Generating coil sensitivity maps from MR-STAT data using ESPIRiT

Bachelor Thesis

Author: Laurie de Vries
Student-number: 6480950

Supervisors:
Oscar van der Heide, Ph. D.
& Dr. Alessandro Sbrizzi

Utrecht University

UMC Utrecht

June 17^{th}, 2022
Department of Mathematics
Utrecht University
The Netherlands

Abstract and Acknowledgments

This thesis aims to investigate whether the MR-STAT k-space data is a suitable input for the ESPIRiT algorithm. So, the question is whether the reconstructed coil maps from MR-STAT data obtained with ESPIRiT provides reliable results. After the introduction, chapters 2 and 3 provide prior knowledge and cover subjects such as MRI, parallel imaging and MR-STAT.

In the last chapter, chapter 4, the actual research takes place. A lot of different datasets are investigated, with different scaling factors and different masks. We will take a closer look at the radio frequency train and the effect it has on the tissue. The conclusion of our research is that all MR-STAT data is suitable as an input for the ESPIRiT algorithm as long as there is no noise. Unfortunately, there is a bit of noise present at all MR-scans. If there is noise, the RF-train needs to be looked at and the signal from the tissue it causes. If that signal is equal to zero at the moment of acquiring the central lines of the k-space, it is not a suitable input for ESPIRiT algorithm.

Writing this thesis would not have been possible without the help of Oscar van der Heide, whom initiated the topic, provided all of the datasets, helped looking for useful articles, etcetera.

Contents

1 Introduction

Magnetic Resonance Imaging (MRI) is an imaging technique that measures the presence of spatial frequencies in the human body, which is done by spinning the protons in the body. After gathering the information, it is stored in a k-space, which is a matrix. The desired image obtained with (qualitative) MRI is reconstructed from the k-space by applying an inverse FFT (fast Fourier transform) to this data. Taking these data acquisition steps is a time-consuming process. To speed up this process, parallel imaging techniques are developed.

Parallel imaging (PI) uses the information from the multiple coils that are used for scanning, for instance the coil sensitivity maps, to reconstruct an image without artefacts and aliasing from an undersampled k-space. To use those PI techniques, knowing the coil sensitivity maps is required. There are two options for gathering the coil sensitivity maps, the first option is making a separate scan. The second option is collecting the information via the measured k-space data itself, this can be done using a technique like ESPIRiT.

Unfortunately, ESPIRiT relies on the assumption that during the scan the spins are in a steady state. Which is true for (qualitative) MRI, but not for the recently developed MR-STAT technique. MR-STAT is an imaging technique that estimates all the information needed for the quantitative tissue parameter maps, for example T1 or T2, in one single short scan. In this acquisition the spins do not reach a steady state, as mentioned before, the k-space data is thus in a transient-state. To get all the tissue parameter maps out of the k-space data, a Bloch-equation based model is fitted to the (transient-state) k-space data. The coil sensitivity maps, that can be reconstructed with the ESPIRiT algorithm, can be a tool to improve the signal to noise ratio (SNR) of the quantitative parameter maps from the MR-STAT. The coil sensitivity maps reconstructed with ESPIRiT can also be used to reduce the acquisition time even more.

The goal for this thesis is to research whether the ESPIRiT algorithm is useful and accurate on the MR-STAT k-space data that is in a transient-state. And if so, to come up with an explanation of which part of the MR-STAT data is suitable as an input for ESPIRiT and which part of the data will not work, based on, for example, the flip angle train.

This thesis starts with two chapters with knowledge needed to reach the aim stated before. The first one, chapter two, is about quantitative MRI in general and parallel imaging. The third chapter is about MR-STAT, which is a recent developed imaging technique that accelerates the acquisition. Then, in the last chapter, we will investigate how suitable ESPIRiT is for the MR-STAT k-space data.

2 Parallel Imaging: focused on Sense

This chapter is mainly about parallel imaging and the artifacts that can be undone by, for example, Sense or Grappa. First there is a part about qualitative MRI in general and the parallel imaging techniques. Then we will discuss a method for reconstructing the image from aliasing.

2.1 Introduction to qualitative MRI

Before we can discuss parallel imaging (PI is the shorter term that will be used) and how important its development is, we will describe how spatial encoding in MRI works. An MRI image shows the localization of the NMR (nuclear magnetic resonance) signal by frequency. To determine where a signal in 3D is arising from, three steps are needed: the first step makes a 'slice' of the body, in the second step a segment of the slice is selected and in the final step a part of the segment is selected. These steps will be explained below.

2.1.1 Signal localization

Slice selection

For the selection of a slice of the body, in order to make an image, determination of the resonant frequency in this slice is necessary. Therefore the Larmor equation, shown in equation 1, is used.

$$\omega = \gamma \cdot B \tag{1}$$

In the equation ω is the resonant frequency, γ is the gyromagnetic ratio, which is 42.58 MHzT^{-1} for a proton, and B is the strength of the external magnetic field at the position of the proton. The strength of the external magnetic field (B) in one slice of the body can be determined, and based on the strength of the magnetic field, the resonant frequency (ω) for that slice can be calculated. In figure 1, slice selection is shown. In this image the slice from 1.55T till 1.57T is selected, if we put the strength of the external field into the Larmor equation we get: $\omega_1 = 42.58\mathrm{MHzT}^{-1} * 1.55\mathrm{T} = 66\mathrm{MHz}$ and $\omega_2 = 42.58\mathrm{MHzT}^{-1} * 1.57\mathrm{T} = 66.85\mathrm{MHz}$. This determines the slice thickness and thus the part of the body that is selected for imaging.

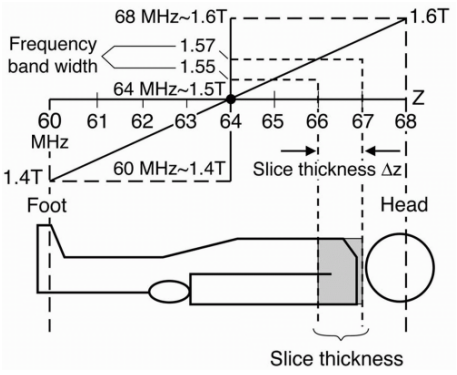


Figure 1

Frequency encoding

Frequency encoding is a localization method, during which, by applying spatially varying magnetic flied gradients, signals from different locations can be separated. By changing the frequency linearly with the position, the Fourier transform gives the relation between the signal detected and the spatial distribution of the signal. In other words, within the imaging plane, from left to right a small gradient is applied, this alters the frequencies allowing us to perform spatial encoding in the x-direction. This way we can select a segment from the slice selected in the previous step.

Phase encoding

There are three directions in which the data can be collected, the first direction is G_z in which the slice selection takes place, then there is G_x in which the frequency encoding takes place. Lastly G_y in which the phase encoding takes place. The y-direction is done using a gradient that alters the phase of the signal. The phase encoding of a squared image of size 256×256 requires 256 unique phase encoding steps.

The phase encoding step fills up the k-space, where k_x is the frequency and k_y is the phase. In this thesis the k-space is filled from the top left corner to the bottom right corner. The protons in the scanning each have a distinct frequency and phase. They are unique for each proton due to the x and y gradients, this makes it possible to encode x and y coordinates for a pixel at the place of the proton. Now that the acquisition of the spatially encoded data is done, it is time to reconstruct the MR image, this will be done in the next section.

2.1.2 Voxel by voxel quantification

The frequency and phase data are saved as points in the 2-dimensional k-space. Reconstructing the image from this k-space data is done with a 2-dimensional inverse Fourier transform. The discrete Fourier transform in 2D is given in equation 2 and its inverse in equation 3.

$$\text{discrete Fourier transform: } F(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\cdot e^{-2k\pi\left(x\frac{m}{M}+y\frac{n}{N}\right)} \tag{2}$$

$$\text{inverse discrete Fourier transform: } f(m,n) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(x,y)\cdot e^{2k\pi\left(x\frac{m}{M}+y\frac{n}{N}\right)} \tag{3}$$

In this equation $f(m,n)$ is the inverse Fourier transform of $F(x,y)$ and it is the pixel value at coordinates (m,n). $F(x,y)$ is the Fourier transform of $f(m,n)$ and it is the value of the image in the frequency domain, k-space, corresponding to the coordinates x and y. The constants M and N are the dimensions of the image domain.

By applying the inverse discrete Fourier transform to the k-space, obtained through the steps above, the MR image is created.

2.1.3 Acquisition time

The problem with qualitative MRI is that it is a time-consuming process. The acquisition time is calculated according to the following equation:

$$\text{Acquisition time} = (\text{TR})(\text{Ny})(\text{NEX}) \tag{4}$$

In this equation TR stands for repetition time, which is the amount of time that is between two consecutive pulse sequences that are applied to the same slice. Ny stands for the number of phase encoding steps and NEX stands for the number of times the whole sequence is repeated.

The acquisition time and the reconstruction time added up causes it to be a long process. To make MRI faster, parallel imaging techniques are developed. These technique are faster because of undersampling. These parallel imaging techniques will be explained in the next section.

2.2 Introduction to Parallel Imaging

In parallel imaging the information about the coils, such as the placement and sensitivity, is used to assist with the spatial localization of the MR-signal. In both parallel imaging and 'normal' qualitative medical imaging there are multiple coils used. The difference between those two is that for PI we can reduce the amount of gradient encoding steps needed for imaging. This is done by undersampling the k-space. If the undersampling factor of the k-space is 2, for example only the even numbered lines are taken into account, this will theoretically lead to a scan time reduced with factor 2. In practice this factor is, unfortunately, smaller than two. Undersampling the k-space will reduce the imaging time drastically, so it accelerates the acquisition, but leads to another problem.

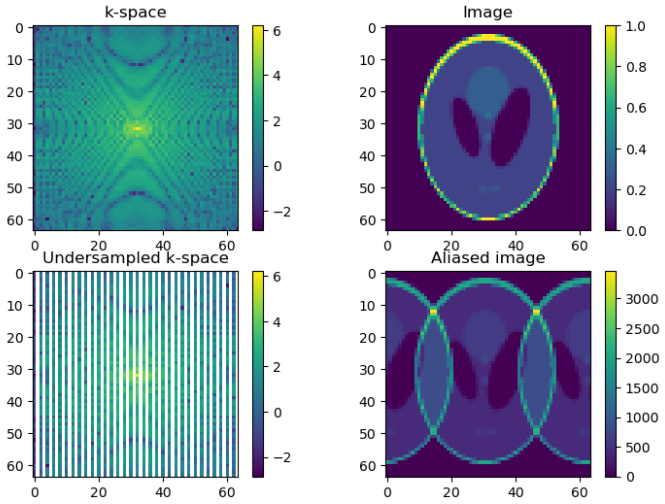


Figure 2

The problem caused by undersampling is aliasing, this is shown in the fourth image of figure 2. To go from the images on the left side to the images on the right side inverse Fourier transform is applied. For reconstructing the image from the aliasing, sort of unfolding the image, a couple of algorithms are developed, these will be explained in section 2.3.

2.3 Methods for reconstructing the image

There are a couple of methods to reconstruct an image from an aliased image, an image with wrap-around artifacts as shown in figure 2. We are going to discuss two different methods. The first method we are going to take a quick look at is Grappa (GeneRalized Autocalibrating Partial Parallel Acquisition). Grappa is primarily performed in the k-space, so before the image is reconstructed. The second method we will discuss is Sense (SENSitivity Encoding). Sense is primarily performed in the image space, so after reconstruction of the data from each coil individually.

2.3.1 Grappa

As stated before, Grappa is primarily performed in the k-space. In this section a very brief summary about the algorithm will be given. The Grappa algorithm consist of four main steps; acquire, estimate, generate and combine.

During the first step, that is acquiring, the k-spaces are filled. Because of the undersampling not all lines are filled and there are a lot of gaps, but the lines in the centre of each of the k-spaces must be filled. After that, the estimating step follows, in this step the k-spaces are filled up again by making use of the central k-space lines. The third step is taking the Fourier transform of the filled k-spaces, this gives an image for each of the coils that is not an aliased image. However, the image was made from one of the coils, so it gives a clearer result near the coil than further away. In the last step those images, that are made from the perspective of their matching coil, are combined. The combining is done with the sum of squares method. This is a really short and quick illustration of how Grappa works, more can be read in source [5] in references.

2.3.2 Sense

For the reconstruction of an aliased image, Sense is a frequently used algorithm. Therefor, the Sense algorithm will be explained and coded in this section. A part of the input for the SENSE algorithm is the undersampled k-space simulated from four coils, if we apply fftshift(fft(fftshift())) to these four undersampled k-spaces, we get the images shown in image 3. The other part of the input consists of the coil sensitivity maps. A small disclaimer about those maps beforehand: the coils are on the sides of the FOV, this results in very easy coil sensitivity maps, normally they are a bit more difficult. In the four images shown in figure 3 the location of the coils can be guessed, for coil 1 for example, the stronger signal is at the upper side, so the placement of the coil is at the upper side.

The goal is to reconstruct the image, so to unfold the aliased images, and combine them into one image. The first step of the Sense algorithm is collecting the input, in other words; the acquisition of the data, to fill up the undersampled k-spaces.

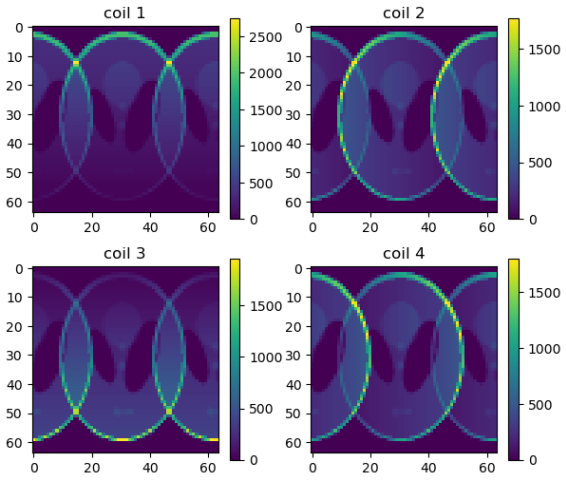


Figure 3

The Sense algorithm made in Julia is shown in appendix A. The steps that are taken will be explained here. In block 1 to 4 the input data is created, consisting of the 4 coil sensitivity maps and the four aliased images, shown in figure 3. In block 5 the Sense algorithm is programmed, starting with a double for loop, which goes through all the elements of the four aliased images. Then C_{nu} is collected, which takes two coordinates for all 4 of the coil sensitivity maps, which gives us a matrix of 4×2. After that, the determination of the inverse of C_{nu} is done, because C_{nu} is not a square matrix, we use the least square method. In equation 5 is shown how the the least square method is used for determining C_{inv}.

$$C_{\text{inv}} = \left((C_{\text{nu}})^T * C_{\text{nu}}\right)^{-1} * (C_{\text{nu}})^T \tag{5}$$

Thereafter, a part of the aliased images is picked. For every coil, the element with the x and y value for the current *for loop* is picked. This 4×1 matrix is called I_{aliased}. We multiply C_{inv} with this 4×1 matrix and call this $\text{Im}_{\text{folded}}$. In matlab there is a single function to determine the $\text{Im}_{\text{folded}}$, this replaces equation 5 and the multiplication. The function is shown in the following syntax: $\text{Im}_{\text{folded}} = C_{\text{nu}} \backslash I_{\text{aliased}}$.

After calculating $\text{Im}_{\text{folded}}$, the last step is to unfold this image, therefore the following syntax is used: $\text{Im}[y : 32 : 64, x] = \text{Im}_{\text{folded}}$, which gives us the desired image.

A disadvantage of Sense is that the coil sensitivity maps are required as input for the algorithm. For Grappa those maps are sort of created out of the central k-space lines, as seen in the previous section. It is possible to reconstruct those coil maps by making a scan prior to the real scanning or using the ESPIRiT algorithm, which will be briefly looked at in section 4.1.

3 Introduction to MR-STAT

The 'original' qualitative MRI consists of two main steps, first the estimation of the magnetic moments distribution, as we have seen in the section 2.1.1. The second step is the voxel by voxel quantification, which has been discussed in section 2.1.2. Taking those steps gives us a simplified computational process, but a disadvantage is that it is relatively time-consuming. The idea of MR-stat is that the signal localization and the parameter quantification are done at the same time. In MR-STAT the signal from the coil is directly converted to quantitative maps, while in quantitative MRI the signal is first converted to a k-space, then a Fourier transform is applied, then it is converted to magnetization images and then finally it is converted to the quantitative maps.

The whole name for MR-STAT is Magnetic Resonance Spin TomogrAphy in Time-domain. The time evolution of a group of likewise behaving spins is called spin isochromat. This spin isochromat depends on two things; the pulse sequence (the RF-train) and the biophysical properties of the tissue, such as T_1 and T_2.

With the MR-system, the time domain signal s is obtained at time t. A discrete formula of the signal s at time t is shown in equation 6.

$$s(t) = \sum_{j=1}^{N_v} m(\theta(\mathbf{r}), t) e^{-2\pi i \mathbf{k}(t)\cdot\mathbf{r}} d\mathbf{r} \tag{6}$$

In the equation above $k(t)$ is the gradient trajectory for the pulse sequence, the variable m is the transverse magnetization component of a spin isochromat and θ is a variable that contains the biophysical properties that are important for MR, such as T_1 and T_2. In figure 4 the principles of the MR-STAT are shown, with the 2 steps; the acquiring of the signal and the reconstruction.

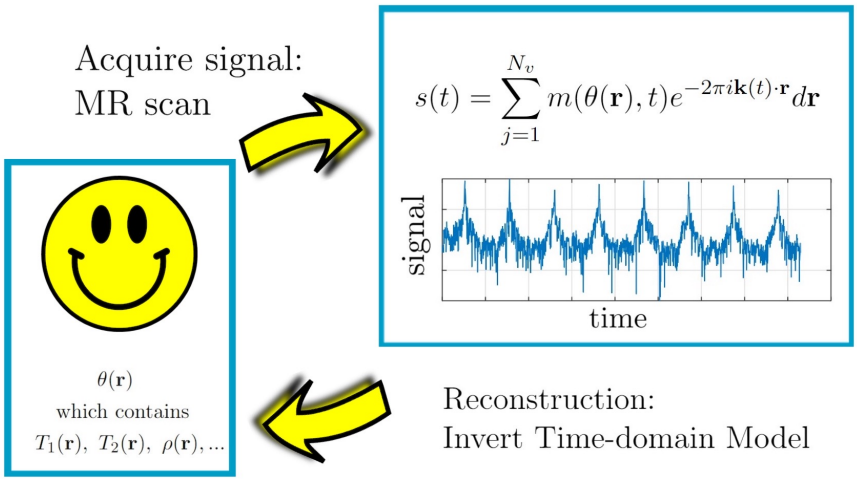


Figure 4: MR-STAT principles based on `https://compimag.org/projects/mrstat.html`

In the MR-STAT algorithm we do not wait for a steady state, there is a radio frequency train applied to the spins, this will be further explained in the next section.

3.1 RF train of the data set

The spins will not achieve a steady state during the scans with an MR-STAT, instead they are in a transient-state. To get all the tissue parameter maps out of the k-space data, a Bloch-equation based model is fitted to the k-space that is in a transient-state. The coil sensitivity maps, that can be reconstructed with the ESPIRiT algorithm, that will be looked at in section 4.1, can be a tool to improve the signal to noise ratio (SNR) of the quantitative parameter maps from the MR-STAT. The coil sensitivity maps reconstructed with ESPIRiT can also be used to reduce the acquisition time even more.

In figure 5 the radio frequency (RF) train for five times filling the k-space with the MR-STAT algorithm is shown. The green lines show the start/stop line of the filling of the k-space, so the first k-space is filled from 0 till 224 on the x-axis. The red lines are the middle of the filling of the k-space, that is when the centre of the k-space is filled. The importance of these lines is covered in section 4.1.1. Those lines are an important input for the ESPIRiT algorithm.

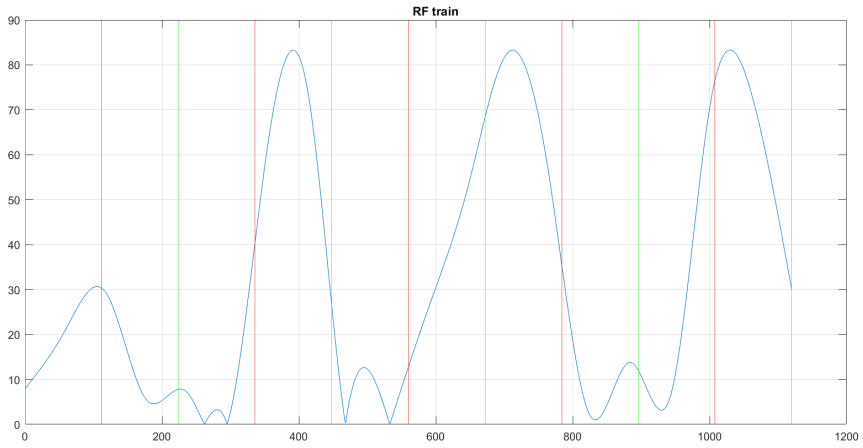


Figure 5

More information about the MR-STAT can be found in the first part of the article 'Accelerated MR-STAT reconstructions using sparse Hessian approximations', in particularly section II. Theory A. MR-STAT Framework, see references [8]. Also, the article 'Fast quantitative MRI as a nonlinear tomography problem' contains a lot of information about the MR-STAT, see references [10].

4 Coil sensitivity maps for MR-STAT

The aim of this chapter is to look at the accuracy of the reconstructed coil sensitivity maps for a k-space constructed with the MR-STAT algorithm, and which characteristics influence the accuracy of the reconstruction. For the reconstruction of the coil maps we use the ESPIRiT algorithm made by Michael Lustig [6] and based on the paper from Uecker about ESPIRiT [4].

Because the spins do not reach a steady state in the MR-STAT scan, as we have seen in chapter 3, the coil sensitivity maps are often more difficult to reconstruct. To solve this, making a scan before the real scanning to reconstruct the coil maps is an option, but this has some disadvantages. The major disadvantage is that the tissue in the scan can move a bit in between the scans and this influences the coil sensitivity maps. Making a scan prior to the real scanning causes that the prior scan could differ from the actual coil maps during the MR scanning. Thus, it is more accurate to reconstruct those coil maps out of the data, which is normally done with ESPIRiT. In this chapter we will investigate whether the ESPIRiT algorithm also works on MR-STAT k-space data that is in a transient-state.

To reach our goal, some prior knowledge is needed. First, there is a short part about the ESPIRiT algorithm, then the algorithm is applied to both the qualitative MRI data, with a steady state and a flip angle of 30 degrees (FA 30), and to the MR-STAT data, in transient-state. These will be compared, and we will take a look at how accurate the reconstructions are. After that, some adjustments are done to the reconstruction, scaling for example. Finally the RF-train will be looked at, and after that there is a conclusion about the use of ESPIRiT on MR-STAT k-space data that is in a transient-state.

4.1 Short introduction to ESPIRiT

For parallel imaging, different coils with each different coil sensitivity maps are used, as seen in section 2.2. To accelerate the acquisition of the k-space, we can undersample the k-space. For the reconstruction of the image, the coil sensitivity maps and their information can be used, as seen in section 2.3.2 about Sense. In the ESPIRiT algorithm, two different types of reconstruction methods are sort of combined. The first one is Sense, an algorithm that is build on knowing the coil sensitivity maps beforehand. The second algorithm is Grappa (section 2.3.1), which is build on the correlation between the pixels in the k-space, in particular the local kernels.

The algorithms where the coil sensitivity maps are known beforehand, allow for an optimal reconstruction, thus the reconstructions are more accurate, in the sense of minimum mean square error for example. Another thing that is better, when we use an algorithm like Sense, is that they are more general, in other words, they can be used with different sampling trajectories. For those algorithms we need the coil sensitivity maps, to make an accurate reconstruction of the image. That is where ESPIRiT comes in, this algorithm can reconstruct the coil sensitivity maps.

ESPIRiT shows that we can rapidly reconstruct those coil sensitivity maps in high quality. This is done by taking the dominant eigenvalues of the central k-space, those eigenvalues behave as sensitivity maps. More about ESPIRiT can be found in the article with reference number [6].

4.1.1 The k-space lines

The lines in the k-space give all the information necessary for reconstructing the coil maps, as it is the only input for the ESPIRiT algorithm. In this section a bit about k-space lines and what can be done with them will be shown. In the ESPIRiT algorithm, the central lines of the k-space are selected and used, why those lines are used will be explained in this section.

To create an image from the k-space lines, the matlab function *ifft2c(A)* is used. This function returns the discrete inverse Fourier transform (in 2 dimension with complex numbers). The inverse Fourier transform of a fully filled k-space is shown in figure 6. It is a row of eight images, where each of the images corresponds with a different coil.

Figure 6

In figure 6 the whole phantom is visible. Applying *ifft2c* to the whole k-space gives us a clear image with details and contrasts. If only the central lines of the k-space are taken into account, those are the lines used in the ESPIRiT algorithm, we get the images shown in figure 7. The contrasts are still visible, it is almost as if it is moving, because all the details are gone.

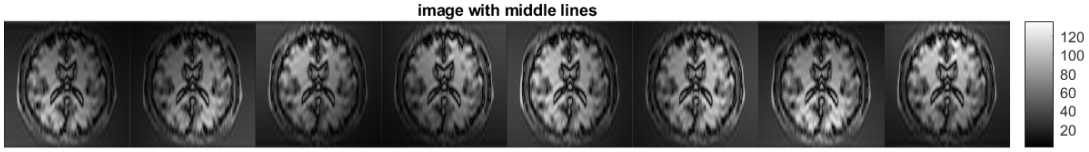


Figure 7

If the central lines are not taken into account, and so only the outer lines of the k-space are used, we get the images shown in figure 8. The details are visible, but the contrasts are as good as gone.

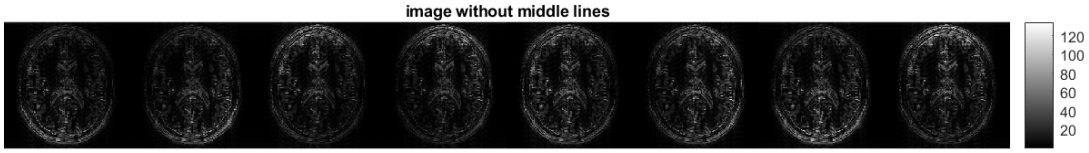


Figure 8

Thus the outer lines provide the details and the inner lines provide the contrast. For the algorithm only the inner lines are used, because a coil sensitivity map is in general quite smooth so it does not contain details. Therefore the ESPIRiT algorithm uses the inner lines.

4.2 Reliability of the reconstructed coil maps

The reconstructions of the coil maps in this section are made with the ESPIRiT algorithm of Michael Lusting [6]. In figure 9 the real coil maps are shown. Those are the coil maps used for making the scan of the phantom, normally there are no 'real' coil maps available, because it differs for each tissue in the scanner. For this experiment they are needed, because for determining how accurate the reconstructed coil maps are, we are comparing real coil maps and the reconstruction of the coil maps. In figure 10 one of those reconstructions is shown, this is the reconstruction for a data set with a constant flip angle of 30 degrees.

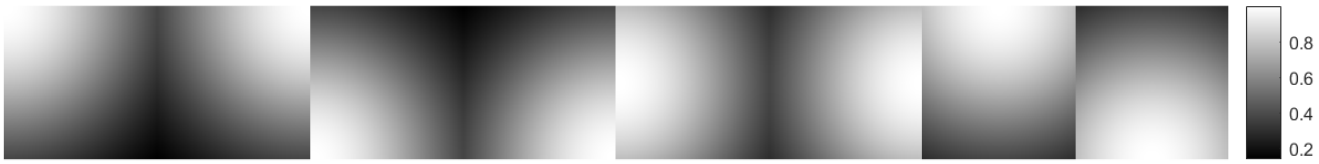


Figure 9: real coil maps

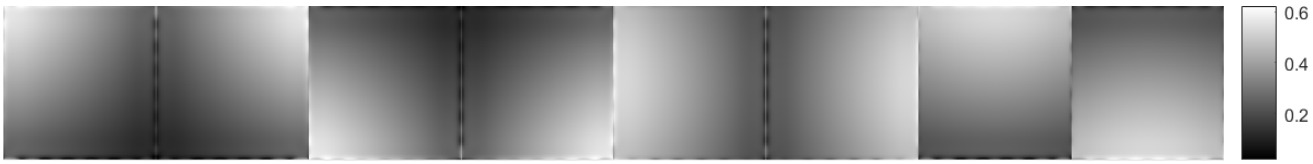


Figure 10: reconstructed coil maps

If we compare the two images there are two things that immediately stand out, namely the scale and the edges. It is noticeable that the scale for the real coil maps is from 0.1 till 1, while the scale for the reconstruction is from 0 till 0.6. So, the scales differ, this will be undone by scaling in section 4.2.2 and even better in section 4.2.4. Another problem is in the edges of the reconstructed coil maps. The edges of figure 10 are not smooth, and are suddenly quite extreme, they are varying between quite low and quite high in terms of the grayscale value.

It is important to determine how accurate a coil map reconstruction is. The calculation of how accurate the reconstruction is, is done by taking the L^2-norm of the difference between the reconstructed coil map and the actual coil map, and we call this difference the reconstruction number, see equation 7, where n is the number of coils.

$$\text{reconstruction number} = \sqrt{\sum_{i=1}^{n} \Big(\text{realcoil}(i) - \text{reconstructed}(i)\Big)^2} \tag{7}$$

The lower the reconstruction number the better, because if the reconstruction number is low, the difference between the real coil maps and the reconstruction of those maps is small.

In the next sections, the reconstruction number is calculated for a couple of different inputs. The four different inputs are: the reconstructed coil map directly given by the ESPIRiT algorithm, the re-scaled reconstruction, a masked reconstruction and a reconstruction with a given mask. These inputs will be compared with the real coil map according to the calculation given in equation 7.

4.2.1 Reconstruction for the data directly

In this section the real coil maps, see figure 9, are compared with the reconstructed coil maps for 5 datasets with a constant flip angle of 30 degrees (FA30) and 5 datasets from MR-STAT. The comparison is done with the formula previously given in equation 7, which gives us the reconstruction numbers. In the two tables 1a and 1b, the reconstruction number of the 10 different datasets (each consisting of 8 different coil maps) are given. For one of the constant flip angle datasets the coilmap reconstruction is shown in figure 10, this is the second dataset in table 1a, so the reconstruction number is 559,28.

Table 1: Reconstruction numbers

(a) For 5 FA 30 data sets

1	2	3	4	5
1816,6	559,28	559,25	559,23	559,28

(b) For 5 MR-stat data sets

1	2	3	4	5
561,33	558,72	1826,4	558,86	559,10

The first value of table 1a is exceptional high, at first we thought this was due to the steady state that is not reached yet, because it is the first scan, but later in section 4.2.4 we came to another explanation which is, most probably, the real reason. The second table, table 1a, also contains a particularly high value, namely the third value, this will also later be explained in section 4.2.4.

If we take out the deviating values, we get 559,26 as the average reconstruction number of the FA 30 dataset and 559,50 as the average of the MR-STAT dataset. These values are almost the same, but quite high. For better comparison scaling the coil maps could be an option. This will be done in the next section.

4.2.2 Reconstruction for scaled data

To be able to compare the coil maps better, and to lower the reconstruction number, we want to scale the maps equally. The scaling will be done in such a way that the maximum value of the reconstruction maps goes up to the maximum of the real coil maps. As you can see in figure 10 the maximum value of the reconstruction is only 0.6, while the maximum of the real coilmap is 1. For this scaling we use the following equation:

$$\begin{aligned} \text{coilmap}_{\text{scaled}} &= \text{coilmap} \ * \ \text{scalingfactor} \\ \text{with} \quad \text{scalingfactor} &= \text{realcoil}_{\text{max}}/\text{reconstruction}_{\text{max}} \end{aligned} \tag{8}$$

We compared the scaled reconstructed coil maps and the real coil maps according to equation 7, this gives us the reconstruction numbers given in table 2. If we, again, take the averages without the deviating values, we get 184,16 for the FA 30 dataset and 154,83 for the MR-STAT dataset. So the scaled MR-STAT maps are closer to the real coil maps than the maps reconstructed from the FA 30 data, this is probably due to the errors in the edges that are stronger for FA 30, causing a less correct scaling factor.

Table 2: Reconstruction numbers

(a) For 5 FA 30 data sets

1	2	3	4	5
2204,6*	185,31	183,96	182,33	185,05

(b) For 5 MR-stat data sets

1	2	3	4	5
123,47	165,71	2306,5*	146,94	183,19

Comparing table 1, from the previous section without scaling, and table 2, with scaling, we see that the exceptional high values become even higher, but the rest of the values decreases quite a bit. So in general, the scaling lowers the reconstruction number. The high values becoming higher is due to a wrong scaling factor for those particular cases, this will be explained and solved in section 4.2.4.

One of the striking differences between figure 9 and 10 is now solved, namely the scaling. This decreased the reconstruction number quite a bit. The other problem, the deviations in the edges, will be looked at in the next section, through a mask.

4.2.3 Reconstruction for data with a L^2-norm mask

In figure 10, as stated before, there is a lot of noise at the edges of all the coil maps. The reason for this noise, is that the proton density of the background of the input data is not equal to zero. So there is a proton density that influences the coilmaps while it is the background that does not contain any tissue. Therefore a new dataset, where the proton density in the background is equal to zero, is introduced in section 4.2.4. Luckily the artifact at the edges should not really influence the resulting image. The edges of the field of view must not contain any tissue, because this can lead to major aliasing problems. Thus the value of the edges of the coil maps are less important for the reconstruction of the image. To ensure that the edges do not play such a major role in the calculation of the reconstruction

number, only the parts of the coil maps that correspond to the parts with tissue are taken into account. Only taking this part is done by a mask. A mask is a binary filter that turns pixels 'on' or 'off', the idea is that the pixels in the tissue are turned 'on' and the pixels outside of the tissue are turned 'off'. The code for creating the mask can be found in appendix B.

As shown in the code in appendix B, we first create a 'total' image, which is a composite image of the 8 images simulated by different 8 coils. This is done by taking the L^2-norm. After that, the maximum is determined, which is multiplied by a factor between 0 and 1. The number that comes out of the calculation serves as a sort of threshold. If the factor is 0, all of the pixels are 'on', because all the values are bigger than 0. For 1 it is exactly the opposite, then it is equal to the maximum, and no value is bigger than the maximum, so all the pixels are 'off'. So we try to find a factor that ensures the correct ratio between 'on' and 'off' pixels. The images of the 5 'total' FA 30 data sets and the 5 'total' MR-STAT data sets after applying the L^2-norm mask are respectively shown in figure 11 and 12. The factors that are used for the FA 30 datasets are shown in table 3 and the factors for MR-STAT are shown in table 4.

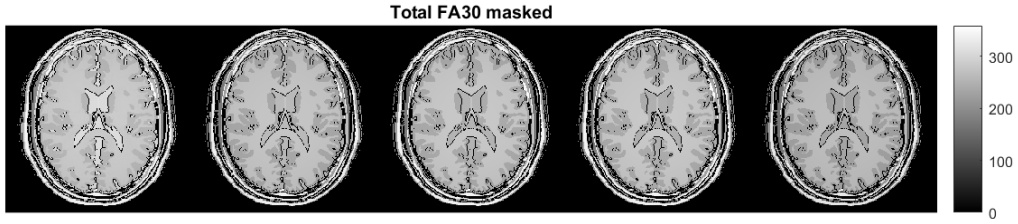


Figure 11

After applying the mask, the scaling was still a bit off. The edges with the extreme values are 'off' pixels now, because of the mask, so they do not take part in the scaling. That means we can scale again with only the 'on' pixels taken into account, hopefully this gives a more accurate result.

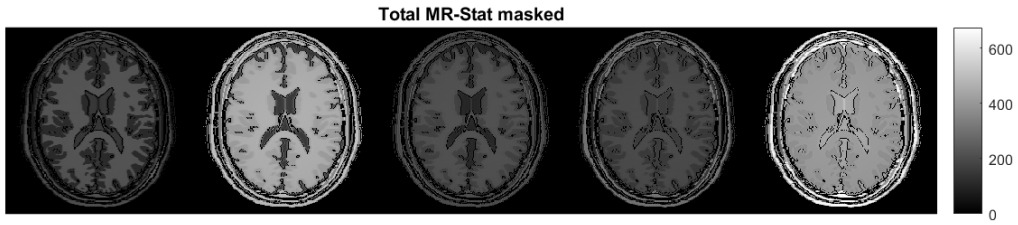


Figure 12

The scaling is done in the same way as in the previous section, so according to equation 8. After applying the mask to the reconstructed coil maps and scaling them afterwards, we calculate the reconstruction number. This gives us the results shown on the bottom line of the tables 3 and 4.

For comparing our data, in particular the reconstruction numbers, the amount of pixels that is 'on', so the amount of pixels that is taken into account for calculating the reconstruction number, must be of comparable size. That is why the row with pixels in mask is added to our tables. We tried to set the amount of pixels in the mask around 28000 for all of our data sets, only this was quite difficult, because changing the factor a bit could influence the amount of pixels radically. The amount of 28000 pixels is chosen, because it is more than half of the total amount, namely 50176.

Table 3: Reconstruction numbers with mask for FA 30 data

scan number	1	2	3	4	5
pixels in mask	30569	28506	27092	27125	28566
factor	0,68	0,63	0,64	0,64	0,62
reconstruction number	1843,2	39,90	39,61	39,72	39,91

Table 4: Reconstruction numbers with mask for MR-Stat data

scan number	1	2	3	4	5
pixels in mask	27900	50176	30739	29263	27918
factor	0,245	0,2	0,51	0,4	0,557
reconstruction number	44,29	165,71	1852,6	61,28	41,04

For the second dataset of the MR-STAT data, the number of pixels in the mask is equal to the total amount of pixels. Finding a factor that gave us around 28000 pixels did not succeed, it came down to all pixels on or all pixels off.

The mask reduces the reconstruction number of the 'normal' values, so disregarding the two extraordinary high values and the dataset where we did not find a suitable mask for, on average by a factor of approximately 4, while the amount of pixels is reduces with a factor less than two, so the mask cancels out the errors at the edges.

The two high values have not really decreased compared to the values in section 4.2.1, while less pixels are taken into account. This is probably due to wrong scaling, the scaling factor does not contain any imaginary numbers, because we base it on the absolute value of the maximum, so the scaling factor can be calculated wrong. For example if the maximum is $\frac{1}{2}i$, then the absolute value is $\frac{1}{2}$, and the whole map is scaled with a multiplication of 2, while in fact it should be multiplied with $-2i$ to get 1. This can cause a wrong scaling if a value has a real and an imaginary part, as shown in section 4.2.2.

Another point where the reconstruction number is too high is the second MR-STAT scan, because we did not succeed in finding a suitable factor, and with that a suitable mask. So the mask function that we have created is not for general use. Also picking a factor yourself is not the most reliable part. Therefore we use a mask that is based on the real tissue, this will be used in section 4.2.5.

4.2.4 Reconstruction for new data and a better way of scaling

Remember the errors in the edges of the reconstructed coil maps, seen in figure 10. They are probably due to the proton density of the background that was not equal to zero. Because of that, the background, which is the least important part, played a major role in the reconstructed coil maps. The new data, that will be used from now on, will have a proton density of zero in the background. The reconstructed coil maps belonging to the new data, do not contain any value on the edges, see figure 13. For comparing these reconstructed coil maps to the real coil maps we need to take out the values that are zero in figure 13 in the real coil maps.

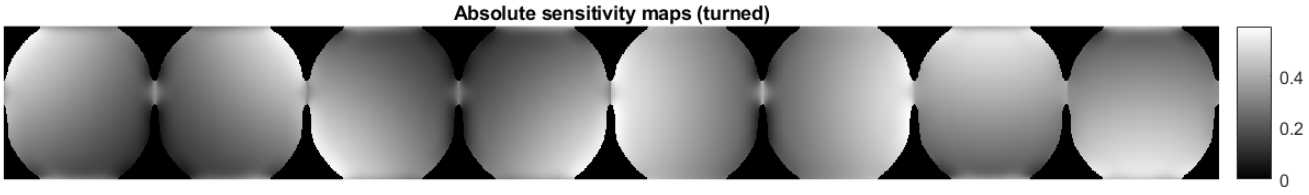


Figure 13

The total amount of pixels in the real coil maps is 401.408, after canceling out all the values that are zero in the reconstructed maps, we are left with 341.592 pixels. That comes down to 42699 pixels for each coil. These pixels will be used to calculate the reconstruction number. This gives us table 5a for the FA30 data and table 5b for the MR-STAT data.

Table 5: Reconstruction numbers

(a) For 5 FA 30 data sets

1	2	3	4	5
124,99	124,75	124,74	124,71	124,75

(b) For 5 MR-stat data sets

1	2	3	4	5
2097,2	2113,6	2113,6	123,54	119,92

There are a couple of things that are striking in the tables above, one of the things is the first value of the FA30 dataset is not that high anymore, so in this measurement the steady state is reached before the scanning had started. The second thing is that the reconstruction numbers of the first three scans of the MR-STAT data are quite high. So, something is wrong for these scans. By looking at the code we concluded that the scaling factor was incorrect, as is mentioned before. This is probably wrong, because it was only based on the absolute value of the maximum and could not contain imaginary numbers. Therefore we created a new way of scaling with a Matlab function called 'backslash', which actually is a \.

$$\text{if } u \approx \alpha v, \text{ then } \alpha \text{ can be found with: } \alpha = v \backslash u$$

The equations above show how our scaling factor is calculated with the Matlab function. In these equations u is a vector of the real coilmap and v is a vector of the reconstructed coilmap. For this vector we chose column 112, because it is in the centre.

The reconstruction numbers of new datasets scaled with the backslash function are shown in table 6a and 6b.

Table 6: Reconstruction numbers

(a) For 5 FA 30 data sets

1	2	3	4	5
82,51	82,34	82,30	82,25	82,32

(b) For 5 MR-stat data sets

1	2	3	4	5
72,04	82,28	82,21	81,75	85,40

In the tables above we see that all the values have reduced compared with the values in table 5a and 5b. Very striking are those values that in the previous scaling where very high and now they are as low as the rest. Another noticeable thing is that the reconstruction numbers for the FA30 data are really similar, while for the MR-STAT data there is some variation. The averages, are quite similar, 82.34 for FA30 and 80.74 for MR-STAT.

Returning to the old data with a star (*) in the upper corner, namely the values in table 2, we can now conclude that those errors where due to scaling. If we apply the new scaling method for FA30 dataset 1 we go from 2204,6 to 95,12, and for MR-STAT dataset 3 we go from 2306,5 to 100,74.Those two new reconstruction numbers are of comparable size with the rest of the old data scaled with the backslash scaling. So now we can conclude that the scaling was off, and that the scaling factor with the possibility to have an imaginary part is a much better option.

In the next section, the reconstruction number is calculated for a given mask. This is a mask that is only 'on' at the sites of the tissue.

4.2.5 Reconstruction for the new data set and a given mask

Instead of creating the mask based on the maximum value, it is more accurate to know where tissue is. This given mask is only 'on' at the sites of the tissue and contains 29888 pixels that are included in the calculation of the reconstruction number.

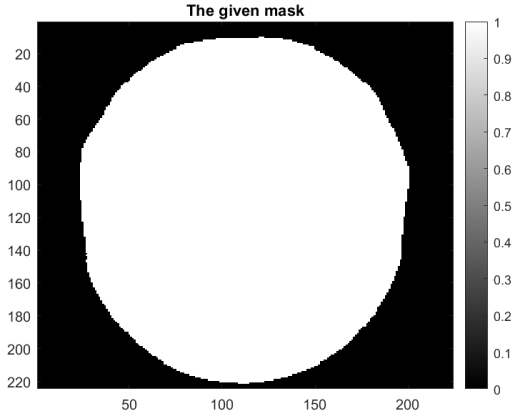


Figure 14

The given mask is shown in figure 14. Normally, it is not the case that there is a mask available of the tissue, but we can approach this mask with the method from section 4.2.3 to create a mask. For comparing the data, only using the pixels with tissue gives a more accurate picture of the reconstruction number. The reconstruction number of the coil maps scaled with the backslash method and masked with the given mask are shown in table 7. The reconstructed coilmaps are shown in C, along with the differences between the real coilmpas and the reconstructed maps. Most of the maps that show the differences only have high values at the edges of the mask, there is a small white line visible.

Table 7: Reconstruction numbers

(a) For 5 FA 30 data sets

1	2	3	4	5
23,21	23,20	23,20	23,20	23,20

(b) For 5 MR-STAT data sets

1	2	3	4	5
23,70	23,25	23,27	23,07	23,05

The values for the FA30 dataset are very similar, while the MR-STAT data varies a bit. In the next section we are going to see which aspects make the reconstruction number of the MR-STAT lower and which aspects cause a higher reconstruction number.

4.3 Comparing with the signals for the T1 and T2 tissue

In table 7b of the previous section the first reconstruction number is a bit higher than the other values. We think that this has something to do with the signal of the varying values in the RF-train, shown in figure 5. Therefore the signals are measured for the different tissues in the phantom with their corresponding T1 and T2 values.

The reaction of the ten different tissues on the RF-train, shown in figure 5, are plotted in figure 15. The vertical lines correspond to the acquisition of the central lines of the k-space. The central lines of the k-space are important lines for contrast and thus for reconstructing the coil maps, as seen in section 4.1.1. So, we focus on the values of the signals around those vertical lines. The T1 and T2 values for the ten different tissues are shown in table 8.

If we take a closer look at the different signals in figure 15, we see that during the first k-space center acquisition, so the first vertical line, the 9th line is equal to zero. This means that there is almost no signal coming from the tissue with $T_1 = 1.400$ and $T_2 = 0.100$. This might have caused the reconstruction number, the first variable of table 6b, to be a bit higher than the other values.

Table 8: T1 and T2 values of the tissues with the corresponding lines

line	1	2	3	4	5	6	7	8	9	10
T1	5.00	0.569	1.00	2.00	0.350	1.00	2.569	0.833	1.400	1.700
T2	0.100	0.329	0.047	0.280	0.070	0.100	0.329	0.083	0.100	0.300

To check whether the signal being off for a part of the tissue, as is the case in the first k-space acquisition, really influences our reconstructed coil maps, we created an extra dataset. In the extra dataset there is a significant part of the tissue that has the values $T_1 = 1.400$ and $T_2 = 0.100$.

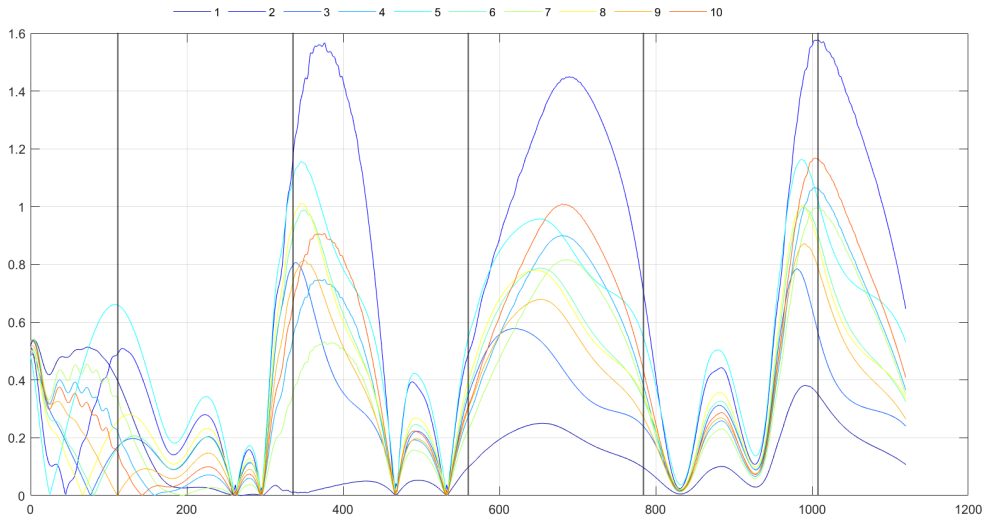


Figure 15: The signals for the different tissues

4.3.1 Reconstruction for a small extra dataset

The difference between the previous dataset and the extra dataset, is only that in this extra dataset there is a significant part of the tissue that has the values $T_1 = 1.400$ and $T_2 = 0.100$. This is expected to lead to a lack of signal in the first k-space, because the tissue does not send much of a signal, as shown in figure 15 for line 9. For this dataset we do not only calculate the reconstruction number, but also the root mean square. This is a function in Matlab that calculates the root mean square, this is done according to the following equation:

$$x_{\text{RMS}} = \sqrt{\frac{1}{N}\sum_{n-1}^{N} |x_n|^2} \tag{9}$$

In equation 9 for the calculation of the root mean square, N is the amount of elements in x. If we calculate the reconstruction numbers and root mean squares for this extra dataset, we get the results, shown in table 9. These values are only the results for the MR-STAT data. All of the root mean square values are multiplied by the factor 10^4.

Table 9: The reconstruction numbers for extra dataset

scan number	1	2	3	4	5
reconstruction number	22,71	23,00	22,94	22,79	22,97
root mean square	5,4712	5,9544	5,7777	5,5151	5,9032

For the root mean square the scaling factor as mentioned in section 4.2.4, has been changed. Instead of only taking the 112 column, we take the whole matrix, with the Matlab function $A(:)$, which creates a vector out of a matrix. This results in a more accurate scaling factor, and will be used from now on.

We expected that creating a bigger part of tissue that gives no signal, would lead to a less accurate reconstruction of the coil sensitivity map. If we take a look at the resulting reconstruction numbers, the opposite is true. The results are even better, so in this case we can conclude that a part of the phantom that gives no signal does not have a negative effect on the reconstruction maps. In other words, missing a signal from part of the tissue while acquiring the central lines of the k-space does not have a negative influence on the reliability of the reconstructed coil sensitivity maps. That it is not negatively affected might be because there is no noise on the data set, therefore the signal is almost zero in a big part of the scanning, and in the calculation this does not effect the reconstruction negatively. To investigate this assumption, we created an extra dataset in which we added noise. This extra dataset gives us a more realistic scenario.

In table 10 the root mean squares are calculated for three different data sets. The first dataset of the table is the same as used in section 4.2.5, the second comes from table 9 and the third is the same data set as the second but with noise added. All the root mean squares are multiplied with 10^4.

Table 10: The root mean square for three datasets

scan number	1	2	3	4	5
dataset 1	6,8426	6,3980	6,4211	6,0818	6,0959
dataset 2	5,4712	5,9544	5,7777	5,5151	5,9032
dataset 3	8,7060	5,7522	4,1334	3,6853	5,5725

If we take a look at the values in table 10, it is striking that the first value of the third dataset is higher than all the other values. This is due to the the signal that is collected from the tissue with T1 value equal to 1.4 and T2 value equal to 0.1. This signal is considerably reduced and almost approaches zero. Therefore the noise plays a major roll for the signal received from that area. As can be seen in the figure 16 below, the differences between the real and reconstructed maps are scattered, while the differences in the rest of the datasets are only at the edges of the mask, these differences between the real and reconstructed maps are shown in appendix D.

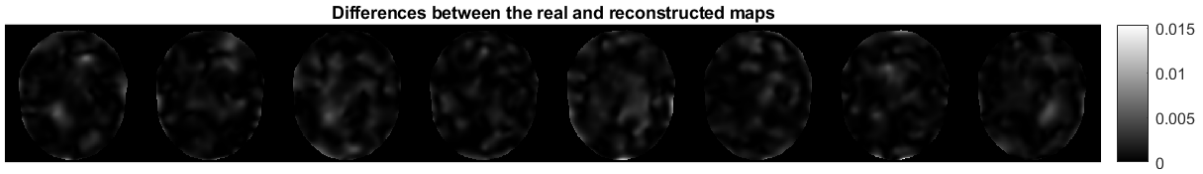


Figure 16: Difference for dataset 1

4.4 Advice and conclusion

The last data set is the most important data set to give an advice about how reliable the coil maps created with ESPIRiT from MR-STAT data are, and what properties the input data should and should not have. We were not able to derive conditions from our earlier data sets used in section 4.2.1 till section 4.2.5 in terms of the RF-train or the signal it causes for the tissue, because all of the MR-STAT data in those sections worked pretty well as an input of the ESPIRiT algorithm. Only the data set with proton density in the background caused problems for the reconstruction of the coil maps with the ESPIRiT algorithm, but that was the same for the FA30 datasets. Those problems due to proton density in the background is what would be expected, because tissue on the edges of the field of view will cause more problems along the way.

From the last data set that is used in section 4.3.1, where a significant part of the tissue has a T_1 value of 1.4 and a T_2 value of 0.1 and noise is added, we do get a case that is not suitable as an input for ESPIRiT. The reconstructed coil sensitivity maps contain a lot of errors and noise for the first k-space filling, while the rest of the k-spaces result in a more accurate reconstruction, shown in appendix D.

The errors for the first reconstruction are probably due to the 180 degrees inverse pulse at the start. The relaxation of the protons caused by this inverse pulse results in the signal, shown in figure 15, being equal to zero. Which causes the reconstruction not to be accurate.

In conclusion, all MR-STAT data is suitable as an input for the ESPIRiT algorithm if there is no noise. For a real quantitative MR scan or MR-STAT scan there is always a bit of noise. So for selecting a part of the MR-STAT data that is a suitable input for ESPIRiT, we have to take a look at the RF-train and the resulting signal for the different tissues that occur in the object. When the signal is equal to zero at the moment of acquiring the central lines of the k-space, it is not a suitable input for ESPIRiT. It is probably better to leave some time between the inverse pulse and the data set that will be used. Because then we do not have to take the relaxation of the protons into account, that will result in the signals being more similar to the RF-train.

References

[1] McRobbie D.W., Moore E.A., Graves M.J. and Graves M.R. (2003, 2006).
From picture to proton. ISBN-13 978-0-511-34849-5

[2] Deshmane, A., Gulani, V., Griswold, M.A., Seiberlich, N. (2012).
Parallel MR imaging (review)
https://mriquestions.com/uploads/3/4/5/7/34572113/deshane_pi_review.pdf

[3] Hashemi, R. H., Bradley, jr. W. G. (1997).
MRI the basics ISBN 0-683-18240-4

[4] Uecker, M., Lai, P., Murphy, M. J., Virtue, P.,
Elad, M., Pauly, J. M., Vasanawala, S. S., & Lustig, M. (2014).
ESPIRiT–an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA. Magnetic resonance in medicine, 71(3), 990-1001.
https://doi.org/10.1002/mrm.24751

[5] Santiago Aja-Fernández, Gonzalo Vegas-Sánchez-Ferrero, Antonio Tristán-Vega, Noise estimation in parallel MRI: GRAPPA and SENSE, Magnetic Resonance Imaging, Volume 32, Issue 3, 2014, Pages 281-290, ISSN 0730-725X, https://doi.org/10.1016/j.mri.2013.12.001 https://www.sciencedirect.com/science/article/pii/S0730725X13003810

[6] The ESPIRiT Code; this is a collection of Matlab functions and demos to reproduce some of the results that are described in the papers:
M. Lustig and JM Pauly "SPIRiT: iTerative Self-consistent Parallel Imaging Reconstruction from Arbitrary k-space", Magnetic Resonance in Medicine, 2010;64(2):457-71
T. Zhang, JM Pauly, SS Vasanawala and M. Lustig "Coil compression for accelerated imaging with Cartesian sampling" Magn Reson Med, 2013;69(2):571-82
D. Bahri, M. Uecker, M. Lustig, "ESPIRiT-Based Coil Compression for Cartesian Sampling" ISMRM 2013 pp:2657
M. Uecker, P. Lai, MJ Murphy, P. Virtue, M Elad, JM Pauly, SS Vasanawala and M Lustig, "ESPIRiT- an eigenvalue approach to autocalibrating parallel MRI: Where SENSE meets GRAPPA", Magn Reson Med, 2013
P. Shin et. al, "Calibrationless Parallel Imaging Reconstruction Based on Structured Low-Rank Matrix Completion" 2013, submitted to MRM.
http://people.eecs.berkeley.edu/~mlustig/Software.html

[7] The course of Dr. Alessandro Sbrizzi called; Scientific computing in the bachelor Mathematics.

[8] van der Heide O, Sbrizzi A, van den Berg CAT. Accelerated MR-STAT Reconstructions Using Sparse Hessian Approximations. IEEE Trans Med Imaging. 2020 Nov;39(11):3737-3748. doi: 10.1109/TMI.2020.3003893. Epub 2020 Oct 28. PMID: 32746119.

[9] The information found about Grappa:
https://mriquestions.com/grappaarc.html

[10] Alessandro Sbrizzi, Oscar van der Heide, Martijn Cloos, Annette van der Toorn, Hans Hoogduin, Peter R. Luijten, Cornelis A.T. van den Berg, Fast quantitative MRI as a nonlinear tomography problem, Magnetic Resonance Imaging, Volume 46, 2018, Pages 56-63, ISSN 0730-725X, https://doi.org/10.1016/j.mri.2017.10.015

References for figures:
Figure 1:
https://radiologykey.com/image-construction-part-i-slice-selection/

Figure 4:
https://compimag.org/projects/mrstat.html

A Code: SENSE algorithm in Julia

For programming Sense, the slides of Alessandro Sbrizzi from the course scientific computing are used, in references, source [7].

```
using FFTW
using PyPlot
pygui(true)
using ImagePhantoms
using FourierTools

## block 1: make original image + k space and aliased and undersampeled
image_size = 64
image = rotr90(shepp_logan(image_size, SheppLoganToft())) # the original image rotated
kspace = fftshift(fft(fftshift(image))) # kspace of the image
undersampled_kspace = copy(kspace)
undersampled_kspace[:,2:2:end] .= 0 # kspace undersampled
aliased_image = fftshift(fft(fftshift(undersampled_kspace))) # the aliased image

## block 2 and block 3: we create the 4 different coils and the coil sensitivity maps
# the vector: new_vector1 is a vector that goes from 1 to 0, below we create the coil_sen maps
for i in 1:image_size
    for j in 1:image_size
        coil_sen1[i,j] = coil_sen1[i,j]*new_vector1[i]
        coil_sen2[i,j] = coil_sen2[i,j]*new_vector1[j]
        coil_sen3[i,j] = coil_sen3[i,j]*new_vector1[image_size-i+1]
        coil_sen4[i,j] = coil_sen4[i,j]*new_vector1[image_size-j+1]
    end
end

## block 4: we create the under sampled images for the coils,
# apply DFT^-1 to them and obtain the folded images
# we do this for all 4 coils, but only one is shown below
# first undersampeled picture for coil 1
kspace1 = fftshift(fft(fftshift(image1))) # kspace of the image
undersampled_kspace1 = copy(kspace1)
undersampled_kspace1[:,2:2:end] .= 0 # the undersampled kspace
aliased_image1 = rot180(fftshift(fft(fftshift(undersampled_kspace1))))
# DFT^-1 applied, which gives us the folded image

## blok 5: reconstucting the images, with the information from previous parts.
# We determine matrix C (all the coil sensitivity maps)
C = cat(coil_sen1, coil_sen2, coil_sen3, coil_sen4, dims=3);
C = permutedims(C, (3, 2, 1));
# the same for the aliased images
aliased_images = cat(aliased_image1, aliased_image2, aliased_image3, aliased_image4, dims=3);
```

```
aliased_images = permutedims(aliased_images, (3,2,1));

R=2 # the reduction factor = amount of times the image is 'fold'
# we want to loop through all x (frequency-encoding) and y (phase-encoding) values
aliased_size = image_size/R
for x in 1:image_size
    for y in 1:image_size/R
        y = floor(Int8,y) # otherwise it will be a float
        C_nu = C[:, y:32:64, x]
        # we want to invert C_nu but it is not a square matrix, so we use the least square method
        C_inv = inv(transpose(C_nu)*C_nu)*transpose(C_nu)
        aliased_nu = aliased_images[:, y, x] # the voxel that is needed for this calculation
        eind_image1 = C_inv*aliased_nu
        # in Julia a shorter version can be used:
        eind_image1 = C_nu\aliased_nu
        # now we want to unfold
        eind_image[y:32:64, x] = eind_image1
    end
end
# eind_image is the unfold images, so the aliasing is undone with Sense
```

B Code: Mask based on L^2-norm in Matlab

The algorithm for making a mask based on the L^2-norm.

```
%% create the L^2-norm of all the coil maps
Sum =zeros(224,224); % array with zeros
for i = 1:8
    image_i = squeeze(images(:,:,i));
    % images are the ifftc2() of the k-spaces from 8 coils
    kwadraat = abs(image_i).^2;
    Sum = Sum + kwadraat;
end
Total = sqrt(Sum);
figure, image(abs(Total)); colormap(gray(256)); colorbar;

%% making the mask, based on Totaal
maxi = max(abs(Total),[],'all');
% a factor is used, to get only the a part of the pixels `on'
mask = abs(Total)> factor*maxi; % part that is on
mask2 = abs(Total)< factor*maxi; % part that is off
% apply it to Total
masked = repmat(Total,1);
masked(~mask)=0; % `on' part
masked(~mask2)=1; % `off' part

% showing the amount of pixels that is `on'
mask_numb = sum(masked,'all');
disp(mask_numb);
```

C The reconstructed coilmaps

In this appendix all the reconstructed coilmaps of the MR-STAT data are shown, with for each of those maps an image of the difference between the reconstruction and the real coil sensitivity map.

Figure 17: MR-STAT dataset 1

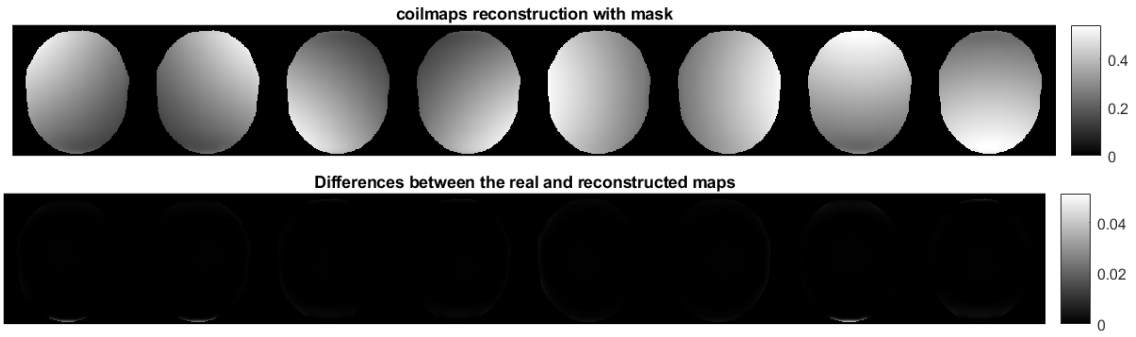


Figure 18: MR-STAT dataset 2

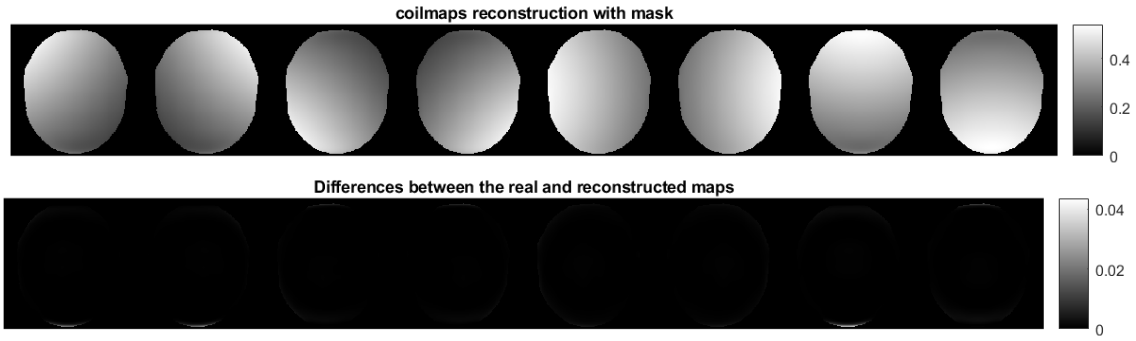


Figure 19: MR-STAT dataset 3

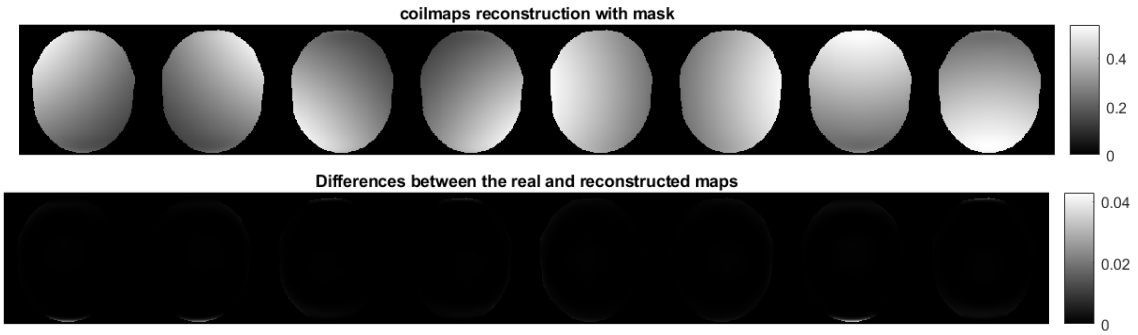


Figure 20: MR-STAT dataset 4

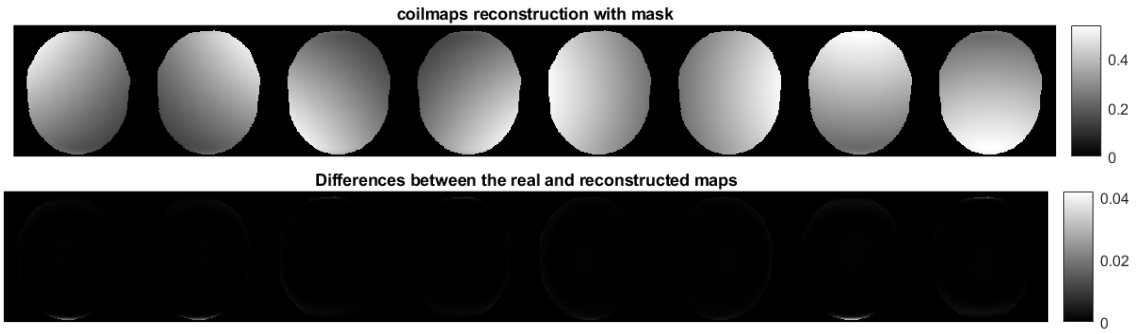
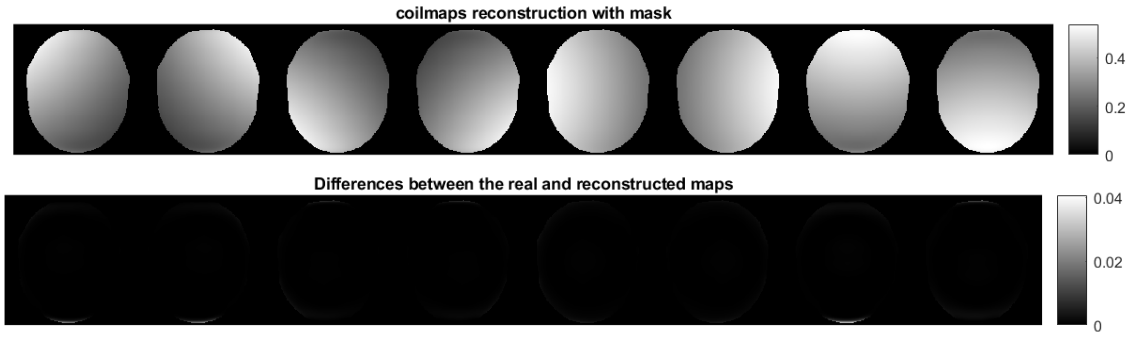


Figure 21: MR-STAT dataset 5



D The reconstructed coilmaps for the extra dataset

In this appendix the reconstructed coil maps and the differences between the real coil map are shown for the extra dataset.

Figure 22: MR-STAT extra dataset 1

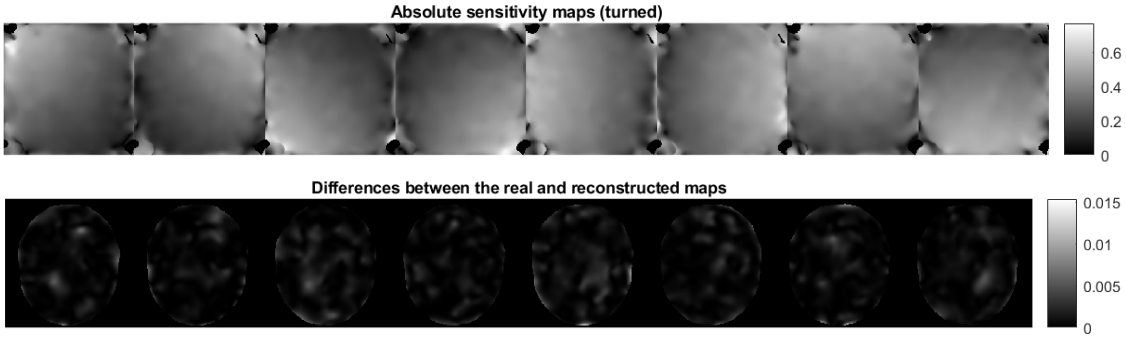


Figure 23: MR-STAT extra dataset 2

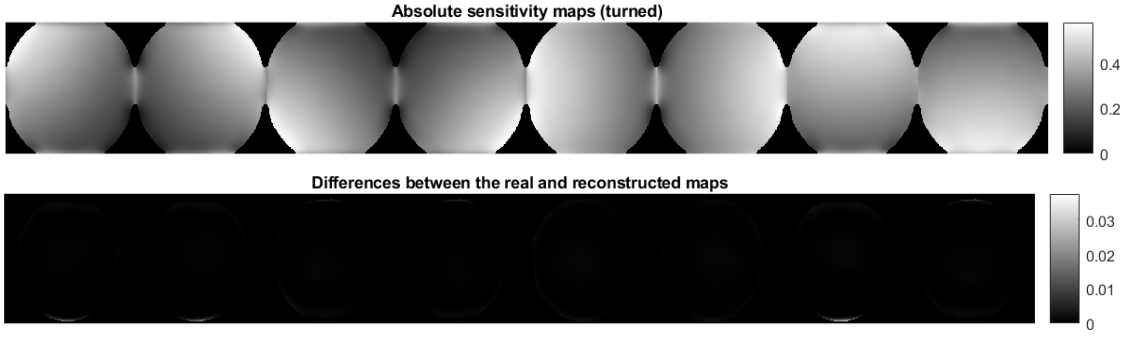


Figure 24: MR-STAT extra dataset 3

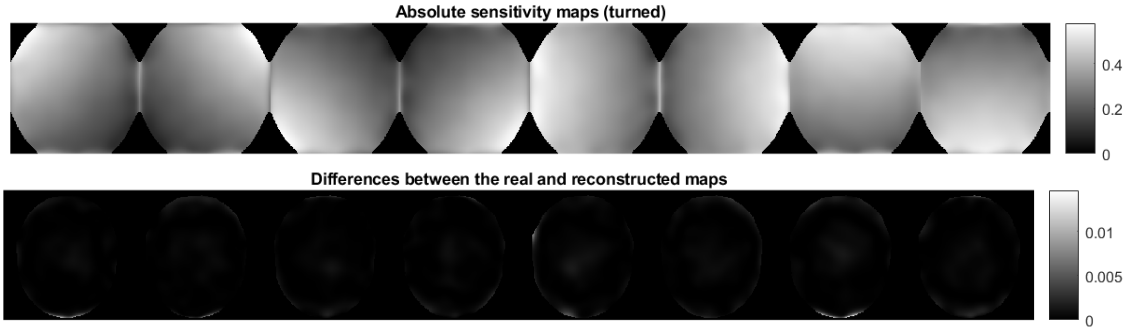


Figure 25: MR-STAT extra dataset 4

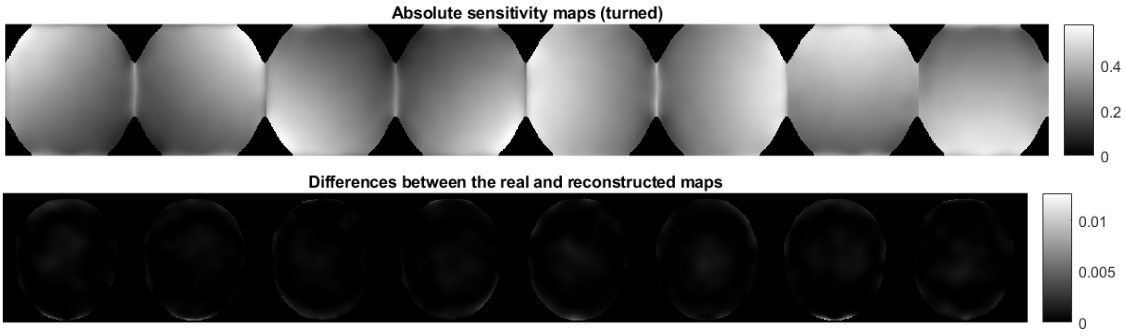


Figure 26: MR-STAT extra dataset 5