	150 deg
kx/ky	128/128	650/650	650/650	650/650
Bandwidth	153 Hz	200 Hz	153 Hz	200 Hz
Echo trains	8	26	23-25	23-25
Echoes/train	16	25	25	25

Table 2: Parameters of the modeled and modified TSE sequences

### 3.1.1 TSE sequence (A)

The first step in modeling the TSE sequence is to increase the number of echo trains and echoes as specified in Table 2. The number of echo trains is dependent on the size of the ky size of the k-space and the number of echoes specified in the script, as can be seen in Equation 1. To achieve the desired 26 echo trains, the number of k-space (ky,kx) data points needs to be increased to 650, and the number of echoes to 25. The number of echo trains in the sequence is determined solely by the size of ky, while kx is set to 650 to ensure a square voxel size.

$$no. EchoTrains = \frac{ky}{no. Echoes} \quad (1)$$

The effective TE is determined by the standard Pulseq TSE sequence file to be the middle echo train in the k-space, which can be seen in Equation 2.

$$TE_{effective} = \frac{no. Echoes}{2} * EchoSpacing \quad (2)$$

Siemens has a different notation of the effective TE in the clinical TSE sequence (Table 1), in which the effective TE is centered around the 10th echo instead of the middle echo (echo no. 13). The Pulseq sequences generated in this report will use the Pulseq effective TE notation. For the sake of efficiency in sequence prototyping and testing, a TR value of 2000 ms was employed for all sequences in this report. The reduction in TR has a negligible impact on signal strength.

### 3.1.2 Modified TSE sequence (A.1-3)

The next step is to modify the TSE sequence (A). The purpose of this project is to showcase the versatility provided by the Pulseseq platform in the rapid prototyping and design of sequences, particularly through modifications to the clinical TSE sequence. These modifications involve undersampling k-space by selectively omitting certain echo trains within the TSE sequence. The omission of echo trains will result in a decrease in total scan time.

The omission of echo trains can be achieved by making specific alterations to the phase encoding section of the script employed. The MATLAB code in Appendix A.4 showcases the changes made to the original script. Compared to the code of the basic TSE sequence, additional lines of code have been included to allow for the omission of certain echo trains and optionally omitting specific echoes within the generated sequence. In Figure 5, the k-space of the subvariants (A.1-3) of the TSE sequence (A) is shown. The figures in Figure 5 are zoomed in to display the omitted lines.

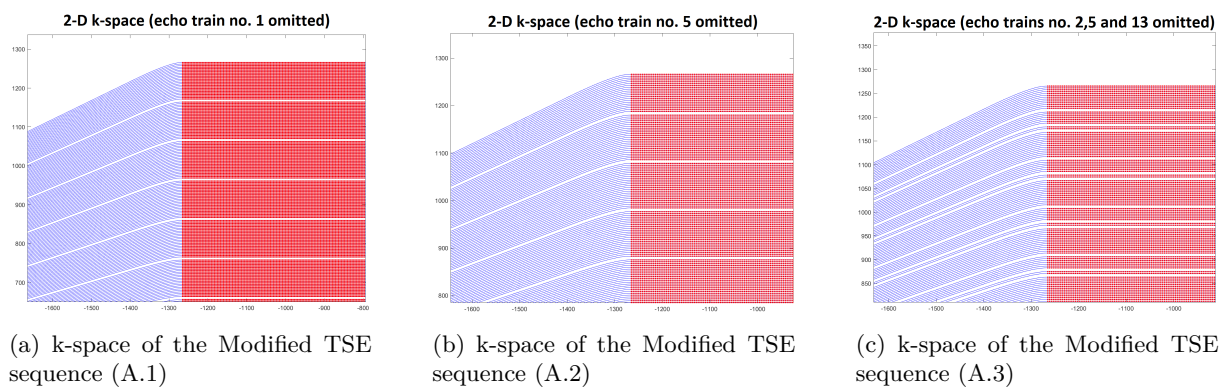


Figure 5: k-spaces plotted of the Modified TSE sequences

It is important to note that omitting echo trains can lead to a decrease in the total scan time, as it is dependent on the number of echo trains. However, omitting individual echoes within a train does not affect the scan time, as it is independent of the number of echoes within each train. The visualizations of the generated sequences in this report are not equivalent to the final sequence file format used in practice. The visualizations serve as a representation for understanding and analysis purposes, but they do not directly correspond to the actual sequence file format employed on the scanner. The sequence files (.seq) generated by the Pulseseq framework need to be uploaded to the scanner in order to be executed. The process of uploading external sequences to the scanner is detailed in the appendix of this report.

### 3.1.3 Modeled TSE (B)

Multiple parameters need to be adjusted to form the Modeled TSE sequence (B). The parameters that need to be changed are:

- Bandwidth: 156 Hz→200 Hz
- Flip Angle: 180 Deg→150 Deg
- FOV: 256 mm<sup>2</sup>→180 mm<sup>2</sup>
- Slice Thickness: 5mm→3mm

Some parameters cannot be directly influenced by changing parameters in the code, so they have to be changed through other parameters. For example, the bandwidth of a k-space voxel is not specified in the current script. The bandwidth can be indirectly changed by changing the sampling time of a data point in k-space, as can be seen in Equation 3. Pulseq employs conservative safety limits (more strict than Siemens) for maximum gradient strengths and maximum slew rates of the gradient coils. This means that some combinations of parameter values are not allowed, even though they might be possible through Siemens' own sequences.

$$Bandwidth = \frac{1}{SampleTime} \quad (3)$$

For example, the desired echo spacing time of 11.2 ms<sup>1</sup> in combination with the other parameters is not allowed by Pulseq safety limits. Therefore, this parameter was changed to 20 ms to adhere to the safety limits implemented by Pulseq. This also means that the effective TE of the sequence will become larger. The k-space analysis of this sequence shows a similar k-space trajectory to Figure 3.

### 3.1.4 Modeled Modified TSE sequence (B.1-3)

In the next step of creating the Modified Modeled TSE sequences (B.1-3), multiple echo trains are removed, a process described in Section 3.1.2. The same script modifications can be applied to eliminate the echo trains in these sequence variants as well.

---

<sup>1</sup>See Table 1)

### 3.2 Executing the generated sequences

Once the sequence variants have been successfully uploaded to the scanner, they can be executed to perform imaging. For the purpose of comparison and evaluation, an MRI phantom (Siemens AG, Erlangen, Germany) is used as the imaging object (see Figure 6). The MRI phantom serves as a standardized reference object and is used to qualitatively assess differences in image quality between the executed sequences. The execution of the sequences on the scanner will be carried out using a 20-channel head RF coil (Siemens AG, Erlangen, Germany).



Figure 6: The Siemens MRI phantom being used

### 3.3 Image Reconstruction

Two files are required to reconstruct MRI images in MATLAB: the sequence (.seq) file<sup>2</sup> and the binary Siemens data (.dat) file. The image reconstruction can be performed using Pulseseq's Fast Fourier Transform image reconstruction script [8], which relies on the open-source "mapVBVD.m" data parsing function [9].

---

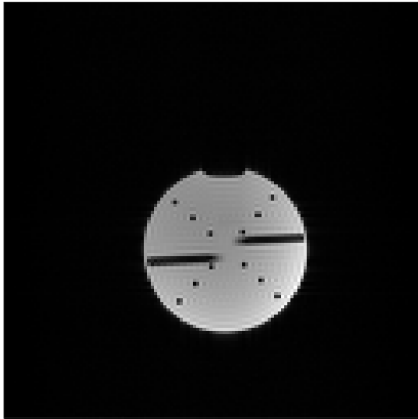
<sup>2</sup>See Figure 2

## 4 Results

The initial sequence that was executed is the Basic TSE sequence, without any modifications. Additionally, the results of the various sequences listed in Table 2 will be presented. The head RF coil utilizes 20 channels, so 20 separate signals are to be expected when reconstructing images. Throughout this section, the sequences will be referenced based on Figure 4.

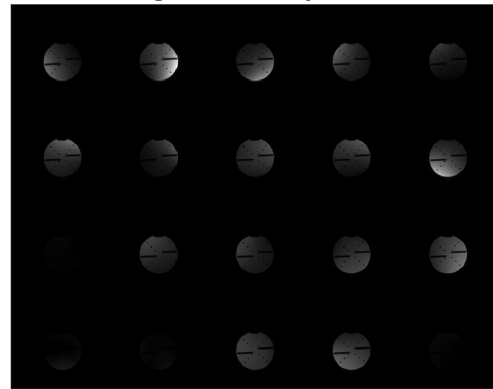
### 4.1 Basic TSE sequence

**Cumulative reconstructed image  
from multiple coils**



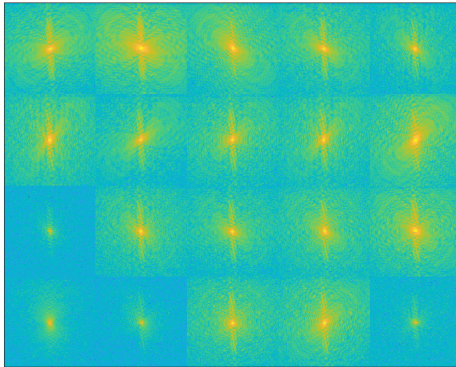
(a) Reconstruction of the image using the combined signal of all the receiver coils

**Image formed per coil**



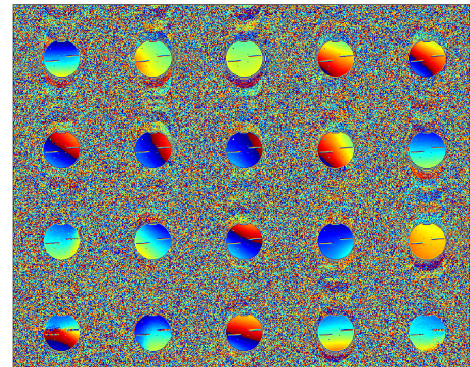
(b) Separate reconstruction of the signal of the coils

**K-space data per coil**



(c) K-space plotted of the separate coils

**Phase angle per coil**



(d) Phase difference of the separate coil image

Figure 7: Image reconstruction of the basic TSE sequence

Figure 7 presents the reconstructed images and plotted k-space for the basic TSE sequence. It includes visualizations of the phase difference and images per receiver coil for this variant of the sequence. The visualizations of the k-space and phase difference are limited to this sequence and the Modified TSE sequence (A.3) in this report. For the remaining visualized sequences only the fully reconstructed image and the separate coil data will be plotted, as can be seen in Figure 7a and 7b.

## 4.2 Clinical TSE sequence

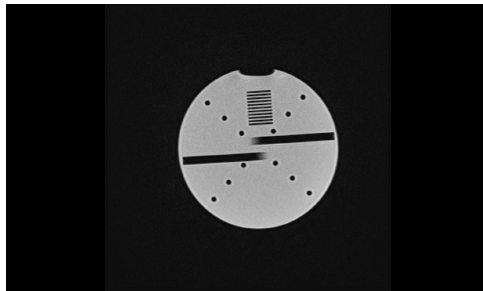


Figure 8: The image generated by the clinical TSE (Siemens)

Figure 8 presents the image generated by the clinical TSE sequence, which serves as the baseline for comparison with the Pulseseq-generated sequences. The specific parameters employed for this sequence can be found in Table 1. This image provides a reference point for evaluating and assessing the quality of the Pulseseq-generated sequences in relation to the clinical TSE sequence.

## 4.3 TSE sequence (A)

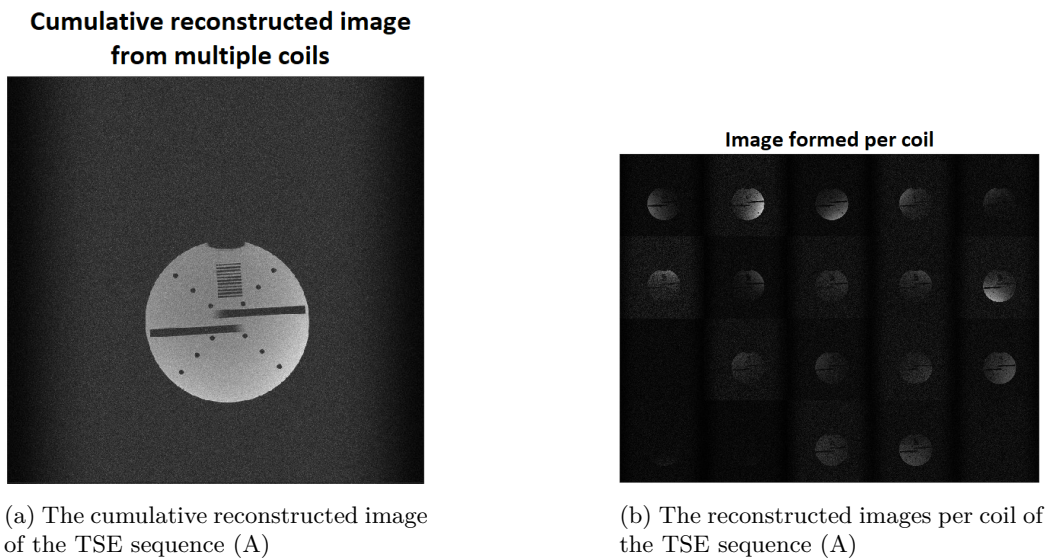


Figure 9: The TSE sequence (A) with 26 echo trains and 25 echoes



## 4.4 Modified TSE sequences

### 4.4.1 Modified TSE sequence (A.1)

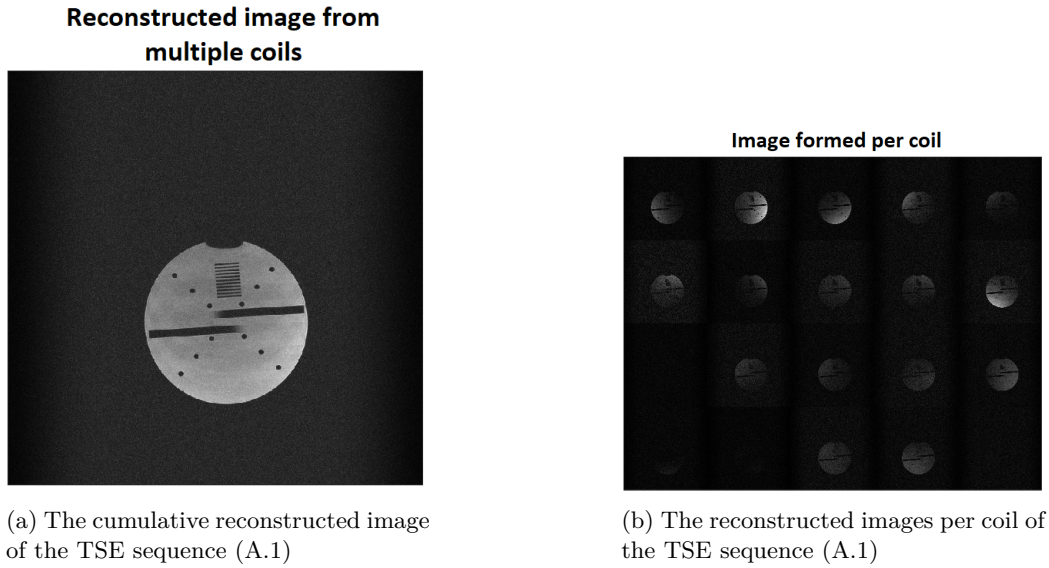


Figure 10: The TSE sequence (A.1) with 26-1 echo trains and 25 echoes

### 4.4.2 Modified TSE sequence (A.2)

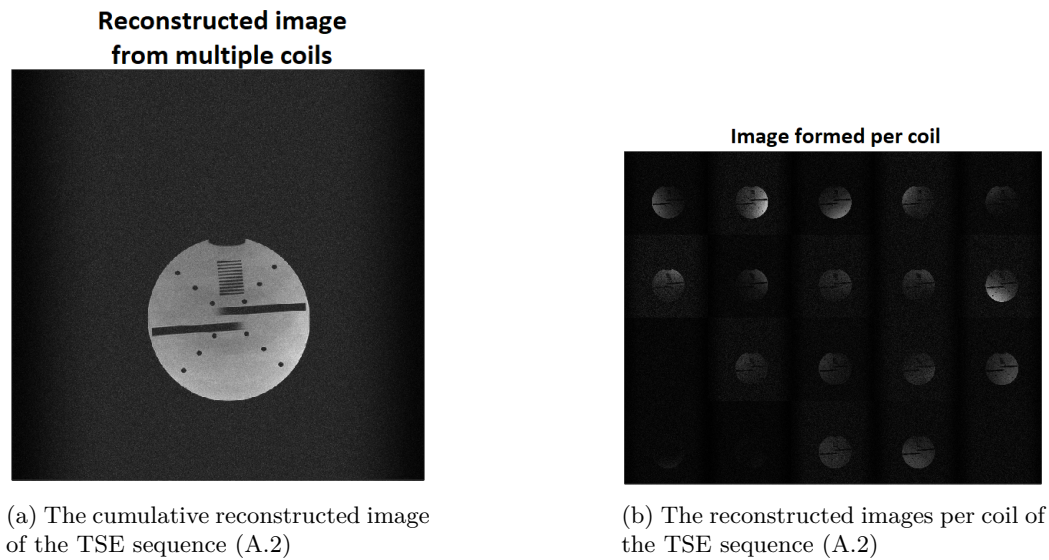
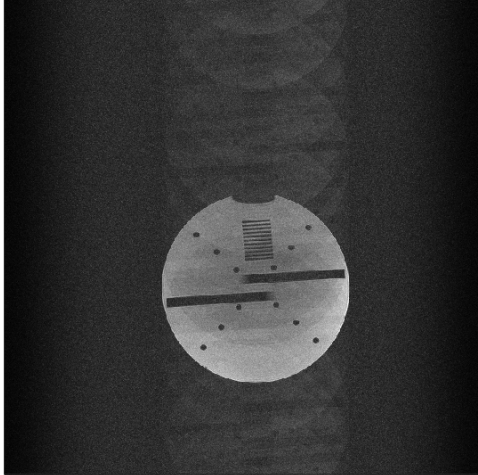


Figure 11: The TSE sequence (A.2) with 26-1 echo trains and 25 echoes

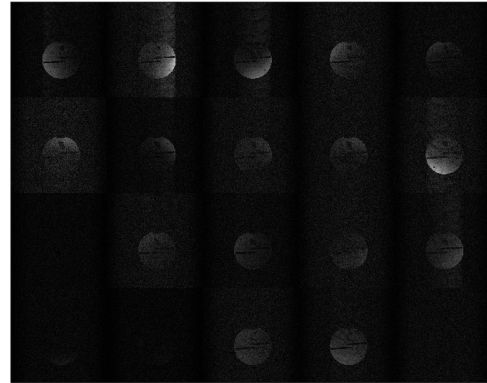
#### 4.4.3 Modified TSE sequence (A.3)

##### Reconstructed image from multiple coils



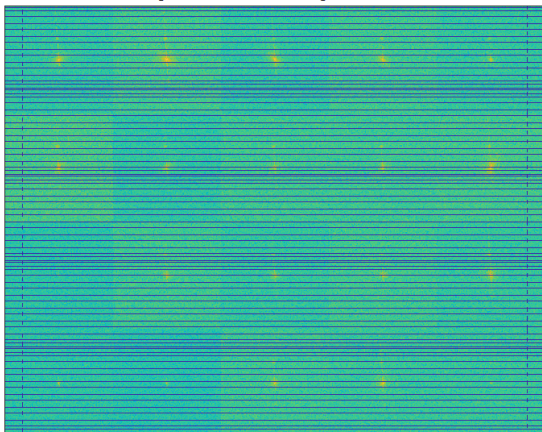
(a) Reconstruction of the image using the combined signal of all the receiver coils

##### Image formed per coil



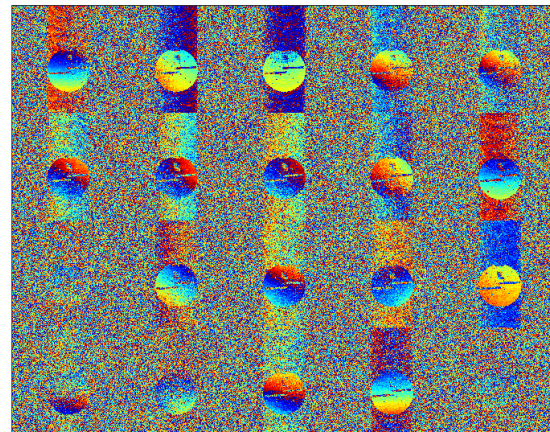
(b) Separate reconstruction of the signal of the coils

##### K-space data per coil



(c) K-space plotted of the separate coils

##### Phase angle per coil



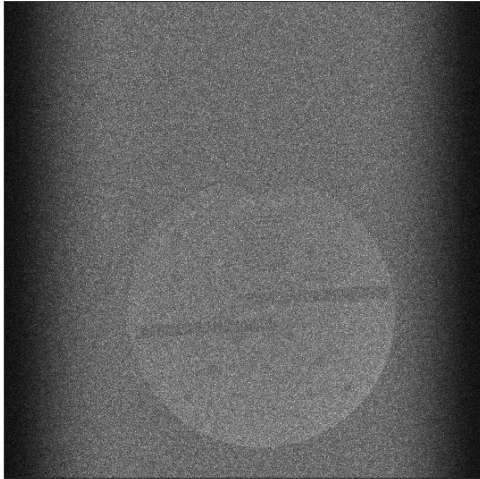
(d) Phase difference of the separate coil image

Figure 12: The Modified TSE sequence (A.3) with 26-3 echo trains and 25 echoes

The missing k-space data, resulting from the missing echo trains can be clearly seen in Figure 12c. The subsequent reconstruction shows ghosting artifacts which will be further discussed in Sections 5 and 6.

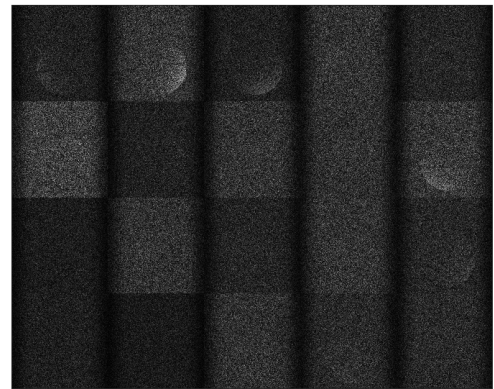
#### 4.5 Modeled TSE sequence (B)

##### Reconstructed image from multiple coils



(a) Reconstruction of the image using the combined signal of all the receiver coils

##### Image formed per coil



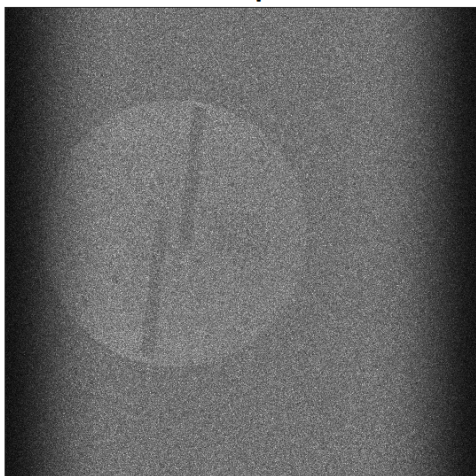
(b) Separate reconstructed images of the receiver coils

Figure 13: Reconstructed images of the modeled TSE sequence

#### 4.6 Modeled Modified TSE sequences

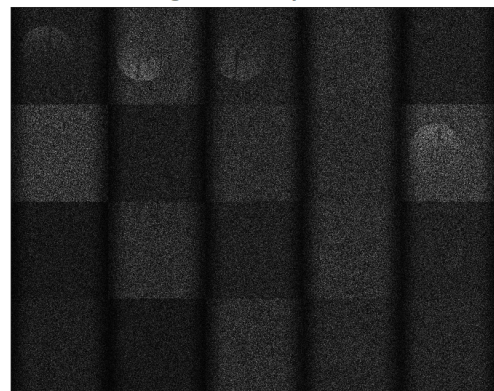
##### 4.6.1 Modeled Modified TSE sequence (B.1)

##### Reconstructed image from multiple coils



(a) The cumulative reconstructed image of the TSE sequence (B.1)

##### Image formed per coil



(b) The reconstructed images per coil of the TSE sequence (B.1)

Figure 14: The Modified TSE sequence (B1) with 26-1 echo trains and 25 echoes

#### 4.6.2 Modeled Modified TSE sequence (B.2)

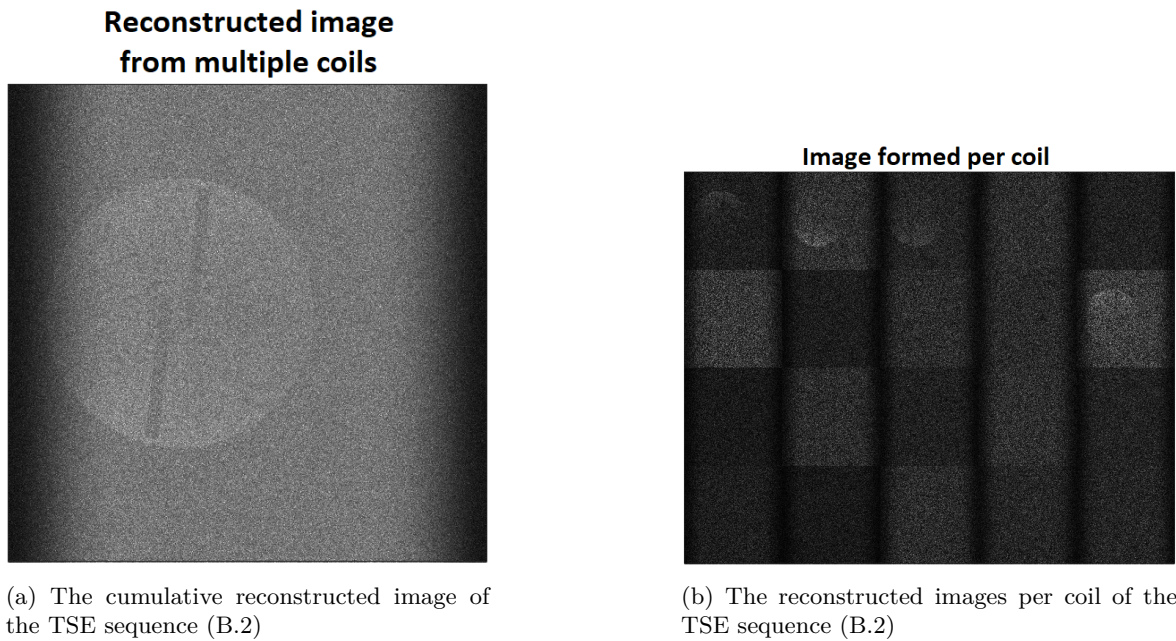


Figure 15: The Modeled Modified TSE sequence (B.2) with 26-1 echo trains and 25 echoes

#### 4.6.3 Modeled Modified TSE sequence (B.3)

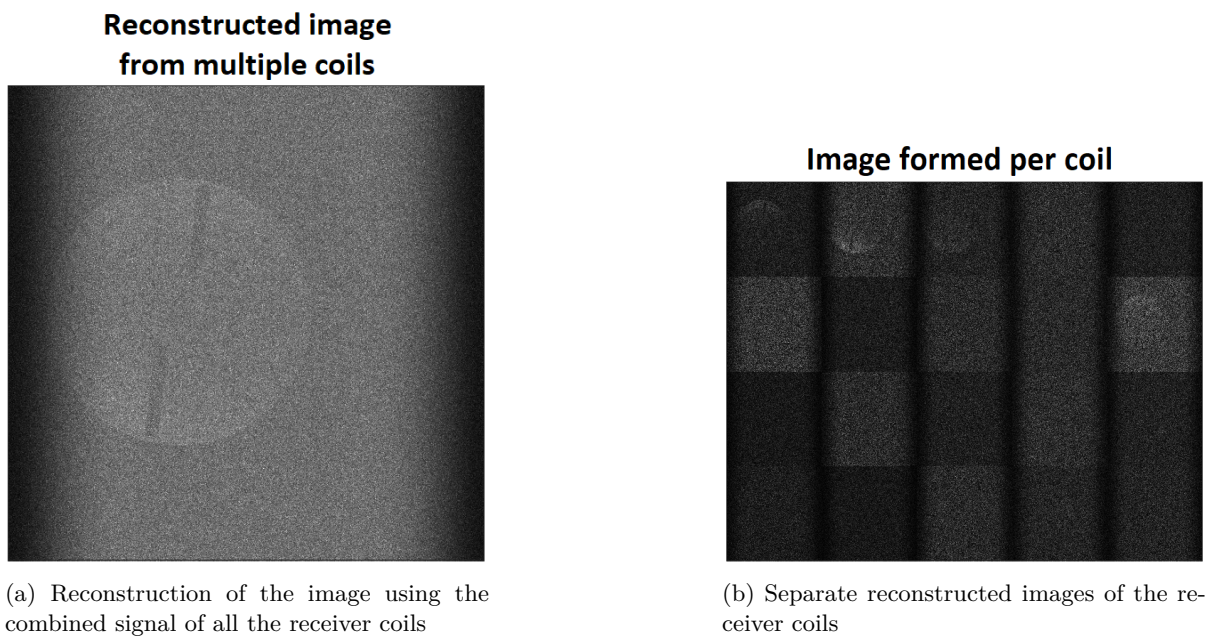


Figure 16: The Modified Modeled TSE sequence (B.3) with 26-3 echo trains and 25 echoes

## 5 Conclusion

This report has focused on the implementation of the Pulseq platform on a clinical MRI scanner through the modeling and modification (undersampling) of a clinical TSE sequence, followed by the execution of the generated sequences. The implementation of the Pulseq platform is successful, multiple TSE sequences were successfully modeled using MATLAB and executed on the scanner. However, certain combinations of parameters in the clinical TSE sequence could not be modeled.

The image quality of the Modeled TSE sequences (Figure 4 (B)) was significantly lower compared to TSE sequence (A). Consequently, no conclusive findings can be drawn from the Modified Modeled TSE sequence (B.1-3) due to the low image quality. Omitting a single echo train (A.1-2) slightly reduces image quality compared to a non-modified TSE sequence (A). The impact of undersampling by omitting three echo trains in the Modified TSE sequence (A.3) is evident in the form of ghosting artifacts.

## 6 Discussions

The discussions of this report will cover both the Pulseq platform and the TSE sequences created using the Pulseq platform.

### 6.1 Installation

Once the installation was completed, the Pulseq platform demonstrated smooth operation on the scanner side without encountering any difficulties. The full step-by-step installation can be found in Appendix A.1. Several internal toolboxes were needed to complement the Pulseq toolbox. The required toolboxes are laid out in the manual in Appendix A.1.

### 6.2 Effective echo time

During the modeling of the TSE sequence, it was observed that certain parameter combinations did not adhere to the safety limits set by Pulseq. As a result, the parameters were adjusted to the closest permissible alternative values. Due to these necessary adjustments made to the parameters of the originally proposed sequence to comply with Pulseq safety standards, it is not possible to model the clinical TSE sequence to its original parameters.

However, it is worth noting that the TSE sequence that was initially planned to be modeled had parameters that were at the limits of the scanner's capabilities. The parameters outlined in Table 1, particularly the echo spacing, had to be adjusted to a value of 20 ms instead of the intended 11.2 ms. The adjustments made to the echo spacing in the sequence also impact the effective TE, which, in turn, can affect the signal strength of the imaged object. Differences in echo spacing can influence the T2-weighting and signal decay characteristics of the acquired data. As mentioned in Section 2, it is important to note that the definition of the effective TE can differ between the Pulseq standards and the Siemens standards. The effective TE represents the TE of the echo train nearest to  $k_0$  in k-space, which accounts for the highest contrast in the reconstructed image. It is worth considering this distinction when comparing and interpreting results obtained using the Pulseq platform and the

Siemens standards. This would partially explain the degraded image quality of the Modeled TSE sequence (B) compared to the regular TSE sequence (A).

### 6.3 k-space dimensions and resolution

The dimensions of the k-space in the Pulseseq-generated TSE sequence are determined by the number of echo trains and the number of echoes. During image reconstruction, the same number of data points as acquired are utilized. In contrast, the clinical TSE sequence employs the same number of echo trains and echoes, but its reconstruction does not involve the same number of data points. The reconstruction of the clinical TSE sequence differs from the Pulseseq-generated TSE sequence in terms of the number of data points utilized. While the Pulseseq sequence would logically have 650x650 data points with 26 echo trains and 25 echoes, the clinical TSE sequence uses a reduced number of data points, specifically 320x320. This indicates that the clinical TSE sequence applies an averaging technique, resulting in an increase in image quality compared to the Pulseseq-generated sequence. The image quality of the clinical TSE sequence cannot be achieved by the Pulseseq-generated sequences.

### 6.4 Modifications to the sequences

The removal of echo trains in the TSE sequence (A) had an impact on the presence of ghosting artifacts (Figure 12a). When a single echo train was omitted, no ghosting artifacts were observed, but a slight degradation of image quality can be seen. (A.1-2) However, when three echo trains were omitted, ghosting artifacts became apparent in the reconstructed images. This implies that a certain level of undersampling is needed to observe the formation of ghosting artifacts. The k-space in Figure 12a of the Modified TSE sequence (A.3) shows the omitted lines of k-space compared to the k-space of the basic TSE sequence (Figure 7a). Due to the low image quality, the same type of ghosting effects on the Modeled Modified TSE sequence (B.3) cannot be seen.

### 6.5 Improvements to Pulseseq sequence design

Further enhancements to the TSE sequence modeling using Pulseseq can be explored by experimenting with different effective TE values and various approaches to k-space data reconstruction. By adjusting the effective TE, it is possible to optimize the T2 contrast and signal strength of the generated images. Additionally, exploring alternative methods of data reconstruction from k-space can potentially improve image quality and reduce artifacts in the Pulseseq-generated sequences.

### 6.6 Future works

The dataflow outlined in Figure 2 is not automatic. Sequences need to be manually uploaded and data needs to be manually extracted from the scanner. The manual extraction of data, in particular, is prone to error since the originally uploaded sequence needs to be manually linked to the generated data. A way to prevent these oversights is to implement an automatic data send-and-receive platform, which can automatically send and receive data over the Internet. [10]

## References

- [1] Layton KJ, Kroboth S, Jia F, Littin S, Yu H, Leupold J, et al. Pulseq: A rapid and hardware-independent pulse sequence prototyping framework. *Magn Reson Med*. 2017 Apr;77(4):1544-52. doi:10.1002/mrm.26235.
- [2] Ravi KS, Geethanath S, Vaughan JT. PyPulseq: A Python Package for MRI Pulse Sequence Design. *Journal of Open Source Software*. 2019 Oct;4(42):1725. doi:10.21105/joss.01725.
- [3] Ravi KS, Potdar S, Poojar P, Reddy AK, Kroboth S, Nielsen JF, et al. Pulseq-Graphical Programming Interface: Open source visual environment for prototyping pulse sequences and integrated magnetic resonance imaging algorithm development. *Magn Reson Imaging*. 2018 Oct;52:9-15. doi:10.1016/j.mri.2018.03.008.
- [4] Mir N, Damen S, Wolterink J, Simonis F, Fütterer J. Shortening T2-weighted TSE for prostate MRI by reducing the number of echo-trains and using CS reconstruction. PHD thesis. 2023.
- [5] Paschal CB, Morris HD. K-Space in the Clinic. *Journal of magnetic resonance imaging : JMRI*. 2004 Feb;19(2):145-59. doi:10.1002/jmri.10451.
- [6] Mugler JP. Optimized three-dimensional fast-spin-echo MRI. *J Magn Reson Imaging*. 2014 Apr;39(4):745-67. doi:10.1002/jmri.24542.
- [7] Pulseq. Pulseq; 2023. [Online; accessed 14. Jun. 2023]. Available from: <https://github.com/pulseq/pulseq/blob/master/matlab/demoSeq/writeTSE.m>.
- [8] pulseq. pulseq; 2023. [Online; accessed 2. Jul. 2023]. Available from: <https://github.com/pulseq/pulseq/blob/master/matlab/demoRecon/reconExample2DFFT.m>.
- [9] Cic-methods. FID-A; 2023. [Online; accessed 19. Jun. 2023]. Available from: <https://github.com/CIC-methods/FID-A/tree/master/inputOutput/mapVBVD>.
- [10] Hansen MS, Sørensen TS. Gadgetron: An open source framework for medical image reconstruction. *Magn Reson Med*. 2013 Jun;69(6):1768-76. doi:10.1002/mrm.24389.
- [11] Freiburg UMC. Open-source pulse sequences: Specifications PDF of the .seq file format; 2023. [Online; accessed 18. Jun. 2023]. Available from: <https://pulseq.github.io/specification.pdf>.

## A Appendix

### A.1 Manual

This manual can be used to install Pulseq on Matlab and on a clinical MRI scanner. The C2P manual

"PulSeq\_C2P\_Manual.pdf"

is needed and will be referenced for the scanner installation.

#### A.1.1 MATLAB Step 1

a) In Matlab, install the following toolboxes:

- Curve Fitting Toolbox
- Signal Processing Toolbox
- Image Processing Toolbox
- Symbolic Math Toolbox

b) Download the Pulseq-Matlab directory from

`{https://github.com/pulseq/pulseq/tree/master/matlab}`

c) In the subfolder of the downloaded directory, move the `+mr` folder to your active MATLAB folder.<sup>3</sup>

d) In the subfolder of the downloaded directory, use a `<demoSeq.m >` file to generate a sequence file (`.seq`)

e) Upload the sequence file to the scanner

---

<sup>3</sup>This folder acts like a toolbox



### A.1.2 Scanner

- a) Request permission to Pulseq files from C2P environment (software VE11E)
- b) Download pulseq.zip from the C2P environment
- c) In the IDEA VM:
  - LibBalance support needs to be **disabled** →check C2P Manual
  - Unzip the attached pulseq.zip file into the pertinent sequence directory:  
`C:/MIDEA/YOUR_IDEA_BASELINE/src/MrImaging/seq`
  - Copy/rename  
`pulseq/makefile_nX.tris -> pulseq/makefile.tris`
  - Make an empty file and upload to the folder  
`MeasSections.h -> MrServers/MeasSrv/MeasSections/MeasSections.h`
  - In the IDEA shell type “cs pulseq”
  - In the IDEA shell type “ms”
- d) The pulseq.dll and libpulseq.so are now created by the IDEA VM. These files will be uploaded to the scanner
- e) Login with Advanced User on the scanner
- f) Follow the steps in the C2P manual for the further installation on the scanner.
- g) Upload the Matlab-generated sequence (.seq) file to the scanner  
`<CustomerSeq>\pulseq\`
- h) Open the sequence on the scanner and run the sequence
- i) Copy raw data (.dat) file from TWIX to external drive

### A.1.3 MATLAB Step 2

- a) Download the open-source `<mapVBVD.m >` parsing function and move it to your active MATLAB folder.
- b) Copy the raw data file from the external drive to the active MATLAB folder.
- c) From the downloaded Pulseq-Matlab directory, under demoRecon open `<reconExample2DFFT.m >`
- d) Use the original sequence file `<testTSE.seq >` and rename to the `<rawdata.dat >` file  
`<rawdata.seq >`
- e) Add the `<testTSE.seq >` and the `<rawdata.dat >` file to the same folder and add the path of this folder in the `<reconExample2DFFT.m >` file
- f) Run the reconstruction in MATLAB

## A.2 Global simulated TSE values

The data below is outputted by the testReport function of Pulseq.

```
1 Timing check passed successfully
2 Number of blocks: 2862
3 Number of events:
4   RF:      702
5   Gx:     2052
6   Gy:     1350
7   Gz:     2160
8   ADC:     650
9   Delay:    0
10 Sequence duration: 81.000000s
11 TE: 0.260000s
12 TR: 3.000000s
13 Flip angle: 90.00°
14 Flip angle: 150.00°
15 Unique k-space positions (a.k.a. columns, rows, etc): 650
16 Unique k-space positions (a.k.a. columns, rows, etc): 650
17 Dimensions: 2
18   Spatial resolution: 0.28 mm
19   Spatial resolution: 0.28 mm
20 Repetitions/slices/contrasts: 1 range: [1 1]
21   422500 k-space position(s) repeated 1 times
22 Cartesian encoding trajectory detected
23 Block timing check passed successfully
24 Max. Gradient: 777546 Hz/m == 18.26 mT/m
25 Max. Gradient: 298933 Hz/m == 7.02 mT/m
26 Max. Gradient: 533333 Hz/m == 12.53 mT/m
27 Max. Slew Rate: 3.11018e+09 Hz/m/s == 73.05 T/m/s
28 Max. Slew Rate: 1.19573e+09 Hz/m/s == 28.08 T/m/s
29 Max. Slew Rate: 2.13333e+09 Hz/m/s == 50.11 T/m/s
30 Max. Absolute Gradient: 779411 Hz/m == 18.31 mT/m
31 Max. Absolute Slew Rate: 3.65393e+09 Hz/m/s == 85.82 T/m/s
```

### A.3 The Pulseseq sequence file format

As can be seen in Figure 2, the (.seq) sequence file can be generated in MATLAB using the Pulseseq toolbox. The (.seq) file format used by the Pulseseq framework is an open file format that is not tied to any specific MRI scanner manufacturer.

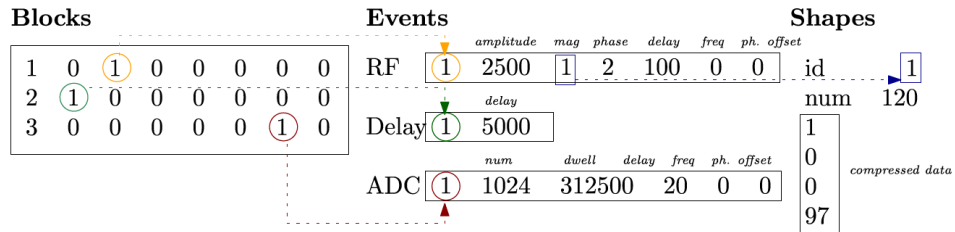


Figure 17: The hierarchical structure of the (.seq) file format

The hierarchical structure of the file format can be divided into three different levels. The layout can be seen in Figure 17. The Pulseseq toolbox in MATLAB builds the designed sequence by generating Blocks that consist of Events, which in turn consist of Shapes. [11]

- **Blocks:** At the top level, the sequence is defined by a series of blocks, where each block represents one or more events that are executed simultaneously.
- **Events:** The definition of events in a sequence depends on their type, such as gradients or ADC (analog-to-digital converter). These basic shapes serve as building blocks for constructing the waveform of an event.
- **Shapes:** A shape is a compressed list of samples. The uncompressed samples can describe, for example, the RF pulse shape or an arbitrary gradient shape.

The modified scanner interpreter can read out the open (.seq) file format and convert it to a manufacturer-specific file format.

## A.4 Code Modifications

```
1 %% Phase encoding
2 % To move the  $k$ -space trajectory away from 0 prior to the readout a
3 % prephasing gradient must be used. Furthermore rephasing of the slice
4 % select gradient is required.
5
6 nex=floor(Ny/necho);
7
8 pe_steps=(1:(necho*nex))-0.5*necho*nex-1;
9 if 0==mod(necho,2)
10     pe_steps=circshift(pe_steps,[0,-round(nex/2)]); % for odd number of
11     echoes we have to apply a shift to avoid a contrast jump at k=0
12 end
13 PEorder=reshape(pe_steps,[nex,necho])';
14
15 selectedtrains=[1,3,4,6:12,14:26];
16 selectedechoes=[1:25];
17 PEorder=PEorder(selectedechoes,selectedtrains);
18 nex=length(selectedtrains);
19 necho=length(selectedechoes);
20 phaseAreas = PEorder*deltak;
```