Technical University of Delft, Technical University Berlin and Philips Research

Institute of Mathematics
Research area Scientific Computing

Master Thesis

Sparsity promoting algorithms for multi-component magnetic resonance parameter mapping

Martijn Nagtegaal

November 15 2018

Supervised by
Prof. Dr. Reinhard Nabben (TU Berlin)

and
Dr. Mariya Doneva (Philips Research)

Philips Research
Röntgenstrasse 24
22335 Hamburg

This master thesis originated in cooperation with Philips Research Hamburg and is the final project for the master programs Scientific Computing at the Technical University of Berlin and Applied Mathematics at the Technical University of Delft. Together with the master Applied Mathematics at KTH Stockholm this forms the COSSE program.

First of all I would like to thank Mariya Doneva for giving me the opportunity to carry out this research, taking the time for many lively discussions, being able to answer almost all of my questions and guiding me in many different ways during my eight months at Philips. Also I want to thank Prof. Dr. Reinhard Nabben at the TU Berlin for his supervision.
I would like to thank Thomas Amthor as well for setting up a large part of the MR Fingerprinting framework and always being enthusiastic about new results. The time Burkhard Mädler took to come over to Hamburg and look into my results and discuss how this can be placed in the field of myelin mapping was something I appreciated a lot as well.
Furthermore I would like to thank my fellow students and colleagues at Philips for thinking along and all the shared coffee breaks. At last I would like to thank Dr. Martin van Gijzen from the TU Delft for having a look at my thesis and inspiring me with a course to look for a thesis subject on the mathematics behind MRI.

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Hamburg, November 15 2018

. . . . . .

Martijn Nagtegaal

Abstract

Magnetic resonance imaging (MRI) is often used to obtain qualitative images of the human brain. Different kinds of tissues can be recognised from the different levels of contrast. Multi-echo spin-echo T_2 MR measurements can be used to obtain a quantitative map of the brain where different tissues or components can be recognised based on a certain tissue property, the T_2 relaxation time this case. This quantitative approach is also called parameter mapping. Multi-component parameter mapping makes it possible to determine the concentrations of different tissues inside a certain region, this is mainly used for the determination of the myelin water fraction. Myelin is a substance present in low concentrations in the brain and can be related to certain neurodegenerative autoimmune diseases.

Magnetic resonance fingerprinting (MRF) provides a new way to such qualitative measurements. An advantage is that MRF is sensitive to both T_1 and T_2 relaxation times, making it possible to distinguish more types of tissues. The drawback is, however, that more combinations of the parameters are possible, leading to a more difficult problem with longer computation times.

In this project a new method to perform a multi-component analysis in MRF and multi-echo spin-echo T_2 measurements is developed, which is faster and less sensitive to noise than previous methods. The algorithm that is developed in this project, is called Sparsity Promoting Iterative Joint Non-negative least squares (SPIJN). It finds a small number of components throughout the region of interest without assumptions about the number of components or their relaxation times by imposing a joint sparsity constraint. This new method is compared to previously published methods in both numerical simulations and in vivo experiments. The multi-component decomposition for brain data results in meaningful structures on first sight, although further research would be required on the meaning of the matched components. Moreover, the algorithm has been used for the calculation of the myelin water fraction from multi-echo spin-echo T_2 data. The obtained maps are comparable to state-of-the-art methods, show improvements in some cases and have significantly shorter computation times.

Zusammenfassung

Magnetresonanztomographie (MRT) wird häufig verwendet um qualitative Bilder des menschlichen Gehirns zu machen. Verschiedene Gewebearten kann man durch die verschiedenen Helligkeitsstufen unterscheiden. Multi-Echo Spin-Echo T_2 MRT Messungen können verwendet werden um bestimmte Gewebearten zu identifizieren, in diesem Fall die T_2 Relaxationszeit. Dieser quantitative Ansatz wird "parameter mapping" genannt. Mehrkomponenten parameter mapping ermöglicht es die Konzentration von verschiedenen Geweben in einer Region zu bestimmen. Dies wird hauptsächlich dazu verwendet um den Myelin-Wasseranteil zu bestimmen. Myeline ist eine Substanz die in niedrigen Konzentrationen im Gehirn vorkommt und relevant für die Diagnose von bestimmten neurodegenerativen Autoimmunerkrankungen ist.
Magnetresonanz-Fingerprinting (MRF) ist eine neue Methode um solche qualitativen Messungen durchzuführen. Ein Vorteil von MRF ist, dass sowohl T_1 und T_2 Relatationszeiten betracht werden und so mehr verschiedene Gewebearten unterschieden werden können. Ein Problem dabei sind allerdings die komplexeren und länger dauernden Berechnungen, durch die erhöhte Anzahl an Kombinationen der Parameter.
In diesem Projekt wird eine neue Methode für diese Mehrkomponentenanalyse bei MRF und Multi-Echo Spin-Echo T_2 Messungen entwickelt, die schneller ist und weniger rauschempfindlich als bisherige Methoden. Der Algorithmus der in diesem Projekt entwickelt wurde heißt "Sparsity Promoting Iterative Joint Non-negative least squares" (SPIJN). Durch die Verwendung einer gemeinsamen Sparsity-Nebenbedingung findet er eine kleine Anzahl an Komponenten im Interessenbereich ohne Annahmen über die Anzahl der Komponenten oder deren Relaxationszeiten zu machen. Die vorgeschlagene Methode wird mit bestehenden Methoden in numerischen Simulationen und in vivo Experimenten verglichen. Die Multikomponenten-Dekomposition von MRF-Daten erzeugt auf den ersten Blick plausible Strukturen, deren Interpretation weiterer Forschungsarbeit bedarf. Der Algorithmus wurde weiterhin verwendet, um den Myelin-Wasseranteil aus Multi-Echo Spin-Echo T_2 zu berechnen. Die erzeugten Bilder werden mit den Ergebnissen der aktuellen Vergleichsmethoden verglichen und zeigen dabei Verbesserungen bei deutlich reduzierten Berechnungszeiten.

Samenvatting

Magnetic resonance imaging wordt vaak gebruikt om kwalitatieve afbeeldingen van de hersenen te maken, waarbij verschillende soorten weefsels verschillende contrasten laten zien. Multi-echo spin-echo T_2 metingen kunnen gebruikt worden om een kwalitatieve weergave van de hersenen te verkrijgen, waarbij de verschillende soorten weefsel herkend kunnen worden op basis van een bepaalde weefsel eigenschap, de T_2-relaxatietijd in dit geval. Deze kwalitatieve aanpak wordt ook wel parameter mapping genoemd. Een meerdere componenten parameter mapping maakt het mogelijk om de concentratie van verschillende weefsels in een bepaald gebied te bepalen. Dit wordt voornamelijk gebruikt voor het bepalen van de myelinewaterfractie. myeline is een stofje dat in lage concentraties voorkomt in de hersenen en dat gerelateerd is aan bepaalde neurodegeneratieve auto-immuunziekten.
MR Fingerprinting (MRF) biedt een nieuwe mogelijkheid om een dergelijke meerdere componenten analyse uit te voeren. Een voordeel is dat MRF zowel gevoelig is voor T_1 als T_2 relaxatie tijden en daardoor meer soorten weefsel kan onderscheiden. Het nadeel is echter dat meer parameter combinaties mogelijk zijn, waardoor het op te lossen probleem lastiger wordt, met langere rekentijden als gevolg.
In dit project is een nieuwe methode om een meerdere componenten analyse uit te voeren voor MRF en multi-echo spin-echo T_2 metingen geïntroduceerd. De methode is sneller en minder gevoelig voor ruis dan eerder voorgestelde methodes. Het algoritme ontwikkelt in dit project heet Sparsity Promoting Iterative Joint Non-negative least squares (SPIJN). Het algoritme zoekt een klein aantal componenten in het interessegebied, zonder verdere aannames over het aantal of de relaxatietijden van de componenten, door het opleggen van een gezamenlijke ijlheidsbeperking. In nummerieke simulaties en in vivo metingen is de voorgestelde aanpak vergeleken met eerder gepubliceerde methoden. De decompositie voor de MRF hersenmetingen laat enkele herkenbare structuren zien, alhoewel meer onderzoek nodig is om een zekere mate van toeval hierbij uit te sluiten. Het algoritme is ook toegepast voor het berekenen van de myelinewaterfractie op basis van T_2 relaxatie metingen. De verkregen afbeeldingen zijn vergelijkbaar met gebruikelijke methoden, laten in sommige gevallen verbeterde resultaten zien en de rekentijden zijn aanzienlijk korter dan met de gebruikelijke algoritmes.

Contents

1. Introduction

This chapter gives an overview of the general setting of this thesis. Starting with the motivation and the general problem which is attempted to solve, followed by the objective this leads to, finished by an outline of the following chapters.

1.1. Motivation

The main objective of diagnostic imaging is to gain information about certain structures in the body and visualise this information as an interpretable image. Magnetic resonance imaging (MRI) offers a non-invasive method to produce two- or three-dimensional images of the body. Compared to computed tomography, MRI has a better soft tissue contrast, which makes it more applicable for usage in the brain or on other organs.
Regular MR images show different contrasts for different tissues, which is then used to distinguish the different tissues. In a medical setting one tries to determine abnormalities from these images. These images thus only offer a qualitative way to gain information on the tissues.
One method which provides more quantitative information on the tissue types present in the studied part of the body is MR fingerprinting (MRF). MRF measures a signal for a certain area and then tries to recognise the tissue type which could be emitting such a signal. Such a volume is called a voxel with a typical size of $1 \times 1 \times 3$ mm^3.
Typically a voxel contains more than one tissue, which leads to the partial volume effect. The measured signal for the voxel is then formed by the mixture of the tissues present in the volume. More information could thus be gained if it would be possible to recognise the presence of multiple components (or tissues) present in one voxel and to indicate the fractions in which these are present. Developing and testing effective and efficient methods to do so will be the main goal of this thesis.
Being able to recognise the different tissues can be useful in several settings. In the brain the axons of some nerve cells are surrounded by a fatty substance called myelin. Myelin is related to brain development and essential for the functioning of the nerve system and therefore the loss of myelin (demyelination) can be related to certain neurodegenerative autoimmune diseases. Since myelin is never the mainly present component, it can only be "seen" with multi-component methods. Being able to quantify the amount of myelin, could lead to clinical improvements and earlier diagnoses. MR measurements are considered to be sensitive for the water surrounding the myelin and therefore it is possible to determine the myelin water fraction (MWF) from certain MR measurements [1, 2, 3, 4].
A second application in the brain would be the ability to get more information about the border zone between grey and white matter, for example cortical dysplasia causes a slight

grey- white matter blurring. A third clinical application is in the field of prostate cancer, making it possible to get more insight how a tumour interacts with the surrounding tissue. Aside from these examples there could be applications in muscles and more fields. Multi-echo spin-echo T_2 measurements, also known as T_2 relaxation measurements, measures the relaxation of the transverse magnetisation for a certain time. The problem setting to perform a T_2 multi-component analysis based on this data, is comparable to the MRF setting and for the kinds of applications just mentioned multi-echo spin-echo T_2 measurements are currently used. Since MRF is sensitive to a combination of transverse (T_2) and longitudinal (T_1) relaxation times, MRF would be able to recognise more tissues, which would be necessary to distinguish white and grey matter.

1.2. Objective

The combination of MR fingerprinting and multi-component analysis is relatively new, but some first approaches have been published [5, 6]. In these papers two methods have been investigated to perform a multi-component analysis. An exhaustive analysis of the different possible methods to perform a multi component analysis will be given in this thesis. In the previous papers a voxel-by-voxel approach is used, gaining no advantage of the information contained by the other voxels.
The current methods have relatively long computation times. With computation times of 1 to 10 seconds for a single voxel, the multi-component analysis will take hours for a single slice and is therefore not practically applicable. A second drawback of the published methods is that their results are hard to interpret. The resulting multi-component matching uses a few components for each voxel, but hundreds for a complete slice of the human brain. This makes it very hard to interpret the results and therefore requires smart grouping strategies [7], which requires extra assumptions about the present tissues and adds extra processing steps and time.
In this project a joint sparsity constraint will be introduced and algorithms to efficiently solve the resulting minimisation problem will be developed. To do so, insights from voxel-wise methods and joint-sparsity algorithms will be combined and where necessary adapted to our problem setting which requires non-negative solutions and is based on highly correlated dictionaries. The introduction of the joint sparsity constraint should lead to easier interpretable results, without introducing an extra bias.
Multiple component analyses have been performed in many different ways for T_2 relaxation [8, 9, 10, 11, 12], these methods have been investigated for a long time. It is therefore useful to look into the methods used here and how these can be used in the MRF case. The current state-of-the-art methods for multi-component T_2 mapping have high computation times as well, especially when compensation for B_1 inhomogeneities are included. Computation times of half an hour per slice can be considered as a bottleneck for clinical applications. Since the multi-echo spin-echo T_2 multi-component analysis is closely related to MRF multi-component analysis the joint sparsity algorithms can be applied to these measurements as well to compare the developed algorithm in a more

mature field.
This project will use a combination of simulations and in vivo measurements to test the current and proposed methods. Beside this, the application to undersampled measurements will be considered. In the single component problem setting MRF is very robust against the artefacts caused by undersampling. If the multiple component methods are also robust against undersampling the measurements can be performed faster, which makes clinical usage more likely. The developed methods will be applied to the field of T_2 multi-component analysis and myelin water mapping as well.
The aim of this work is thus to find a fast, stable algorithm to perform a multi-component analysis with MRF and multi-echo spin-echo T_2 measurements, with results which are easy to interpret. The way to reach this goal would be by introducing a joint sparsity constraint. A clear application of such an algorithm is in the field of myelin water mapping, but there are also potentially more applications like prostate T_2 mapping, which would benefit from a fast and reliable multi-component analysis method.

1.3. Outline

Chapter 2 gives an introduction to MRI and will build up to the problem setting which is considered in this project. The introduction to MRI will aim at introducing the necessary concepts to understand the main topics of this project. To make it easier to interpret the results as presented later, a short description about the human brain in MRI is included. This is followed by the introduction of MR relaxometry, enabling the mapping of T_1, $T_1\rho$, T_2 or T_2^*, and exponential T_2 multi-component analysis. After discussing these established methods, the main concepts of MR Fingerprinting are introduced, which can then be combined with multi-component analysis. This leads to several minimisation problems which are studied in the rest of the thesis. In the last section specific T_2 mapping methods are named to give an overview of this field, without discussing them in detail.
In Chapter 3 the different numerical methods used in this thesis are discussed in detail. It starts with some methods to pre-process the data. This is followed by the voxel-by-voxel algorithms which will be discussed in detail since they will often be used for comparisons and insights from these methods were used in the rest of the thesis.
Section 3.3 is dedicated to different algorithms which incorporate the information of all the voxels by introducing the joint sparsity constraint. This section could therefore be considered as the core of this thesis. In this section two new algorithms are proposed: the Sparsity Promoting Iterative Joint Non-negative least squares (SPIJN) and Joint Regularised Non-Negative Orthogonal Matching Pursuit (JRNNOMP). In this joint approach several modifications are considered such as a segmentation, based on the mainly present tissue types followed by a regrouping and a relaxation of the number of components, by selecting better similar combinations of relaxation times.
After discussing several possible methods, these are investigated in detail in Chapter 4. It starts with a comparison between several voxel-by-voxel methods and the proposed joint sparsity approaches in a numerical simulation including two components. Afterwards

simulations are performed with three components where one component is intended to represent myelin water.
The most promising methods are then applied to a phantom and in vivo measurement with and without undersampling for MRF measurements. Beside that the results are shown for the SPIJN algorithm applied to multi-echo spin-echo T_2 measurements and compared to a state-of-the-art method for myelin water mapping.
In Chapter 5 first the different voxel-by-voxel methods are discussed in detail, followed up by the joint sparsity methods where most attention is paid to the SPIJN algorithm. This is followed by a more general discussion of the effects of the assumption of a smooth T_2 spectrum compared to the introduction of the joint sparsity constraint.
The last chapter gives the conclusions of this project combined with a short recommendations on which next steps could be made.

2. Theory and related work

This chapter will give an overview of the necessary theory and relevant related work. It starts with the basic principles of Magnetic Resonance Imaging (MRI) in Section 2.1, addressing the most important concepts. After the introduction of these basic concepts, Section 2.2 is dedicated to MRI in combination with the human brain. This is followed by the introduction of MR parameter mapping and multi-component analysis (Sections 2.3 and 2.4). After this Magnetic Resonance Fingerprinting (MRF) is introduced in Section 2.5and how it provides a quantitative mapping, focusing on the parts of MRF which are relevant later on. In Section 2.6 a mathematical problem setting of multi-component analysis is given, in the single pixel case as well as in the case where all pixels in the image are considered at once .
The chapter ends with Section 2.7 giving an overview of current methods used for T_2 multi-component analysis and a few words on MRF multi-component analysis methods not addressed in this project.

2.1. Magnetic Resonance Imaging

Magnetic resonance imaging is based on the alignment of the hydrogen nuclei (protons) in the body within a strong magnetic field and the possibility to excite these protons to a higher energy state. MRI measures how the hydrogen nuclei slowly return to their equilibrium state, parallel to the magnetic field. This section will give a short introduction with the important subjects necessary to understand the work done in this project. No attempt to give a complete overview has been made here, a more elaborate description can be found in e.g. [13].

2.1.1. Magnetic fields and longitudinal and transverse relaxation

The human body is mainly composed of water molecules. These water molecules consist of two hydrogen nuclei (protons) and one oxygen atom. The hydrogen nuclei act as little magnets when placed in a magnetic field and just as the needle of the compass they will try to align with the magnetic field. When the human body is placed in a strong magnetic field B_0 (typically 1 to 7 Tesla) a small part of the protons present in the body align with the field. Placing the body in such a strong magnetic field does not do any harm to the tissue, which is an advantage of MRI over other imaging methods.
When the protons are aligned to the magnetic field a net magnetisation M_0 in the direction of the field is formed. It is possible to apply an extra magnetic field B_1, in a perpendicular direction to the main magnetic field. If this extra field is only turned on

for a short time, it can be seen as a radio-frequency (RF) pulse. One can choose the ω frequency of a RF-pulse. The protons with this resonance frequency ω are excited to a state which is not aligned with the main magnetic field any more. This so called Larmor frequency ω is linearly depending on the strength of the main magnetic field B_0:

$$\omega = \gamma B_0, \tag{2.1}$$

where γ is the gyromagnetic ratio. Before the RF pulse the net magnetisation of the protons is in the z-direction, the RF pulse tips the net magnetisation away from the z-axis with a certain flip angle α. Since the RF pulse is a electromagnetic pulse, it is a magnetic field as well, which can be denoted as B_1.
The time it takes to return to the equilibrium state is related to the longitudinal relaxation time T_1, this exponential behaviour can be described as

$$M_z(t) = M_0 - (M_0 - M_z(0))e^{-t/T_1} \tag{2.2}$$

where $M_z(t)$ is the magnetisation in the z direction over time and M_0 the magnetisation in the equilibrium state. The value of this longitudinal relaxation time T_1 is different for different tissues, since the molecules surrounding the water molecules influence the relaxation time. Water molecules which are able to freely flow around have very long relaxation times, water molecules surrounded by fat molecules have very short relaxation times in general. It is good to mention that the value of T_1 also depends on the strength of the magnetic field.
When the RF pulse has just been applied the nuclear spins of the protons are rotating nicely in sync and together they have a magnetisation in the xy plane $M_{xy}(0)$. At the beginning all the spins thus have the same phase and they rotate with the same frequency. However, after some time the proton spins start to get out of phase, leading to a decay in the M_{xy} magnetisation. This decay also behaves exponentially

$$M_{xy}(t) = M_{xy}(0)e^{-t/T_2}, \tag{2.3}$$

where T_2 is the transverse relaxation time. This transverse relaxation time depends on the type of tissue and the molecules surrounding the water molecules, just as the longitudinal relaxation time.
In any case the restriction holds that $T_2 \leq 2T_1$ [14], but in almost all cases T_1 is greater than T_2 and typically T_1 is a order of 10 larger than T_2. In the beginning of a measurement typically T_2 is dominant and when the transverse magnetisation has disappeared T_1 becomes increasingly important.
Unless explicitly stated differently the T_1 and T_2 relaxation times are in milliseconds.

2.1.2. Gradient and spin echo, flip angle, echo time and repetition time

The RF-pulse as mentioned in the previous section pushes the net magnetisation out of the equilibrium state away from the z-axis with a certain angle. This angle with which the proton spins are rotated, is called the flip angle.

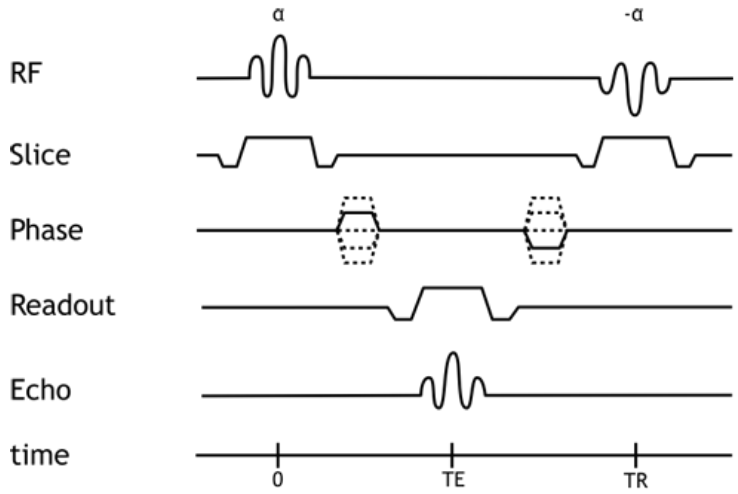


Figure 2.1.: The pulse sequence diagram for a gradient-echo sequence. RF is the used RF pulse, Slice is the signal used to select the slice of interest, phase is used for the phase encoding and readout corresponds to the gradient signal and Echo shows the effective signal. The diagram is from [15]

The MR scanner can only measure the magnetisation in the transverse plane, however, this magnetisation decays very rapidly with T_2^*. To be possible to measure the magnetisation, the transverse magnetisation needs to be refocused forming an echo. This can either be a gradient echo or a spin echo. The echo time TE is the moment at which the magnetisation is refocused again or an echo occurs.
A gradient echo uses magnetic field gradients to refocus the magnetisation. The pulse sequence to perform such a gradient-echo is shown in Figure 2.1. The refocused magnetisation is only available for a short time during which only a small amount of data can be acquired. Therefore the pulse sequence in Figure 2.1 usually needs to be repeated multiple time to obtain a whole image. The repetition time TR is illustrated as well, which is the time between applying two successive RF pulses.

In a spin echo pulse sequence, initially a 90 degree pulse is applied to tip the magnetisation to the transverse place, after which a 180 degrees pulse is applied at time $TE/2$, which changes the direction of the precision leading to refocusing in the signal (a spin echo) at time TE. A pulse sequence diagram for a spin-echo scheme is shown in Figure 2.2. The spin echo pulse reverts the dephasing because of B0 field inhomogeneity effects and creates an echo after time TE from the peak of the RF pulse.

It is also possible to measure the signal several times by sending multiple echo pulses, possibly preceded by an RF pulse corresponding to a certain flip angle, leading to a sequence of measurements. When several echos are created in the spin-echo scheme with only one 90 degree RF pulse, a multi-echo spin-echo measurement is performed. The normal choice is a constant echoing time, resulting in measurement of the measurement of the exponentially decaying M_{xy} magnetisation with the relaxation time T_2 with constant

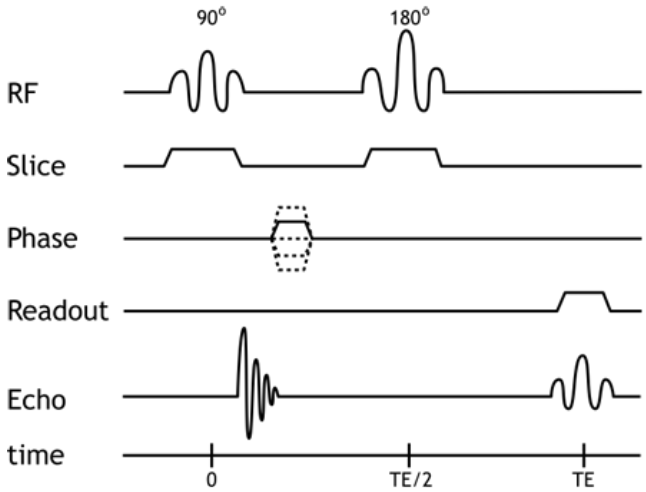


Figure 2.2.: The pulse sequence diagram for a spin-echo sequence. RF is the used RF pulse, Slice is the signal used to select the slice of interest, phase is used for the phase encoding and readout corresponds to the gradient signal and Echo shows the effective signal. The signal decays exponentially with T_2. The diagram is from [15]

time intervals between the measurements.
A situation in which the RF-pulse as represented by the B_1-field is not completely homogeneous can occur. B_1 inhomogeneity arises because the RF power is absorbed differently across the patient, due to the changing permittivity and conductivity of tissues and standing waves in tissues. This causes a variable flip angle across the patient. This will be called the flip angle inhomogeneity (FAI) and is expressed as the ratio between the effective flip angle and the intended flip angle. Since these inhomogeneities lead to an RF-pulse which is not exactly at the resonance frequency, this is also called off-resonance compensation.

2.1.3. Measuring a signal, weighting, k-space and undersampling

The measurement of a signal is based on the electric signal which is induced in a coil when the magnetisation changes. The change in the magnetic field, which is caused by the relaxation of the proton spins, creates an electrical signal in a coil dictated by Faraday's law, this effect is known as induction and can easily be measured. This measured signal can then be related to the magnetisation $M = \sqrt{M_z^2 + M_{x,y}^2}$ of the protons.
This measured signal gives information about the state of the protons at the moment of the measurement. The behaviour of the protons is, as described earlier, depending on T_1 and T_2. As can be seen in equations 2.2 and 2.3 the measurement time t, which corresponds to TR and TE respectively, the initial value of $M_z(0)$ and $M_{xy}(0)$ determine

the outcome of the measurement and the respective influence of T_1 and T_2. When a large t is used, the resulting image will mainly be determined by the T_1 value and is thus called T_1-weighted. When a small t is used and approximately $M_z(0) = 0$ the resulting image will mainly be determined by the T_2 value and is thus T_2 weighted.
When no further measures are taken in the measurement process, the protons in the entire measured area would be excited and influencing the signal, making it impossible to recognise different areas. The measured signal would be an average of the whole volume and would not be spatially resolved. To be able to select certain volumes, methods as slice selection, phase encoding and frequency encoding are used. In these methods the magnetic field changes with a certain gradient throughout the area, making it possible to selectively excite certain areas corresponding to their Larmor frequency ω. After the slice encoding used during the excitation pulse, frequency and phase encoding are used to measure only a certain (x, y) location in the slice.
So instead of measuring the total magnetisation of the area over time, one tries to measure the magnetisation of certain areas by using the afore mentioned methods. As a result of this, one does not directly measure the magnetisation $M(x, y)$ in the area of interest but the 2D Fourier transform of the signal, which can then be transformed back to the magnetisation $M(x, y)$. This measured 2D Fourier transform is measured in the so-called k-space. The k-space can be considered as a grid with the same size as the image. To be able to fully reconstruct the image, the full k-space needs to be sampled. This can be done in several ways, for example in a Cartesian manner or using a spiral or radial trajectories.
One way to speed-up the measurements is by using undersampling [16], which will be explained here briefly. To reproduce a signal of N time-points, one needs a Fourier transformed signal of N time-points as well, which is known as the Nyquist–Shannon sampling theorem. The low frequencies give the general shape of the signal and the higher frequencies represent the details of the signal and by removing a part of the Fourier transform certain details will be lost. To fully reconstruct the image the same number of points in k-space is needed as the number of pixels. When one only measures a part of the k-space, the k-space is called undersampled. The inverse Fourier transform can still be used to obtain an image based on the undersampled data, the image will, however, contain aliasing artefacts because of the undersampling.

2.2. The human brain in MRI

As mentioned in the introduction, the different tissues in the brain play a central role in this thesis. In this section a short background about the different tissues recognised in the brain will be given, including their relaxation times.
In MRI typically 3 main components (tissues/fluids) can be recognised from T_1- or T_2-weighted images. Figure 2.3 shows a brain measurement, with the different parts annotated as follows; 1) grey matter, 2) white matter and 3) cerebrospinal fluid (CSF). Grey matter is typically located at the outer parts of the brain and white matter more to the centre, the centre of the brain is however formed by the CSF. An important

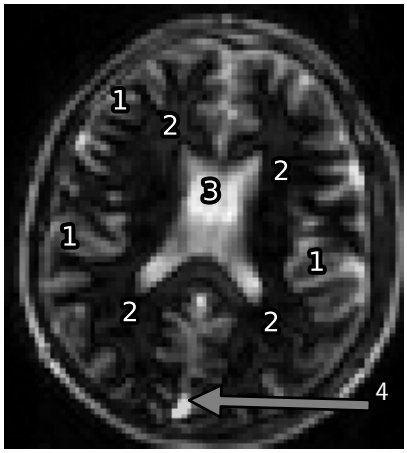


Figure 2.3.: A brain image from an MRF measurement to show the different tissues. Grey matter is annotated with 1, white matter with 2 and CSF with 3.

difference between white and grey matter is the presence of myelin in the white matter. Myelin is a fatty substance surrounding nerve cell axons [17], which plays a role in the communication between nerve cells. Since myelin is never the main component present in a certain area, it is hard to detect.
In Figure 2.3 a vessel is annotated with 4, because of the blood flow different behaviour is observed in this region. As explained in the previous section MR imaging is sensitive to protons or water present in the field of view (FOV). Typically three groups of water are recognised in the brain: free water which has very high relaxation times of several seconds since the water molecules are not bound, intra- and extra-cellular water which is the water in and surrounding the cells and as a third myelin water (MW), the water surrounding the myelin. Intra- and extra-cellular water is found in both grey and white matter. Since the myelin water is bound to the fatty myelin, the relaxation times of myelin are assumed to be short (10-15ms).
Table 2.1 gives an overview of relaxation times in the brain from the database of tissue properties [18] based on published measurements. This overview mainly contains white and grey matter relaxation times. In general white matter has shorter relaxation times than grey matter. The relaxation times as mentioned here were not taking multi-component effects into account, which means that the white matter included myelin, possibly leading to lower relaxation times. Since myelin can not be measured in a pure form, its relaxation times are difficult to measure. In general $T_2 \leq 40$ ms is considered to belong to myelin water. An estimate of the relaxation times of myelin at 1.5T are given in [19] as $T_1 = 63$ms and $T_2 = 13$ms.
The myelin water fraction (MWF) gives the fraction of myelin water compared to the total amount of water in a certain area. An overview of techniques to determine this

Table 2.1.: An overview of relaxation times [ms] from [18]. The tables include the number of literature values taken into consideration.

(a) T_1 relaxation times at 1.5T.

	Average	Std	# measurement	Min	Max
Grey matter	1260.82	178.4	8	1096.5	1531
White matter	999.85	443.3	7	612	1587
CSF	4627.25	788.1	2	4070	5185

(b) T_1 relaxation times at 3.0T.

	Average	Std	# measurement	Min	Max
Grey matter	1433.2	186.0	5	1275	1763
White matter	866.9	95.3	5	812	1110

(c) T_2 relaxation times at 1.5T.

	Average	Std	# measurement	Min	Max
Grey matter	109.4	52.9	4	77.9	217
White matter	112.3	82.9	4	66.2	282

(d) T_2 relaxation times at 3.0T.

	Average	Std	# measurement	Min	Max
Grey matter	92.6	16.9	5	66	110
White matter	60.8	13.1	4	49.5	79.6

are given in [20]. The measured cerebral white matter MWF is typically around 10% for a healthy subject. In [20] an overview of MWF values is given, ranging from 6.9% to 15.6% (and one outlier of 29.5%) for 1.5T and 10.2% to 13.4% for 3.0 T.
The human brain is very interesting, but many things are still unknown, therefore it can be easier to test new ideas on a more predictable object. Such an object is called a phantom. This phantom consists of known structures and the measurement outcome is therefore predictable and easier to verify. The phantom can be very similar to a human head or have a simpler shape. It is also possible to use a numerical phantom, making comparisons even more flexible. Numerical phantoms similar to a brain are available, but do normally not contain myelin water.

2.3. Magnetic resonance relaxometry imaging

As explained in the previous section conventional MRI offers a way to get a qualitative image where the image can be weighted for either T_1 or T_2 or a combination of the two, to make it possible to recognise which different tissues are present in the image by looking for intensity differences caused by their difference in T_1 and T_2. The information obtained is thus qualitative on a voxel level.
After the pulse which excites the tissue, the magnetisation intensity decays exponentially

over time. This exponential decay can be measured by measuring the magnetisation intensity at different moments in time. Depending on the measurement type, the exponential T_1, T_2^* or T_2 decay can be measured. This idea forms the principle of magnetic resonance relaxometry imaging [21, 22], also called parameter mapping. The used measurement type for T_2 is called a multi-echo spin-echo measurement.
The measured exponential decay is fitted to an exponential curve $\exp(-t/T_2)$, where the T_2-value which represents the exponential decay the best is determined. By measuring this exponential decay for all voxels over time it is possible to acquire several time signals which can each be matched to a certain T_2 value, giving a way to qualitative distinguish different tissues.
When this qualitative parameter mapping is performed it becomes more important to take certain effects such as the B_1 or FAI inhomogeneities into account, as shortly mentioned in Section 2.1.2. For multi-echo spin-echo measurements B_1 inhomogeneities cause deviation from the exponential decay, which can lead to large errors.

2.4. Exponential multi-component analysis

In the previous section the idea of T_2 mapping has been introduced, where a mono-exponential signal is fitted to the measured multi-echo spin-echo signal. This approach assumes that only one tissue is present in each voxel. This is, however, often not the case [23] and it is therefore more realistic to assume that multiple exponential curves are present in one signal and thus possibly multiple tissues are present in one voxel. Therefore a multi-exponential fit is performed [8, 24] determining a spectrum of relaxation times. Determining a spectrum of relaxation times can be relevant in many physical applications as described in [25]. To be able to discuss different methods in the next chapter, a more theoretical setting will be given.
When the signal $x(t)$ is assumed to be formed by a combination of exponentially decaying curves with some noise ϵ, the signal can be described by

$$x(t_i) = x_i = \int_{T_{\min}}^{T_{\max}} c(T) \exp(-t_i/T) dT + \epsilon_i \ , i = 1, 2, ...M, \tag{2.4}$$

where M data points x_i are measured at M different times t_i and $c(T)$ is the (unknown) distribution of relaxation times T_2 [8]. The spectrum of $c(T)$ can either consist of several delta peaks or be a smooth distribution.
To discretize the system, it is assumed that the system can be written in the following form:

$$c(T) = \sum_{j=1}^{N} c_j \delta(T - T_j), \tag{2.5}$$

where T_j are different relaxation times distributed over the interval $[T_{\min}, T_{\max}]$. It is then possible to write eq. (2.4) as

$$x_i = \sum_{j=1}^{N} c_j \exp(-t_i/T_j). \tag{2.6}$$

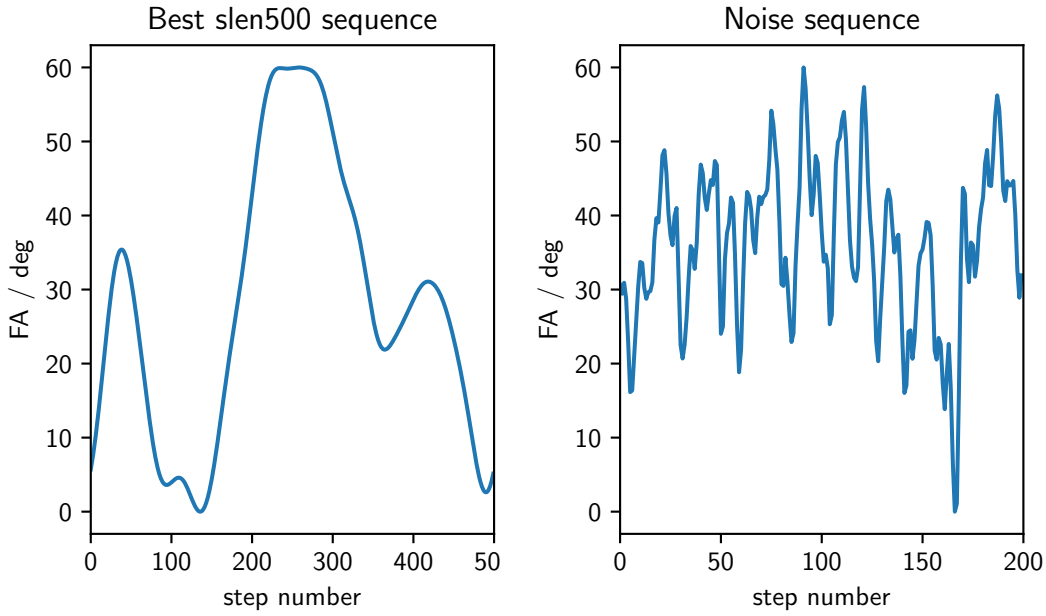
This formulation can be rewritten as the matrix vector product $\mathbf{x} = D\mathbf{c} + \epsilon$, where D is a matrix containing the signals as columns, thus with elements $D_{ij} = \exp(-t_i/T_j)$. This matrix is called a dictionary and can also contain different types of signals, as will be discussed later in the case of MR Fingerprinting. The vectors $\mathbf{x}$ and $\mathbf{c}$ contain the measured signal and the weights for each component respectively. The goal is thus to find a vector $\mathbf{c}$ which approaches the original distribution $c(T)$ as correct as possible. The different methods used to find a solution for this problem are discussed in Chapter 3. Since the exponential signals for similar relaxation times are very hard to distinguish, the demands on the initial signal to noise ratio (iSNR or SNR) is rather high and lies between 500 and 1000 for conventional methods [26], where the iSNR is defined as the signal to noise ratio in the first echo.

Multi-component analysis and partial volume effect

With the introduction of multi-component analysis it is good to explain the difference between the partial volume effect and the multi-component problem. The partial volume effect means that the voxel which is taken into consideration is a combination of two different volumes or tissues with a clear border between them. In a simplistic case the left part of the voxel could contain white matter and the right part grey matter. It is thus not considered that the different tissues are mixed.
Multi-component analysis covers a broader setting than the partial volume effect and includes the case where two tissues are mixed, for example myelin and white matter or a gradual transition between tissues.

2.5. Magnetic Resonance Fingerprinting

Magnetic Resonance Fingerprinting (MRF) has been introduced in 2013 by Ma et al. [27]. MRF provides a way to obtain quantitative information about the tissues or materials and their properties present in the scanned region. The aim is to identify the properties of each pixel or voxel present in the scanned region by identifying it as a certain combination of tissue properties. MRF is an approach to quantitative MRI, where a multi-parameter mapping is performed by measuring the signal evolution as a response to a pulse sequence with constantly varying acquisition parameters in a transient state measurement.
Instead of performing one measurement with a certain combination of flip angle and repetition time as in regular MRI, a series of measurements is taken in time, with varying acquisition parameters, i.e. flip angles and/or repetition times. The idea in this method is that each tissue or material reacts in a different way to this sequence of measurement parameters. This sequence of measurement settings should be chosen in a way that each tissue reacts as different as possible. Two flip angles sequences which will be used in this project are shown in Figure 2.4.
Performing an MRF measurement using a certain MRF sequence thus results in a set of images taken at different times with different measurement parameters. Typically a few hundred images are taken and for each voxel a signal can be obtained from the series



(a) A sequence of length 500, the phase angle is constant 0°.

(b) A sequence of length 200, the phase angle is alternating between 0° and 180°.

Figure 2.4.: Two possible MRF sequences of length 500 and 200. A repetition time of 15 ms has been used. The sequences will be referenced as MRF500 and MRF200.

of images. Each tissue can be characterised by its tissue properties as the longitudinal relaxation time T_1, the transverse relaxation time T_2 and other properties as diffusion, which are not considered in the parameter sets used here. Combined with the system parameters as the B_0 and B_1 magnetic fields and possible inhomogeneities, it is possible to calculate the signal response to an MRF sequence for different combinations of tissue and system properties. The expected behaviour for several combinations of tissue and system parameters can be calculated, these combinations form the building blocks for a dictionary of signal evolutions. The so called dictionary is formed by these atoms, each corresponding to a different combination of parameters. The calculation of these dictionaries has already been implemented and will thus not be discussed much further. When a measured signal is equal (or close) to a dictionary signal, the T_1 and T_2 values of the corresponding voxel can be concluded, in Section 2.5.1 this is discussed in more detail. This process of finding the corresponding dictionary atoms is called matching and a an the maximum inner product is used to do so. This choice for performing the parameter mapping in MRF has historical reasons, since the idea was adopted from using learned dictionaries in compressed sensing reconstruction for MR parameter mapping [28], where orthogonal matching pursuit (OMP) was used to find a sparse representation of a signal

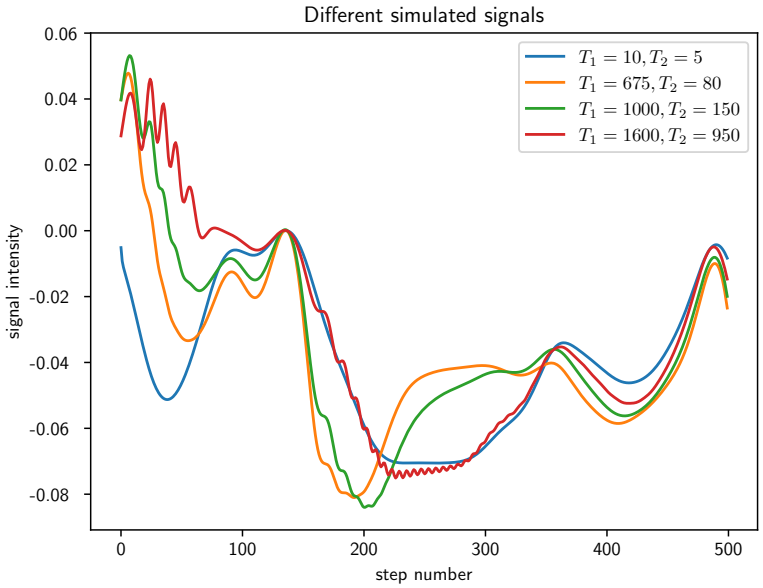


Figure 2.5.: Simulated signal evolution curves for four different relaxation time combinations.

evolution in a model-based dictionary. In MRF the dictionary learning step is omitted and the training data itself is used as a dictionary, reducing the signal representation to a single step of the OMP algorithm which is finding the dictionary atom with the largest inner product with the test signal.

In Figure 2.5 different simulated signal evolutions curves are shown. A dictionary can contain many atoms and, as can be seen in Figure 2.5, the corresponding signals have different shapes and they are more different than the exponential signals as measured with a multi-echo spin-echo measurement. When the T_1 and T_2 values are close to each other, the signals will become similar, just as with exponential signals with close relaxation times. The number of dictionary atoms can differ largely, in the introducing paper 563,784 [27] atoms were used, but for different implementations smaller dictionaries have been tried. In the introducing paper the off-resonance frequency was included as a characteristic system parameter.

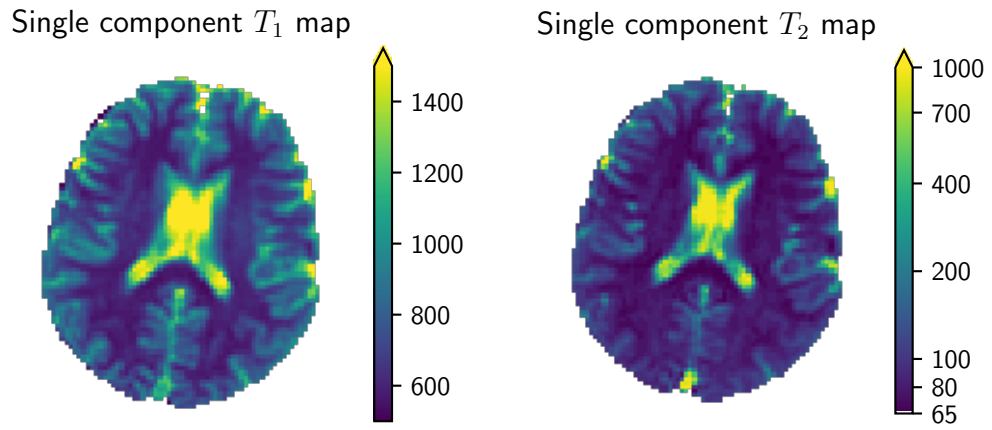
Using the above described method, MRF provides a new way to perform quantitative imaging measuring multiple tissue properties at the same time. The matching as described above is essentially the first iteration of a model based compressed sensing reconstruction. For spatio-temporally incoherent sampling and long sequences undersampled data can be used for the matching, which gives a great improvement in the acquisition time. In the general case on may need an iterative reconstruction [29].

2.5.1. Single component matching

In the introducing paper [27] a simple pattern recognition algorithm was used to find a single component. With a single component matching a process is meant in which each pixel is matched to one dictionary atom. This simple pattern recognition algorithm is based on the inner products between the measured signal and the calculated signals. For each pixel a complex signal $\mathbf{x} \in \mathbb{C}^M$ consisting of M time points is measured. The complete dictionary can be written as the matrix $D := [\mathbf{d}^1, \mathbf{d}^2, ..., \mathbf{d}^N] \in \mathbb{C}^{M \times N}$, where N is the number of different parameter combinations. For N sets of characteristic parameters the time signals $\mathbf{d}^i \in \mathbb{C}^M$ $(i = 1, 2, ...N)$ are calculated and normalised to $\|\mathbf{d}^i\|_2 = 1$. The dot product between the measured time signal and the simulated dictionary entries gives a measure of the similarity between $\mathbf{x}$ and each $\mathbf{d}^i$. This results in the matching formula:

$$i^* := \underset{i \in \{1,2,...N\}}{\arg\max} |\mathbf{d}^i \cdot \mathbf{x}^*|, \tag{2.7}$$

where i^* is the index of the matched dictionary atom and $\mathbf{x}^*$ is the Hermitian conjugate of $\mathbf{x}$. This method has proven to give good results and has a certain degree of tolerance for errors, as long as these errors are incoherent in time[30]. In Figure 2.6 the T_1 and T_2 maps from a single component matching are shown, which give an idea of the possibilities of MRF.



(a) T_1 single component mapping. (b) T_2 single component mapping.

Figure 2.6.: A single component mapping, based on the noise sequence of length 200.

2.5.2. Compression

The signal evolutions are usually compressible in the temporal direction and it is therefore possible to compress the time signals of length M (typically 200-1000) to a length k (typically 20 to 40), which allows speed-up in the vector-dot product with a small error. The main method to do this compression is by using a singular value decomposition

(SVD) as introduced by McGivney et al. for MRF [31].
The compression using an SVD is performed in the following way. First the SVD of the transposed dictionary D^T is determined,

$$D^T = U\Sigma V^* \tag{2.8}$$

where Σ is a matrix with the singular values of D^T on the diagonal ordered nonincreasingly, the columns of U contain the left singular vectors of D^T and similarly V the right singular vectors. The matrices U and V are complex unitary matrices. When D^T is real valued, the SVD is also real valued. The rank of the matrix D is denoted by r and thus the first r columns of V form an orthogonal basis for the rows of D^T. By taking only the first k right singular vectors of D a compression matrix V_k can be formed, which compresses the dictionary D to a lower-dimensional matrix

$$DV_k =: D_k^T \in \mathbb{C}^{N\times k}. \tag{2.9}$$

The compression matrix can also be used to compress the measured time signal $\mathbf{x}$ to a compressed signal of length k:

$$\mathbf{x}_k := \mathbf{x}V_k. \tag{2.10}$$

Using the property that V is a unitary matrix, the product $V_kV_k^*$ will thus approximate the identity matrix, the inner product $\mathbf{d}^i\mathbf{x}$ can be approximated as $\mathbf{d}^iV_kV_k^*\mathbf{x} = \mathbf{d}_k^i\mathbf{x}_k$, which results in a small matching error. Since the compression of the dictionary has to be performed only once, it is efficient to use this compression. Unless explicitly mentioned differently it can be assumed that the compressed dictionary and signal are used.

2.5.3. Connection to relaxometry and Inversion Recovery balanced Steady State Free Precession

Just as in the case where a multi-echo spin-echo is performed, multiple measurements are taken over time in MRF measurements. Where the flip angle and repetition time are normally varied in the MRF method, the T_2 exponential analysis has certain similarities to MRF. In the general case of MRF more parameters are encoded in the measured signal. It can therefore be interesting to see how multi-component T_2 analysis methods can be used in the more general MRF setting to develop a general framework which is applicable to both MRF and T_2 multi-component analysis. There are however also methods, which are specifically designed for exponential signals as discussed in Section 2.7.9.
For a multi-echo spin-echo T_2 measurement the shape of the signal is determined by the T_2 relaxation:

$$S(t) = e^{-t/T_2}, \tag{2.11}$$

where t is the time and S is the signal. The scaling of the signal is determined by the spin density.
One method for simultaneous T_1 and T_2 mapping is *Inversion Recovery balanced Steady State Free Precession* (IR-bSSFP)[32, 33]. In this measurement a fixed flip angle α and repetition time TR are used, but precessed by an RF-pulse with flip angle $-\alpha/2$ and

repetition time $TR/2$. This preparation pulse makes sure that the resulting signal is exponential. As shown in [22] the measured signal encodes the T_1 and T_2 relaxation times. Signal $S(t)$ can be described by:

$$S(t) = S_{stst}\left[1 - INV \cdot \exp\left(-\frac{t}{T_1^*}\right)\right], \tag{2.12a}$$

$$T_1^* = \left(\frac{1}{T_1}\cos^2\frac{\alpha}{2} + \frac{1}{T_2}\sin^2\frac{\alpha}{2}\right)^{-1}, \tag{2.12b}$$

$$INV = 1 + \frac{\sin\frac{\alpha}{2}}{\sin\alpha}\left[\left(\frac{T_1}{T_2}+1\right) - \cos\alpha\cdot\left(\frac{T_1}{T_2}-1\right)\right], \tag{2.12c}$$

$$S_{stst} = \frac{M_0\sin\alpha}{\left(\frac{T_1}{T_2}+1\right) - \cos\alpha\cdot\left(\frac{T_1}{T_2}-1\right)}, \tag{2.12d}$$

where M_0 is the proton density. When $\alpha = 180°$ the signal reduces to (2.11) with a scaling of $-M_0$, although this might not seem directly clear. The signal thus only depends on the T_1, T_2 and M_0 values and forms a step from the T_2 single parameter mapping and the multi parameter mapping as in MRF.
In the paper [34] the IR-bSSFP sequence with a constant flip angle was used to perform a multi-component analysis for T_1^*. The inversion recovery balanced SSFP sequence was therefore considered as well in this thesis, since it forms a step between the exponential signal only depending on T_2 and the complicated sequences used in MRF decoding T_1 and T_2. Instead of distinguishing T_1^* values it was attempted to recover the underlying T_1 and T_2 combination. However, it turned out while performing simulations that the IR-bSSFP was not usable for multi-component analysis distinguishing between both T_1 and T_2 instead of only T_1^*. This result is discussed in a conference abstract submitted for the ISMRM (International Society for Magnetic Resonance in Medicine) 2019 conference[35].

2.6. Multi-component analysis

The aim of a multi-component analysis is to recognise the presence of multiple tissues in one voxel and to determine which tissues are present in the considered volume with the corresponding intensities of each tissue. As said before, this can be of relevance in several cases, for example to detect tissues which do not dominate a voxel but are only present in a low concentration, this would be the case for myelin in white matter. It would also provide a better way to inspect the boundary zone between white and grey matter.
To perform such a multi-component analysis, the techniques and the dictionary of MRF can be used. Instead of assuming that the measured signal can be represented by one single dictionary atom, several dictionary atoms should be used. It is assumed that only a small number of tissues are present in each voxel. The problem setting in which each voxel is considered individually is discussed in the following section. Afterwards a more generalised setting is discussed in which multiple voxels are considered at once. The methods to solve these problems are discussed in the next chapter.

In the following sections it is assumed that the dictionary contains many atoms and the goal would thus be to find the atoms which provide a good representation of the measured signal. When a small dictionary with only a few atoms [27] would be used, the problem formulation would stay the same, but the dictionary would be less correlated. Because of that simpler methods than considered here would probably work.

2.6.1. Voxel-by-voxel multi-component analysis

In this section each voxel is considered independently. The measured temporal signal evolution for a single voxel is will be denoted as $\mathbf{x}$, which can contain noise. The multi-component analysis explained in Section 2.4, can be expanded to a multi-parameter setting:

$$x(t_i) = x_i = \iint_P S(p;t_i)c(p)dp + \epsilon_i \ , i = 1, 2, ...M, \tag{2.13}$$

where P is the parameter space, which consists of T_1 and T_2 values in this project. Similar as in Section 2.4 it is assumed that the signal $\mathbf{x}$ is formed by a linear combination of different tissues as represented by the dictionary atoms formed by signals corresponding to the values in the discretised parameter space. Combined with noise $\mathbf{e}$ this leads to

$$\mathbf{x} = D\mathbf{c} + \mathbf{e}, \tag{2.14}$$

where D is the dictionary and $\mathbf{c}$ a vector containing the weight of each dictionary atom. The dictionary signals will only approximate the true tissue signals, since a dictionary can not contain every possible combination of tissue parameters. To deal with the presence of noise a least squares problem can be formulated:

$$\min_{\mathbf{c}\in\mathbb{C}^N} \|\mathbf{x} - D\mathbf{c}\|_2^2, \tag{LS}$$

where $D \in \mathbb{C}^{M\times N}$ is the dictionary. In this formulation the minimal solution can be complex, which does not represent a proton density or concentration, as sought. Therefore the solution is restricted to non-negative real values, denoted as $\mathbb{R}_{\geq 0}$:

$$\min_{\mathbf{c}\in\mathbb{R}^N_{\geq 0}} \|\mathbf{x} - D\mathbf{c}\|_2^2, \tag{NNLS}$$

where the dictionary D and signal $\mathbf{x}$ are expected to be real. It is assumed, as mentioned earlier, that only a small number K of components should be present in each voxel. This would lead to following problem formulations:

$$\min_{\mathbf{c}\in\mathbb{R}^N_{\geq 0}} \|\mathbf{x} - D\mathbf{c}\|_2^2 \qquad \text{subject to } \|\mathbf{c}\|_0 \leq K, \tag{NNℓ_0a}$$

and

$$\min_{\mathbf{c}\in\mathbb{R}^N_{\geq 0}} \|\mathbf{c}\|_0 \qquad \text{subject to } \|\mathbf{x} - D\mathbf{c}\|_2^2 \leq \epsilon, \tag{NNℓ_0b}$$

where $\|\mathbf{c}\|_0$ denotes the ℓ_0 pseudo-norm which counts the number of non-zero elements in $\mathbf{c}$. The problem formulation (NNℓ_0a) restricts the solution to a maximum number of components, without any more generalisations and would therefore be the main problem of interest. However, these problems can in general not be solved easily, since they have been shown to be NP-hard [36].
The problem (NNℓ_0b) can be simplified by changing from the ℓ_0-norm to the ℓ_1 norm, which could then be written as

$$\min_{\mathbf{c}\in\mathbb{R}^N_{\geq 0}} \|\mathbf{c}\|_1 \qquad \text{subject to } \|\mathbf{x} - D\mathbf{c}\|_2^2 \leq \epsilon. \tag{NNℓ_1}$$

It has been shown by Donoho [37] that this problem has the same solution as (NNℓ_0b) when the matrix D is nearly orthonormal. In compressed sensing this is often the case, since dictionaries are selected based on this property. In the MRF case, the dictionary is however not close to orthonormal, meaning that there is no guarantee that the solutions of (NNℓ_0b) and (NNℓ_1) are equivalent.

Regularisation

The constrained ℓ_1-norm minimisation problem (NNℓ_1) can be reformulated in its Lagrangian form as a regularised least squares problem:

$$\min_{\mathbf{c}\in\mathbb{R}^N_{\geq 0}} \|\mathbf{x} - D\mathbf{c}\|_2^2 + \lambda\|\mathbf{c}\|_1, \tag{NNLSℓ_1}$$

where λ is a regularisation parameter, determining how much weight is addressed to the $\|\mathbf{c}\|_1$ term and thus how sparse the solution will become. The balancing of λ depends on the amount of noise present in the signal.
Besides this regularisation other regularisation choices are possible. A general ℓ_2 regularisation, also called Tikhonov regularisation, can be written as

$$\min_{\mathbf{c}\in\mathbb{R}^N_{\geq 0}} \|\mathbf{x} - D\mathbf{c}\|_2^2 + \lambda\|S\mathbf{c}\|_2^2, \tag{NNLSℓ_2}$$

where S is a matrix which alters the way of Tikhonov regularisation. The following Thikonov regularisations can be achieved in this way:

- Minimising the "energy" of the spectrum, setting S to be the identity matrix, which is equivalent to adding the constraint

$$\sum_{j=1}^{M} c_j^2. \tag{2.15}$$

- Minimising the energy of the derivative, using a finite difference:

$$\sum_{j=1}^{M-1} |c_{j+1} - c_j|^2. \tag{2.16}$$

  This regularisation is the common regularisation of choice used in T2NNLS.

- Minimising the curvature of the spectrum:

$$\sum_{j=1}^{M-2} |c_{j+2} - 2c_{j+1} - c_j|^2. \tag{2.17}$$

  This method, called rNNLS, was considered in [38] with the conclusion that the non regularised method was giving better results.

The first kind of regularisation, where S is the identity matrix is often used to promote small solutions with respect to the ℓ_2-norm. In this setting there is no guarantee that sparse solutions are favoured over any other solution. The result of this regularisation is that the energy of the solution is minimised, which can be favourable. Another beneficial property is that the problem (NNLSℓ_2) can be rewritten as

$$\min_{\mathbf{c}\in\mathbb{R}^N_{\geq 0}} \|\mathbf{y} - A\mathbf{c}\|_2^2, \tag{2.18}$$

where $A = \begin{bmatrix} D \\ \sqrt{\lambda}S \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{0}_N \end{bmatrix}$ with $\mathbf{0}_N$ the zero vector of length N. In this case general methods used for non-negative least square problems can be used. However, it results in a matrix A of size $(N+M)\times N$ and since $N \gg M$ the problem is much larger than the original problem.
A third regularisation is of the form

$$\min_{\mathbf{c}\in\mathbb{R}^N_{\geq 0}} \|\mathbf{x} - D\mathbf{c}\|_2^2 + \lambda\|\mathbf{c}\|_1^2, \tag{NNLSℓ_1^2}$$

where the ℓ_2-norm is replaced by an ℓ_1-norm, but is still squared. This problem can be rewritten [39] in the form of (2.18), but with $A = \begin{bmatrix} D \\ \sqrt{\lambda} \quad ... \quad \sqrt{\lambda} \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix}$. This can be done since

$$\|c\|_1^2 = \left(\sum_{i=1}^{N} c_i\right) = \begin{bmatrix} 1 & ... & 1 \end{bmatrix} \mathbf{c} \tag{2.19}$$

for $\mathbf{c} \geq 0$. Where in the previous regularisation N rows were added to the system, only 1 row is added to the system in this regularisation method, making it computationally less intensive.

2.6.2. Joint sparsity approach

In the previous section a voxel-wise approach was used. It might however be useful to consider multiple voxels at once. By doing so, it can be possible to include the assumption that the region of interest contains only a small number tissues and that the relaxation times don't vary too much in one tissue. This core idea can be applied by introducing a joint sparsity constraint, meaning that all the measured signals can be represented by a small common group of dictionary atoms.

When a set of voxels is considered, the set of measured signals $\mathbf{x}^j$ $(j = 1, ...J)$, where J is the number of voxels, can be combined in the matrix $X = \begin{bmatrix}\mathbf{x}^1 & ... & \mathbf{x}^J\end{bmatrix}$. The non-negative least squares problem (NNLS) can than be written as a joint problem:

$$\min_{C \in \mathbb{R}^{N \times J}_{\geq 0}} \|X - DC\|_F^2, \tag{JNNLS}$$

where C is the matrix containing the voxel-wise solutions and $\|\cdot\|_F^2$ denotes the Frobenius norm. Again the assumption that only a small number of components is present in the solution is made. The resulting problem is known by different names: the joint sparsity model with common sparse support (JSM-2) [40], the multiple measurement vector (MMV) problem with a common sparsity profile[41], the simultaneous sparse approximation problem (SSA) [42, 43] or joint sparse recovery [44]. All these problem settings, do not contain the non-negativity constraints, which are added here:

$$\min_{C \in \mathbb{R}^{N \times J}_{\geq 0}} \|X - DC\|_F^2 \quad \text{subject to: the matrix } C \text{ has at most } K \text{ nonzero rows.} \tag{SSA}$$

This means that all of the signals X can be approximated with the fixed set of K dictionary atoms. The requirement that C has at most K nonzero rows, can also be written as

$$\|C\|_{\text{row}-0} := |\text{rowsupp}\,(C)|,$$

where $\text{rowsupp}\,(C) := \{\omega \in \Omega \,:\, c_{\omega k} \neq 0 \,\text{for some}\, k\}$. This can be written as:

$$\min_{C \in \mathbb{R}^{N \times J}_{\geq 0}} \|X - DC\|_F^2 \quad \text{subject to}\, \|C\|_{\text{row}-0} \leq K \tag{SSAa}$$

or in the different form:

$$\min_{C \in \mathbb{R}^{N \times J}_{\geq 0}} \|C\|_{\text{row}-0} \quad \text{subject to} \|X - DC\|_F^2 \leq \epsilon. \tag{SSAb}$$

Since this problem is, just as the (NNℓ_0a) problem, NP hard, it can be interesting to look into a family of relaxed problems [41, 43]:

$$\min_{C \in \mathbb{R}^{N \times J}_{\geq 0}} \|X - DC\|_F^2 + \lambda J_{p,q}(C), \tag{2.20}$$

where

$$J_{p,q}(C) := \sum_{i=1}^{N} \left(\sum_{j=1}^{J} |c_{ij}|^q \right)^{p/q}.$$

Typically $p \leq 1$ and $q \geq 1$. The row-0 norm is the case where $p = 1$ and $q = \infty$. This relaxation family is convex when $p, q \geq 1$.
The problem on which will be focused on most (or relaxations of this) can be written as:

$$\min_{C \in \mathbb{R}^{N \times J}_{\geq 0}} \|X - DC\|_F^2 + \lambda \sum_{i}^{N} \|\mathbf{c}^i\|_0, \tag{2.21}$$

where $\mathbf{c}^i$ denotes the i$^{\text{th}}$ row of C.
The problem formulations 2.20 and SSAa lead to multiple directions to find a joint solution for a larger set of voxels. Methods to do so will be discussed in the next chapter.

2.7. T_2 multi-component analysis algorithms

In this section several algorithms which are not implemented in this project are discussed, to give a more complete overview of the field.
Exponential T_2 multi-component analysis methods have been applied for a long time [8, 26, 45, 46]. In the T_2 approach it is assumed in general that the T_2 values are distributed as a smooth spectrum, where in the here discussed approach it is assumed that the T_2 spectrum is sparse. The smoothness assumption is based on the thought that the relaxation times vary naturally and do not have exactly one value. Adding the smoothness also gives a better posed problem, making the problem solving more robust. At the same time the final result is often that the smooth peaks which are found are grouped as one, only noting the mean relaxation time of the grouped peak, leading to a sparse interpretation in the end.

2.7.1. T2NNLS

When an exponential multiple component analysis is performed, several methods are possible. A general method is the T2NNLS [8, 47, 26] method based on the NNLS algorithm where the total energy of the derivative of the spectrum is minimised, which is the same as a Tikhonov regularisation as NNLSℓ_2. The regularisation is used to smooth the results and to prevent a too sparse solution, containing only some peaks. The value of the regularisation parameter is chosen based on the value of the misfit

$$\chi^2 = \sum_{i=1}^{M} \frac{(x_i - \tilde{x}_i)^2}{\sigma^2}, \tag{2.22}$$

where $\tilde{\mathbf{x}}$ is the approximated solution and σ the standard deviation of the noise. The goal is then to find a λ such that

$$\chi^2 = M + x\sqrt{2M}$$

with $0 \leq x \leq 1$ and where M is the number of time points measured. In this method the absolute value of the signal is used, since the exponential decay is assumed to return a positive, real signal. If one wants to use compression, the absolute value of the signal should be taken before the compression. However, the use of compression is less necessary because of the shorter time signals used in relaxography.

2.7.2. Spatial regularisation

In the previously introduced joint problem setting information of all the voxels is used, without considering the spatial location of the voxels. It is also be possible to only use the information from neighbouring voxels and the components present in these voxels, instead of using all voxels.
One method used for T_2 relaxation multiple component analysis is proposed in [9], where it is assumed that the distributions of relaxation times in neighbouring voxels are similar. The spatially regularised NNLS problem is introduced as follows:

$$\min_{\mathbf{c}\in\mathbb{R}^N_{\geq 0}} \|\mathbf{x} - D\mathbf{c}\|_2^2 + \lambda\|S\mathbf{c} - \mathbf{p}\|_2^2, \tag{2.23}$$

where $\mathbf{p}$ is the a priori spectrum. The a-priori spectrum used in [9] is the average over the spectra of the 8 closest pixels and the pixels itself when the (NNLSℓ_2) has been solved for a smooth spectrum. For the regularisation matrix S the identity matrix has been used. Using this Thikhonov regularisation leads to a smooth spectrum of relaxation times. This is different from the problem settings on which the focus lies in this thesis, since the proposed methods in this thesis are searching for a sparse solution, instead of a smooth spectrum. To include this spatial regularisation either one should search for a smooth solution or use a relatively coarse dictionary, making it more likely that solution in the neighbouring pixels contain the same dictionary atoms.

2.7.3. Relaxation-Relaxation Correlation Spectroscopic Imaging

In the paper [48] a comparable method is proposed, using a 2 dimensional parameter space consisting of the T_1 and T_2 relaxation times and addressing all the voxels at once instead of performing one iteration in the method of [9]. Although the method is not explicitly applied to MRF measurements, it uses a 2D parameter space just as MRF. The proposed method is called Relaxation-Relaxation Correlation Spectroscopic Imaging (RR-CSI). The problem is formulated as

$$\{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_J\} = \underset{\{\mathbf{c}_i\in\mathbb{R}^N_{\geq 0}\}_{i=1}^J}{\arg\min} \left[\sum_{i=1}^{J} t_i\|\mathbf{x}_i - D\mathbf{c}_i\|_2^2 + \lambda\sum_{l\in\Delta i}\|\mathbf{c}_i - \mathbf{c}_l\|_2^2\right], \tag{2.24}$$

where t_i is either 0 or 1 and indicates whether a voxel is taken into account or not. The Δi gives the set of neighbouring voxels. This problem is solved with an alternating directions method of multipliers (ADMM) algorithm. No notions about the computation times are made, but from the method as mentioned in [9] it is known that the method is computationally rather expensive, which is expected here as well, but unconfirmed.
In [48] results are shown for MRI scans of the brain, where they find a smooth spectrum consisting of 6 peaks, which are translated to six components. Four components are located around $T_1 = 1$s, $T_2 = 100$ms, one component coupled to myelin is located around $T_1 = 100$ms and $T_2 = 10$ms and the last component seems to resemble CSF

around $T_1 = 2$s and $T_2 = 300$ms. It is noted here that the relaxation times of these two last components do not correspond to the relaxation times reported in different literature[4, 49].

2.7.4. B_1-compensated regularised NNLS

One of the first papers which includes the correction for B_1 inhomogeneities is [10] by Prasloski et al.. This algorithm will be referred to as B_1-compensated regNNLS. The extended phase graph (EPG) [50] is used to calculate the signal evolution for a given B_1 variation (noted as A) for different T_2 evolutions. In the matching a range of logarithmically spaced T_2 values from 10ms to 2s is used. The algorithm is applied voxel wise and works as follows:

1. Determine the B_1 correction value A
   a) T_2 dictionaries for eight different A values are calculated.
   b) For each of the eight dictionaries a NNLS multi-component matching is performed.
   c) For each matching and A value the sum of squares (SoS) of the residuals is determined.
   d) A smooth curve along the A and SoS values is determined.
   e) The optimal A $\hat{A}$ is determined from this smooth curve.
2. $\hat{A}$ is used to compute a dictionary for this B_1 correction value.
3. Using regularised T2NNLS a multi-component matching is performed.

In each of the T2NNLS computation steps the regularisation as in equation 2.16 is used. The regularisation parameter is determined with the method as discussed in Section 2.7.1 based on the χ^2 value. The regularisation parameter is determined independently for each voxel, making it necessary to use the NNLS algorithm with different regularisation parameters several times per voxel.
Besides this, the algorithm is relatively slow since the T_2 dictionaries with different B_1 compensation are calculated in each iteration. It would probably lead to a significant speed-up when the T_2 dictionaries with a range of B_1 values would be calculated on forehand.

2.7.5. B_1-compensation and spatial smoothing

A method similar to the method in Section 2.7.3 for T_2 relaxation is proposed in [12] and followed up by [51], where this method includes the correction of B_1 errors using extended phase graphs. This method thus proposes a combination of spatial and temporal regularisation and correction for flip angle inhomogeneities, caused by B_1. The processing of full brain data ($80 \times 80 \times 64$ voxels) took 15-16 hours using Matlab on a computer

with 5 cores and 32 GB RAM.

2.7.6. Multi-Gaussian models

Some other methods assuming spatially smooth T_2 spectra have been published as well. A spatially constrained Multi-Gaussian model for T_2 relaxation is proposed in [11]. The computation time for 8 slices with 256×256 pixels is reported to take 1.2 hours in this implementation. Another Bayesian method has been proposed in [52] reporting a processing time of 5 hours for the same problem size.

2.7.7. Fixed dictionary

The first approach to a multi-component decomposition was proposed in the supplemental material of the original MRF paper [27]. It was proposed to use a dictionary D with only 3 dictionary atoms. The decomposition was then determined as $\mathbf{c} = D^{\dagger}\mathbf{x}$, where $D^{\dagger}$ is the pseudo inverse of D and thus giving the least square solution. This approach is very sensitive to the assumptions on the chosen relaxation times for the dictionary atoms.

2.7.8. GAP-MRF

An arXiv preprint [53] came on-line in the second half of this project proposing a greedy approximate projection algorithm for MRF with partial volumes. This method aims at finding a few signals representing the different pure tissues. The five resulting tissues are white- and grey matter, CSF, muscle and fat tissue. These five components, forming the dictionary, are determined through a Greedy algorithm. In this algorithm the Bloch equations are solved several times, to take different combinations of T_1 and T_2 into account. As indicated by the term partial volume, this method is mainly focusing on components which are somewhere present in a pure form and is thus not sensitive to myelin water.

2.7.9. Not dictionary based T_2 relaxation multi-component methods

EASI-SM A method which is specifically aimed at exponential signals is the EASI-SM (Exponential Analysis via System Identification using Steiglitz-McBride) algorithm [54, 55, 56]. The EASI-SM tries to find the appropriate T_2 values to be used to form the dictionary, which is a difference with the T2NNLS method, where the T_2 values forming the dictionary are chosen on forehand. This algorithm is compared in [57] to the T2NNLS algorithm and it is shown to give better results. This method uses a voxel-by-voxel approach and does not include information about neighbouring voxel.

mcDESPOT A specific method used to find two components and therefore aimed at the determination of the myelin water fraction is called mcDESPOT (multicomponent

driven equilibrium single pulse observation of T_1/T_2) [58]. This multi-component method assumes that the signal can be decomposed to two underlying species, a fast relaxing and a slow relaxing component. These two components are sought in two pools of possible values using a non-linear least squares fitting, making it relatively dependent on the initial conditions. As a result T_1, T_2 and a fraction for the fast and slow component are found for each voxel. The fraction of the fast component is normally seen as the myelin water fraction.
In [59] a comparison between the T2NNLS and mcDESPOT method is made. The conclusion here is that the resulting MWF maps can not be compared one-to-one and that the myelin fractions are higher in the mcDESPOT method than in the T_2NNLS method.

3. Numerical methods

Based on the different problem settings as given in Chapter 2 the main methods to perform a multi-component matching on an MR fingerprinting signal are addressed. This chapter starts with an intermezzo about the preparation of the signals in Section 3.1, this includes the change from complex to real signals and the decision concerning which pixels relevant. Afterwards different methods for voxel-by-voxel analysis are discussed in Section 3.2 and algorithms including the joint sparsity constraint in Section 3.3. This chapter ends with a short remark how to include the compensation of flip angle inhomogeneities or other inhomogeneities in the region of interest in Section 3.4.

3.1. Signal preparation

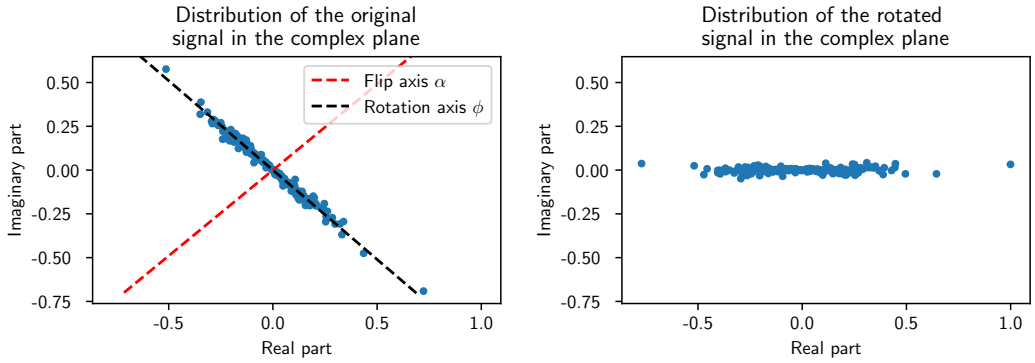
For certain algorithms it can be necessary to first restructure or slightly change the measured signals. This will be explained in this section.

3.1.1. From complex to real signals

The measured and simulated signals are in general complex valued, as discussed in Section 2.1. The algorithms discussed in the following sections are however not always able to deal with complex signals and dictionaries, especially when non-negative solutions are sought. To overcome this difficulty, it is necessary to transform each complex valued signal to a real valued signal. In this section it will be discussed how to perform such a transformation, without losing useful properties.
As discussed before, caused by the MR techniques and properties, each measured voxel signal is rotated with a voxel dependent angle. This angle ϕ is caused by several factors and is therefore difficult to calculate, but is assumed to be constant for each voxel over time. The signal $\mathbf{x}$ can thus be written as

$$\mathbf{x} = e^{i\phi}\mathbf{x}_r, \tag{3.1}$$

where $\mathbf{x}_r$ is the signal without the phase shift. In the single component matching, this phase shift is no problem since the constant phase shift does not influence the inner product.
A first approach would be to take the absolute value of each signal and use that as input for the used algorithms. However, when the dictionary signals are partly negative, which is normally the case, the order of summation of the signals and taking the absolute value influences the outcome, which is not a desired behaviour. For example when two summed signals contain 1 and -1 at a certain time point, the sum would be 0, but the absolute



(a) The original signal (b) The rotated signal

Figure 3.1.: The distribution (dotted) in the complex plane of a rescaled, measured signal of length 200, before rotation to the real axis (a) and after rotation (b) .

sum would be 2, leading to completely different results.
A second approach would be to split the complex signal in its real and imaginary part and to combine it in to one vector of length $2M$. However, this approach does neither result in a behaviour similar to a complex method and it would mean that the length of the signals would become twice as long, increasing the calculation times.
It is however possible to make the step from a complex signal to a real signal in a different way, the idea of this approach can be seen in the notation as in equation (3.1). When an angle ϕ can be determined such that $\mathbf{x}_r$ is dominantly real-valued, taking the real values of $\mathbf{x}_r$ would give all the desired properties.
In Figure 3.1a the distribution in the complex plane of a rescaled, measured signal is shown. As can be seen the measured values are approximately on one line in the complex plane, with a certain angle ϕ. This ϕ can be used to determine the rotation axis, which is rotated to the real axis. In Figure 3.1b the distribution of the corrected signal is shown.
The rotation angle has been determined using the newly introduced algorithm 1. In determining this rotation angle a possible problem is that the signal can either be alternating with a phase of 180 degrees or not, it is relatively easy to detect this and thus to compensate for this.
If the signal $\mathbf{x}_r$ is completely non-negative, the signal $\mathbf{x}$ is completely located in one half of the complex plane. It is then possible to determine the rotation angle by taking the mean of the phases of the original signals. When the signal $\mathbf{x}$ is not located in one half of the complex plane, this leads to an error in the determined angle. A short example would help here again, if half of the signal is 1 and the other half -1, the resulting average phase will be π (or $-\pi$) which is clearly incorrect.
This can be prevented by first flipping the signal to one half of the complex plane. The flipping to the other complex halve is equivalent to a multiplication by -1 and is performed in line 7 of the algorithm. To perform this flipping a flipping angle α is determined,

which is used to divide the complex plane in two halves.
This flipping angle is a rough guess of an angle which is far away from the rotation angle and therefore the mean phase of the signal is fine for an alternating signal and for a non-alternating signal, this angle is rotated by 90 degrees.
After flipping the signal, the phases should be close to each other and close to the rotation angle. When the rotation angle is however close to π (or $-\pi$), taking the mean angle can give wrong results. To explain this a bit better, an example will be given. Assume that the phase of a signal is slightly smaller than π, a small amount of noise can cause a part (one quarter for example) of the phases to be $\pi + \epsilon$, which corresponds to a phase of $-\pi + \epsilon$ since the phase is always between $-\pi$ and π. Assuming that one fourth of the signal has a phase of $-\pi + \epsilon$ and the rest π, the resulting average phase will be $\frac{1}{2}\pi$, which is not the expected phase of $-\pi$.
When this problem occurs the whole signal can better be flipped, leading to a rotation angle close to 0, which is done in line 10. When the rotation angle has been determined,

Algorithm 1 Algorithm to find the rotation angle

INPUT: $\mathbf{x}$ - a complex signal
OUTPUT: ϕ - the phase to rotate $\mathbf{x}$ to the real axis

1: $\alpha \leftarrow \text{mean}(\text{angle}(\mathbf{x}))$
2: **if** signal is not alternating **then** $\alpha \leftarrow \alpha + \frac{\pi}{2}$
3: **end if**
4: $n \leftarrow \texttt{len}(\mathbf{x})$
5: **for** i in $\texttt{range}(n)$ **do**:
6: **if** $\alpha \leq \text{angle}(\mathbf{x}[i]) \leq 2\pi - \alpha$ **then**
7: $\mathbf{x}[i] \leftarrow -\mathbf{x}[i]$ ▷ Flip the signal
8: **end if**
9: **end for**
10: **if** angle($\mathbf{x}$) is close to π **then** $\mathbf{x} \leftarrow -\mathbf{x}$
11: **end if**
12: $\phi \leftarrow \text{mean}(\text{angle}(\mathbf{x}))$
13: **return** ϕ

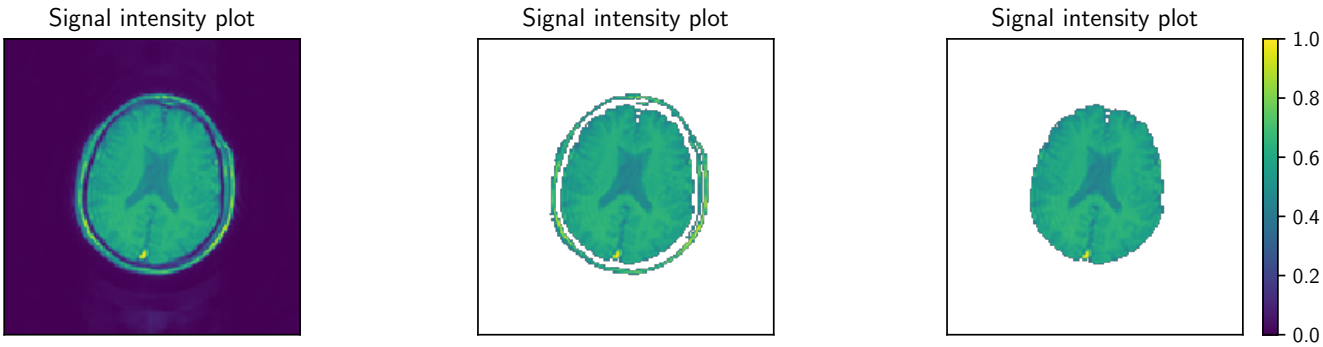
the real signal can be determined as

$$\mathbf{x}_r = \text{Re}(e^{-i\phi}\mathbf{x}). \tag{3.2}$$

In this step it is assumed that the complex part of the signal only contains noise.
There is however one problem in this approach. It might be that the rotation angle ϕ is π off, resulting in a sign difference with the true signal. By comparing the determined signal $\mathbf{x}_r$ with the simulated dictionary atoms, it is possible to correct for this sign difference. If most of the dictionary atoms have a large negative inner product with the signal $\mathbf{x}_r$ it can be assumed that the rotation angle ϕ was π off.

3.1.2. Image masking

Often only the centre of the image contains the relevant information. The rest of the image can either contain air (in the case of a brain scan) or less interesting tissues and organs. It can therefore be beneficial to mask these uninteresting areas.
In Figure 3.2a the relative intensity for each pixel is shown of a measurement of the brain, the signal intensities are calculated as the 2-norm of the signal and it is rescaled such that the largest signal has value 1. It can be seen that the brain is located in the centre of the image. The surrounding area contains almost no signal.
The first method to mask the part of the image which contains no information is by choosing a minimum value of the signal intensity. In Figure 3.2b the pixels with an intensity below 0.4 are skipped. This leads to a mask where 75% of the pixels are skipped. It can be seen that the air around the head and the skull are masked, however the tissue of the outer side of the skull is still considered as relevant.
To mask the surrounding tissue as well, a different method is used. When the resolution of the image is high enough and the right minimum intensity value is chosen, the surrounding tissue and the brain should be separated by a masked area. Using the `label` function from the `scipy.ndimage.measurements` package in Python it is possible to identify and label the different groups of pixels separated by a masked pixels. Using this labelling the largest group can be selected, which should correspond to the pixels corresponding to the brain. The result of this is shown in Figure 3.2c.



(a) The original image intensity.

(b) The masked image, pixels with a relative signal strength below 0.4 are skipped.

(c) The final masked image, only the largest group is maintained.

Figure 3.2.: The masking procedure of the image. In the final image 81% of the pixels are skipped.

3.1.3. Segmentation

When a joint signal approach is used, it is assumed that the signals are similar and the that the voxels contain similar tissues. When for example the brain is considered different areas can contain completely different tissue types, only showing the effect of mixture at the boundaries of these areas. It can therefore be beneficial to segment the image in to multiple segments, which can be considered sequentially in these joint approaches.
This segmentation can be done using a single component matching with a dictionary containing only a few atoms. These atoms should represent the main tissues present in the brain. In Figure 3.3 a segmentation based on three T_1, T_2 combinations is shown.

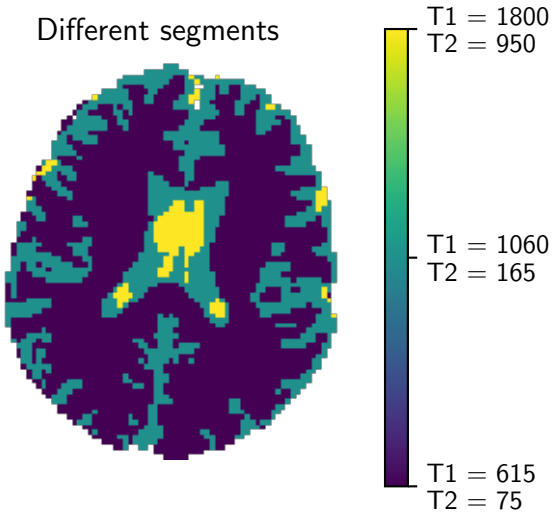


Figure 3.3.: The segmentation of a brain image in three segments.

The selection of the three tissues is however a difficult step. The chosen atoms should lead to a correct segmentation, where a segmentation is considered to be correct when each segment only contains a small number of tissues and at the differs from the other segments. One approach can be to choose a set of tissue parameters based on literature values or experience, which are used to form this small dictionary.
A different approach is by using information from the single component matching. For each voxel it is possible to determine which n (in this case 10) dictionary atoms are most similar to the measured signal. The number of times each of this dictionary atoms is represented in this top-n is counted, leading to a histogram. Since each atom corresponds to a combination of T_1 and T_2 a 3D-representation of the histogram can be retrieved. This 3D-histogram is shown in Figure 3.4. In this distribution it is possible to determine the local peaks, hopefully representing the main tissues and leading to meaningful segments. The red dot in Figure 3.4 indicate the selected peaks. The selection of the peaks is in

this case done by using the `maximum_filter` function from the `scipy.ndimage.filters` Python package.

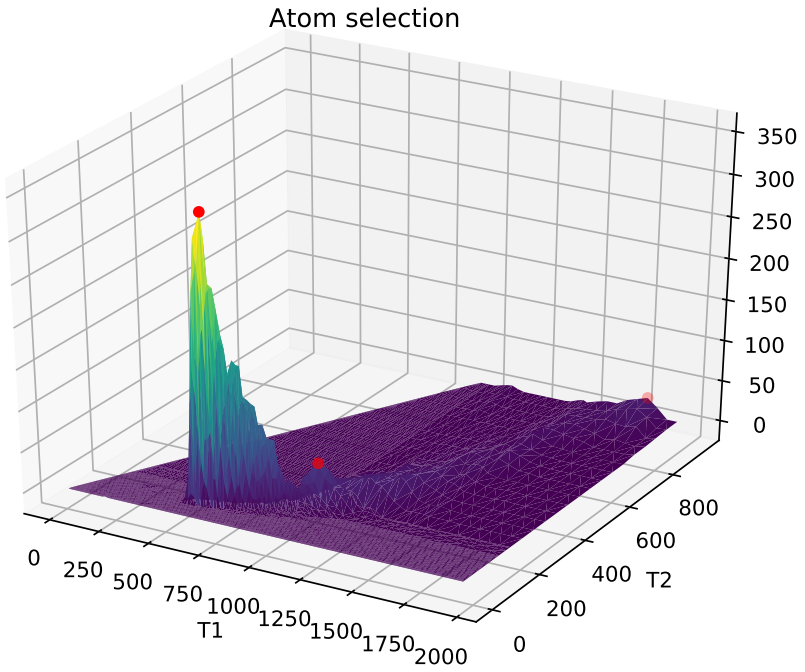


Figure 3.4.: The distribution of the use of the dictionary atoms as present in the whole image. The red dots indicate the selected dictionary atoms to be used for the segmentation.

3.1.4. Overview of algorithms

In the next sections several algorithms which can form the machinery of a multi-component analysis are discussed. In this section a short of overview of the different methods is given, to make it easier to keep the general overview.

- Voxel-by-voxel multi-component analysis algorithms
  - **Non-Negative least squares algorithm** (NNLS) of Lawson and Hanson *Section 3.2.1.*
    NNLS is used to denote this general algorithm to solve a NNLS problem and is a so-called active set algorithm, which normally tends to give sparse solutions, with short computation times.
  - **Bayesian approach** from [5] *Section 3.2.2.*
    Is an algorithm based on bayesian statics and a-priori knowledge about the distributions. To obtain a sparse solution pruning of the dictionary is used

and for the main solving step a conjugate gradient method is used. To obtain non-negative solutions the absolute value is taken in the end. The shape parameters α, β of the distributions and the regularisation parameter needs to be chosen in this method.

- **Reweighed ℓ_1-norm regularisation** from [6] *Section 3.2.3* .
  A reweighed ℓ_1-norm method is proposed, which uses a log-barrier method to obtain non-negative result and an interior point method where a Newton system is solved is used. To obtain sparse solutions the reweighting is of importance, multiple reweighting schemes are proposed to do so. Again the regularisation parameter needs to be optimised.

- **Non-Negative Orthogonal Matching Pursuit** (NNOMP) from [61, 62] *Section 3.2.4*. A combination of Orthogonal Matching Pursuit and the NNLS algorithm can be used to find a basis with a chosen number of components.

- Variants on NNOMP

  * **Support shrinkage NNOMP** (SNNOMP)
    When an element of the found support is not used any more, it is removed from the basis. This support shrinkage step also finds a basis with a pre-set size.

  * **Regularised NNOMP** (RNNOMP)
    The RNNOMP method expands the found NNOMP basis with one more element, to check whether this element improves the solution, removing the least used basis element to retrieve the required basis size again.

- Algorithms including the joint sparsity constraints, developed as part of this thesis

  - **Joint NNOMP** (JNNOMP) based on [42] and NNOMP. *Section 3.3.1*
    This algorithm combines the greedy simultaneous sparse approximation algorithm with the non-negativity restriction.

  - **Joint RNNOMP** (JRNNOMP) based on [63] and NNOMP. *Section 3.3.1*
    This algorithm combines the regularised NNOMP with the non-negativity restriction.

  - **Sparsity Promoting Iterative Joint NNLS** (SPIJN) a combination of the NNLS algorithm and ℓ_2 reweighting schemes from [64] *Section 3.3.2*.
    This algorithm combines joint sparsity with non-negative least squares by a joint reweighting scheme.

  - **Search Around** *Section 3.3.4*
    To relax the joint sparsity constraint, relaxation times close to the matched T_1 and T_2 are considered to improve the found basis voxel-wise.

3.2. Voxel-by-voxel methods

In this section several methods to solve the problem (NNLS) or relaxations of this problem are discussed. Starting with an old, general method, continuing with two methods designed for the multi-component MRF problem implemented in previously published papers, followed by two matching pursuit based methods.

3.2.1. Non-negative least squares algorithm

The non-negative least squares method has been introduced by Lawson and Hanson in 1974 [60], where they provided a Fortran implementation[1], which lead to it becoming an often implemented standard non-negative least squares solver. The algorithm is included in MATLAB as `lsqnonneg` and in Python in the package `scipy.optimize` as `nnls` with the earlier mentioned Fortran implementation. The NNLS algorithm is the most used algorithm to perform T_2 relaxation multiple component analysis [1, 8, 45, 46].
The algorithm has been discussed and rewritten in many papers (e.g. [65, 66, 39]). In Algorithm 2 the NNLS algorithm is given.
Here $D^\dagger$ denotes the pseudo inverse of the matrix D. The notation $D_\mathcal{P}$ means that the matrix D is restricted to the columns indicated by the set $\mathcal{P}$ and $\mathbf{c}_\mathcal{P}$ indicates that only the elements of $\mathbf{c}$ with indices in $\mathcal{P}$ are considered.
The NNLS algorithm is an active set algorithm [67]. This means that the algorithm searches for a set which is the support of the non-negative least square solution ($\mathcal{S}$) and a complementary set of active equations ($\mathcal{Z}$). Active refers here to the fact that $\forall i \in \mathcal{Z} : c_i = 0$, thus that the equality is active, contrary to $\forall i \in \mathcal{S}\, c_i > 0$ which are the in-active equalities. When this support of the NNLS solution has been found, the solution to $D_\mathcal{S}^\dagger \mathbf{x}$ must be a positive solution and since the pseudo inverse always results in a least squares solution, the result is the NNLS solution.
This section continues now with a more structured discussion of the NNLS algorithm. The algorithm takes as input the dictionary D and signal $\mathbf{x}$ and optionally an initial non-negative solution $\mathbf{h}$ to give the algorithm a "warm start". If no initial solution is provided, the zero vector is taken as initial solution. The support set and active set can be determined from the initial solution $\mathbf{h}$, if the initial solution is the 0-vector, the initial support set is empty.
In line 6 the residual $\mathbf{r}$ is determined, which is then used to determine $\mathbf{a} = D^T\mathbf{r} = D^T(D\mathbf{h} - \mathbf{x})$. The vector $\mathbf{a}$ is relevant since it is equal to the gradient which is minimised apart from a factor -2:

$$f(\mathbf{c}) = \|\mathbf{x} - D\mathbf{c}\|_2^2 \tag{3.3a}$$

$$f'(\mathbf{c}) = 2D^T(D\mathbf{c} - \mathbf{x}) \tag{3.3b}$$

$$f'(\mathbf{c})\big|_{\mathbf{c}=\mathbf{h}} = -2\mathbf{a}. \tag{3.3c}$$

[1] The code can be found here `https://github.com/scipy/scipy/blob/v0.18.1/scipy/optimize/nnls/nnls.f`

Algorithm 2 The NNLS algorithm [60]

INPUT: $\mathbf{x}$ - a real valued signal
D - a real valued dictionary
$\mathbf{h}$ - a non-negative initial solution (optional)
OUTPUT: $\mathbf{c}$ - the non-negative least square solution to (NNLS)

if $\mathbf{h}$ not provided **then**
  $\mathbf{h} = \mathbf{0}$ ▷ Set initial feasible solution
end if
$\mathcal{Z} = \{i | h_i = 0\}$ ▷ Set of active equations
$\mathcal{S} = \{i | h_i > 0\}$ ▷ Support set of the solution
$\mathbf{r} = \mathbf{x} - D\mathbf{h}$ ▷ Residual
$\mathbf{a} = D^T\mathbf{r}$
while $|\mathcal{Z}| > 0$ and $\exists i \in \mathcal{Z} : a_i > 0$ **do** ▷ Expand $\mathcal{Z}$ by one atom if possible
  $i^* = \arg\max \mathbf{a}$ ▷ The element used for the expansion
  $\mathcal{Z} \leftarrow \mathcal{Z} \backslash i^*$
  $\mathcal{S} \leftarrow \mathcal{S} \cup i^*$
  $\mathbf{c}_\mathcal{S} = D_\mathcal{S}^\dagger \mathbf{x}$ ▷ Find a LS solution within $\mathcal{S}$
  $\mathbf{c}_\mathcal{Z} = 0$ ▷ The elements in $\mathcal{Z}$ are set to zero
  while $\exists j \in \mathcal{S} : c_j < 0$ **do** ▷ Check if the solution is feasible
    $\alpha = \min_{k \in \mathcal{S}} h_k/(h_k - c_k)$
    $\mathbf{h} \leftarrow \mathbf{h} + \alpha(\mathbf{c} - \mathbf{h})$ ▷ Adjust feasible solution
    $\mathcal{Z} \leftarrow \{i | h_i = 0\}$ ▷ Update Z
    $\mathcal{S} \leftarrow \{i | h_i > 0\}$ ▷ Update S
    $\mathbf{c}_\mathcal{S} = D_\mathcal{S}^\dagger \mathbf{x}$ ▷ Find new solution
    $\mathbf{c}_\mathcal{Z} = 0$
  end while
  $\mathbf{h} \leftarrow \mathbf{c}$ ▷ Update $\mathbf{h}$ to feasible solution $\mathbf{c}$
  $\mathbf{r} = \mathbf{x} - D\mathbf{h}$
  $\mathbf{a} = D^T\mathbf{r}$
end while

Since the gradient has to be minimised, it can be seen that the optimal solution has been found if

$$a_i = 0, \qquad i \in \mathcal{S} \tag{3.4a}$$
$$a_i < 0, \qquad i \in \mathcal{Z} \tag{3.4b}$$

holds. When (3.4a) holds the solution found in the set $\mathcal{S}$ is an optimal solution, since no improvements can be made to decrease $f(\mathbf{c}_\mathcal{S})$. The requirement (3.4b) implies that if c_i ($i \in \mathcal{Z}$) would be changed from 0 to a positive value, $f(\mathbf{c})$ would increase since $f'(c_i) > 0$. Since $\mathbf{a}$ is formed as the product of the dictionary and the residual in line 24, 3.4a will be fulfilled for all the elements in $\mathcal{Z}$. Line 8 checks then if (3.4b) is fulfilled which means that the solution $\mathbf{c}$ is the optimal solution, if this is not the case, the element which improves $f(\mathbf{h})$ the most is added to $\mathcal{C}$ and removed from $\mathcal{Z}$. With this expanded set $\mathcal{S}$ a new LS solution $\mathbf{c}_\mathcal{S}$ is calculated. It is possible that this LS solution contains negative elements, which is corrected by the while loop started at line 14.
The value α as picked in line 15 is used to change the feasible solution $\mathbf{h}$ such that one or more positive values in the feasible solution become zero and the set $\mathcal{S}$ thus can be shrunk. The value of α is always positive and corresponds to the element $i \in \mathcal{S}$ where the difference $h_i - c_i$ is relatively the largest compared to h_i. The updated h_i will then be zero and i will thus be removed from $\mathcal{S}$, possibly with more elements $j \in \mathcal{S}$ for which $h_j = 0$.
After shrinking $\mathcal{S}$ a new least squares solution is calculated, which is then checked whether it is feasible. If this is the case, the feasible solution $\mathbf{h}$ is set equal to $\mathbf{c}$ and requirement (3.4a) is thus fulfilled. When $a_i < 0 \, \forall i \in \mathcal{Z}$, the requirement (3.4b) is fulfilled as well and the algorithm can thus be ended.
To make the algorithm more robust and prevent small $\mathbf{c}$-values the while-conditions in lines 8 and 14 can be changed to $a_i >$ `tol` and $c_j <$ `tol` respectively.
The convergence of this algorithm can be summarised in the following corollary from [68], where $|\mathcal{S}|$ denotes the number of elements in the set $\mathcal{S}$:

Corollary 3.0.1 *Let S be a subset of column indices of the dictionary matrix D such that $|\mathcal{S}| = L \leq N$. If the inequalities*

$$(\mathbf{x} - D_\mathcal{S}\mathbf{c}_\mathcal{S})^T \mathbf{d}_j < 0 \ \forall j \in \mathcal{S}^C \tag{3.5}$$

hold for the solution $\mathbf{c}_\mathcal{S} \in \mathbb{R}^J$ of the restricted NNLS problem over the column subset $\mathcal{S}$, then all solutions to the general NNLS have their supports contained in $\mathcal{S}$.

Simultaneous NNLS

The use of algorithms to solve joint problems as discussed in Section 2.6.2 will be discussed in a later section, however a small remark about the NNLS algorithm can be made here. The NNLS algorithm can be used to solve the (JNNLS) problem, but will not result in a solution with advantages of a joint solution. To do so the individual columns of X

are used as input for the NNLS problem resulting in the vectors which combine to the matrix C.
This can be computationally demanding, because of the pseudoinverses calculated in lines 12 and 19. This step can be made more efficient by saving the computed $D_{\mathcal{S}}^{\dagger}$ and reusing them when the same set $\mathcal{S}$ is used. An efficient way of doing so is proposed in [69].

3.2.2. Bayesian approach

In the paper [5] a Bayesian approach to performing a multiple component analysis with MRF is proposed. In this section the outlines of this method will be given, for more background the original paper is recommended. The authors mention that the method is aimed at the problem of partial volumes, where the components in the considered area have distinct boundaries and not a setting where the components are on a scale which are smaller than the diffusion distance. No prior knowledge of the individual tissue components is assumed, thus every tissue in the dictionary is equally likely on forehand.
In a Bayesian approach all the unknowns $\mathbf{c}$ are modelled as random variables with associated probability density functions. Based on that model a posterior density function is formed, which gives the probability for a certain combination of variables based on the measured values $\mathbf{x}$. It is then possible to determine which combination of variables is most likely.
First the construction of the posterior density function will be discussed, followed by the method to solve the problem. In this method the dictionary and signal can be complex and therefore the coefficients can be complex too. A non-negative solution is obtained by taking the absolute value of the found coefficients. All the dictionary signals and the measured signal are assumed to be normalised.
The posterior density function $\pi(\mathbf{c}|\mathbf{x})$ can be determined using Bayes' law:

$$\pi(\mathbf{c}|\mathbf{x}) \propto \pi(\mathbf{x}|\mathbf{c})\pi(\mathbf{c}), \tag{3.6}$$

where $\propto$ denotes proportionality. First the likelihood density $\pi(\mathbf{x}|\mathbf{c})$ can be determined. Assuming that the variable $\mathbf{c}$ is known, the probability for a certain measured signal with Gaussian noise with a constant variance is given by

$$\pi(\mathbf{x}|\mathbf{c}) \propto \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - D\mathbf{c}\|_2^2\right). \tag{3.7}$$

The values c_j are assumed to be independent and normally distributed with zero mean and variance θ_j. The prior density is

$$\pi(\mathbf{c}) = \frac{\det T_{\boldsymbol{\theta}}^{-1/2}}{2\pi} \exp\left(-\frac{1}{2}\|T_{\boldsymbol{\theta}}^{-1/2}\mathbf{c}\|^2\right), \tag{3.8}$$

with the covariance matrix $T_{\boldsymbol{\theta}} = \mathrm{diag}(\theta_1, ..., \theta_N)$. Since the variance is unknown as well, this is modelled as a collection of random variables as well. These are assumed to be

distributed according to a Gamma distribution with shape and scale parameters α and β respectively as:

$$\pi(\theta_j) \propto \theta_j^{\alpha-1} \exp\left(\frac{-\theta_j}{\beta}\right). \tag{3.9}$$

Using Bayes' formula the posterior density function can be written as:

$$\pi(\mathbf{c}, \boldsymbol{\theta}|\mathbf{x}) \propto \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - D\mathbf{c}\|_2^2 - \frac{1}{2}\|T_{\boldsymbol{\theta}}^{-1/2}\mathbf{c}\|^2 - \frac{1}{\beta}\sum_{j=1}^{N}\theta_j - \left(\alpha - \frac{3}{2}\right)\sum_{j=1}^{N}\log\theta_j\right). \tag{3.10}$$

To determine which $\mathbf{x}$ and $\boldsymbol{\theta}$ are most likely, the posterior density function (3.10) needs to be maximised, which is equivalent to minimising the negative logarithm of (3.10). This leads to the Bayesian minimisation problem:

$$\arg\min_{\mathbf{c},\boldsymbol{\theta}} \left\{ \frac{1}{2\sigma^2}\|\mathbf{x} - D\mathbf{c}\|_2^2 + \frac{1}{2}\|T_{\boldsymbol{\theta}}^{-1/2}\mathbf{c}\|^2 + \frac{1}{\beta}\sum_{j=1}^{N}\theta_j + \left(\alpha - \frac{3}{2}\right)\sum_{j=1}^{N}\log\theta_j \right\}. \tag{3.11}$$

A solution to this problem is determined is by alternatingly solving for $\boldsymbol{\theta}$ and $\mathbf{c}$. The initial solution is set to be the zero vector ($\mathbf{c}^{(0)} = \mathbf{0}$) and the initial dictionary the complete dictionary ($D^{(0)} = D$). The variance can be updated through an analytic solution each iteration k:

$$\theta_j^{(k)} = \beta\left(\eta + \left(\frac{(c_j^{(k-1)})^2}{2\beta} + \eta^2\right)^{1/2}\right), \tag{3.12}$$

where $\eta = (\alpha - 3/2)/2$. The next step is to solve for $\mathbf{c}^{(k)}$ with fixed θ. To do so, a conjugate gradient method for least squares[2] with prior conditioning and a change of variables are used:

$$w^{(k)} = \arg\min\left\{\|\mathbf{x} - D^{(k-1)}\left(T_{\boldsymbol{\theta}}^{(k)}\right)^{1/2}\mathbf{w}\|^2 + \mu\mathbf{w}^2\right\}. \tag{3.13}$$

To improve the sparsity of the solution a Tikhonov regularisation scheme is used here, depending on the regularisation parameter μ Then $\mathbf{c}$ is updated as

$$\mathbf{c}^{(k)} = \left(T_{\boldsymbol{\theta}}^{(k)}\right)^{1/2}\mathbf{w}^{(k)}. \tag{3.14}$$

A second measure to improve the sparsity of the solution and to improve the speed is pruning the dictionary. After each iteration the the columns corresponding to the smallest weights in $\mathbf{c}$ are removed. In this application the smallest 5% is pruned.

[2]Implemented based on the code from http://web.stanford.edu/group/SOL/software/cgls/

Remarks

The above described method needs many iterations before a sparse solution is obtained. In the original paper a running time of 12 seconds for each voxel is reported when 130 iterations are used to reduce the dictionary from 5970 to 8 entries. By revisiting the method and using compression of the dictionary it was possible to reduce the matching time with a factor of at least 50, while still using the pruning.
The complete method can be seen as a reweighted least squares method using Tikhonov regularisation. In this method the reweighting is based on the values of θ. Different reweighting schemes to do this are proposed in [64]. An ℓ_2-reweighted least squares method will therefore be considered in the comparison as well, using the weights

$$w_i = \frac{1}{|c_i| + \epsilon} \tag{3.15}$$

where epsilon is small (10^{-5}).
The choice of α seems to be depending on the tissue types considered, the used values in the analysis where 1.75 and 3.5, for β 0.1 is used. For the regularisation parameter $\mu = 0.01$ is used.
The method is compared to the fast iterative shrinking tresholding algorithm (FISTA) [70] which seems to give rather poor results, as observed in later analysis [6] as well, where with a poor performance it is meant that the obtained solution is not an accurate, sparse representation. This poor behaviour can be explained by the high coherence in the MRF dictionaries and therefore the non-uniqueness of the solution. In the typical FISTA or LASSO [71] applications it is assumed that the dictionary is incoherent, as based on wavelet or random Gaussian signals.

3.2.3. Reweighted ℓ_1-norm regularisation

In the paper by Tang et al. [6] a different method to find a multiple component solution is proposed and analysed. A first attempt is done by solving the problem (NNLSℓ_1) with the fast iterative shrinking tresholding algorithm (FISTA) [70] which is not able to solve the problem decently. It is therefore proposed to solve a sequence of weighted ℓ_1-norm regularised problems:

$$\min_{\mathbf{c}\in\mathbb{R}^N_{\geq 0}} \frac{1}{2}\|\mathbf{x} - D\mathbf{c}\|_2^2 + \lambda \sum_{i=1}^{N} w_i^{(k)} c_i, \tag{3.16}$$

where $w_i^{(k)}$ are the weights for each iteration. The entries of weights in the first iteration are 1. In the following iterations the weights are updated. Two different schemes are proposed for this. The first scheme, proposed in [72] approaches a ℓ_0-norm and promotes sparse solutions:

$$w_i^{(k)} := \frac{1}{\epsilon + |c^{(k-1)}|}, \; 1 \leq i \leq N, \tag{3.17}$$

where ϵ is a fixed parameter to prevent division by zero. This parameter should be set slightly smaller than the smallest expected nonzero value of $\mathbf{c}$. This first reweighting

method can be related to the FOCUSS method by Gorodnitsky and Rao [73], where a reweighted ℓ_2-norm regularisation is proposed with $\epsilon = 0$.
The second reweighting scheme, originally proposed in [64], is a non-separable reweighting scheme in which the correlation between dictionary entries is considered. The scheme is of the form:

$$w_i^{(k)} := \left(\mathbf{d}^{iT}\left(\epsilon I_M + DC^{-1}WD^T\right)^{-1}\mathbf{d}^i\right)^{\frac{1}{2}} \tag{3.18a}$$

$$C := \operatorname{diag}\left(\mathbf{c}^{(k-1)}\right), \tag{3.18b}$$

$$W := \operatorname{diag}\left(\mathbf{w}^{(k-1)}\right). \tag{3.18c}$$

The parameter ϵ is used to prevent the matrix $\left(\epsilon I_M + DC^{-1}WD^T\right)$ of being singular. The weights produced by this scheme are more smooth in the sense that weights for similar dictionary signals are similar.
To solve the reweighted non-negative ℓ_1 minimisation problem (3.16), without the reweighting, at first an interior point method is proposed in [6] using a log-barrier function [74, 75]. In this method a sequence of cost functions is used

$$\phi_t(\mathbf{c}) := \frac{1}{2}t\|D\mathbf{c} - \mathbf{x}\|_2^2 + t\lambda\sum_{i=1}^{N} c_i - \sum_{i=1}^{N}\log c_i, \tag{3.19}$$

where $t > 0$ is a parameter which varies from 0 to ∞. The minimisers of ϕ_t form the central path of this interior point scheme. The interior point scheme finds the minimisers $\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, ...$ for increasing values of t which leads to a, sparse non-negative solution. This is done by iteratively solving the Newton system

$$\left(Z + tD^TD\right)\left(\mathbf{c}^{(k)} - \mathbf{c}^{(k)}\right) = -\nabla\phi_t\left(\mathbf{c}^{(k-1)}\right) \tag{3.20}$$

where

$$Z := \operatorname{diag}\left(\frac{1}{\left(c_1^{(k)}\right)^2}, ..., \frac{1}{\left(c_N^{(k)}\right)^2}\right). \tag{3.21}$$

This approach is proposed in [76, 77]. Solving (3.20) is the computationally most intensive step, which can be accelerated by using a preconditioner or the Woodburry inversion lemma [78]. This method imposes the non-negativity by the log-barrier, which pushes the solution away from 0, at the same the sparsity has to be promoted by the regularisation pushing small values in the solution to zero. This thus forms a complex balancing exercise. The code is implemented based on an example as given in [79].
The proposed unweighted method is applied to the weighted problem by applying a change of variables; $\mathbf{c}' = W\mathbf{c}$ and $D' = DW^{-1}$.

3.2.4. Non-Negative Orthogonal Matching Pursuit

The Non-Negative Orthogonal Matching Pursuit (NNOMP) [61, 62] is an adaptation of Orthogonal Matching Pursuit (OMP) [80, 81] which tries to solve the problem (NNℓ_0a) as introduced in Section 2.6.1. NNOMP is meant to solve a problem with a real signal and a real dictionary. This section will start with a short description of the OMP algorithm, followed by a description of the NNOMP algorithm, which will also address the differences with OMP.
Orthogonal Matching Pursuit tries to find a solution to

$$\min_{\mathbf{c}\in\mathbb{C}^N} \|D\mathbf{c} - x\|_2^2 \quad \text{subject to } \|\mathbf{c}\|_0 \leq K. \tag{3.22}$$

This is done by obtaining a the best selection of atoms from a normalised dictionary D for the signal $\mathbf{x}$ by expanding the elements of the basis through several iterations.

Algorithm 3 The OMP algorithm [80]

INPUT: $\mathbf{x}$ - a complex valued signal
D - a complex valued dictionary
K - the maximal number of components
OUTPUT: $\mathbf{c}$ - the returned solution to (NNℓ_0a)

$\mathcal{S} \leftarrow \emptyset$ ▷ Support set of the solution
$\mathbf{r} \leftarrow \mathbf{x}$ ▷ Residual
$\mathbf{a} \leftarrow D^T\mathbf{r}$
while $|\mathcal{S}| < K$ and $\exists\, i \in \mathcal{S}^C : |a_i| > 0$ **do**
  $\mathbf{c} \leftarrow \mathbf{0}$
  $i^* \leftarrow \arg\max |\mathbf{a}|$ ▷ The element used for the expansion
  $S \leftarrow \mathcal{S} \cup i^*$
  $\mathbf{c}_\mathcal{S} \leftarrow D_\mathcal{S}^\dagger \mathbf{x}$ ▷ Gives the least squares solution using the basis $\mathcal{S}$
  $\mathbf{r} \leftarrow \mathbf{x} - D\mathbf{c}$
  $\mathbf{a} \leftarrow D^T\mathbf{r}$
end while

A description of the algorithm is given in Algorithm 3. Each iteration a residual $\mathbf{r}$ is determined which is used to determine the atom i^* which should be added to the basis $\mathcal{S}$. This selection is done according to the following selection rule:

$$i^* = \underset{i\in\{1,\ldots,N\}}{\arg\max} |\mathbf{d}^{iT} \cdot \mathbf{r}|. \tag{3.23}$$

After determining this new basis the least squares solution is determined as $\mathbf{c}_\mathcal{S} = D_\mathcal{S}^\dagger \mathbf{x}$, $\mathbf{c}_{\mathcal{S}^C} = 0$) and the residual is updated as $\mathbf{r} = \mathbf{x} - D_\mathcal{S}\mathbf{c}_\mathcal{S}$. The number of iterations determines the size of the basis in this method, unless the iterations are stopped earlier because no more significant improvements can be made ($|\mathbf{d}^{iT} \cdot \mathbf{r}| < \delta$ for some small δ).

The NNOMP algorithm is given in Algorithm 4 omitting line 10. In this description the set $\mathcal{S}$ contains the indices of the used basis, where $\mathcal{S}^C$ is the complementary set, containing the indices not part of the basis. The NNOMP algorithm is very similar to the OMP algorithm, it differs however in two steps because of the non-negativity constraint. The first difference is the selection rule in line 5, where in OMP the absolute value is used, here the largest positive value of $\mathbf{a}$ as formed in line 3 is selected. The second difference is that the non-negative least squares solution in the basis S is sought instead of the least squares solution in line 7, as a result of this the method can only deal with real valued dictionaries and signals.

The NNLS solution can be found using the NNLS-algorithm as discussed in Section

Algorithm 4 The NNOMP (without line 10) and SNNOMP algorithm [62]

INPUT: $\mathbf{x}$ - a real valued signal
D - a real valued dictionary
K - the maximal number of components
OUTPUT: $\mathbf{c}$ - the returned solution to (NNℓ_0a)

1: $\mathcal{S} \leftarrow \emptyset$ ▷ Support set of the solution
2: $\mathbf{r} \leftarrow \mathbf{x}$ ▷ Residual
3: $\mathbf{a} \leftarrow D^T\mathbf{r}$
4: **while** $|\mathcal{S}| < K$ and $\exists\, i \in \mathcal{S}^C : a_i > 0$ **do**
5: $i^* \leftarrow \arg\max \mathbf{a}$ ▷ The element used for the expansion
6: $S \leftarrow \mathcal{S} \cup i^*$
7: $z \leftarrow \arg\min_{\mathbf{z}\geq 0} \|\mathbf{x} - D_{\mathcal{S}}\mathbf{z}\|_2^2$ ▷ Using the NNLS algorithm with a warm start
8: $\mathbf{r} \leftarrow \mathbf{x} - D_{\mathcal{S}}\mathbf{z}$
9: $\mathbf{a} \leftarrow D^T\mathbf{r}$
10: $\boxed{\mathcal{S} \leftarrow \mathcal{S}(\text{supp}(\mathbf{z}))}$
11: **end while**
12: $\mathbf{c}_{\mathcal{S}} \leftarrow \mathbf{z}$
13: $\mathbf{c}_{\mathcal{S}^C} \leftarrow \mathbf{0}$

3.2.1. The improvement by Nguyen et al. on the NNOMP algorithm is to note that the computational efficiency can be increased by using a different initial solution than the 0-vector in the NNLS algorithm. The initial solution to be used is the vector $\begin{bmatrix}\mathbf{z}\\0\end{bmatrix}$, which makes it possible to use information from the previous iteration, leading to a faster solution to the NNLS problem.
Because of the non-negativity constraint it is possible that some of the elements of $\mathbf{z}$ become 0. This behaviour does not occur in the OMP algorithm. These vanishing elements can thus be removed from the set $\mathcal{S}$, without changing the solution $\mathbf{c}$. This step of shrinking the set $\mathcal{S}$ to the support of $\mathbf{z}$ is done in line 10 and leads to the support shrinkage non-negative orthogonal matching pursuit (SNNOMP) algorithm. This

adjustment leads to a smaller set $\mathcal{S}$ while having the same or a smaller error, which is in general a desirable behaviour, but it can increase the number of iterations drastically.

3.2.5. SNNOMP-2

When one compares the descriptions of NNLS algorithm and the SNNOMP algorithm, it can be seen that these two are very similar. It is therefore even possible to combine the two algorithms into one algorithm, combing these two would simply mean that the NNLS algorithm is stopped when $\mathcal{S}$ has a certain size. This leads to the SNNOMP-2 algorithm, Algorithm 5, which tries to find a solution for the problem (NNℓ_0a), just as the SNNOMP algorithm does. This is also proposed in [66] in the area of non-negative matrix factorisation.

However, the earlier stop of the SNNOMP-2 algorithm when $|\mathcal{S}| = K$ is reached, can mean that the optimal solution was not found at the moment of termination. To find the optimal solution one should keep track of the optimal solution for a certain basis size, while different options are sought. However, this is computationally highly inefficient.

Other OMP variants

Improvements of the OMP algorithm in the setting of compressed sensing are regularised OMP (ROMP [82]), stage-wise OMP (StOMP) [83] and compressive sampling matching pursuit (CoSaMP) [84]. These algorithms select multiple atoms at once to add to the basis, making the methods possibly faster. However, these algorithms are aimed at systems where the dictionaries are highly incoherent, which is in general not the case in the MR fingerprinting problem setting.

In these algorithms one starts with a basis of size K, this basis can be found with the normal OMP algorithm or by taking the largest K elements in the selection of equation 3.23. A solution using this basis is then sought and the corresponding residual is determined. This basis is then expanded with the $\tilde{K}$ atoms with the largest inner product with the residual. This expanded basis is then used to find a new solution which is used to restrict the basis to the K largest coefficients in this solution, leading to a basis of size K. This step can be repeated several times, leading to better results for incoherent dictionaries.

In the rest of this thesis a variant of this will be used where first a basis of size K is found using the SNNOMP algorithm, which is then expanded with $\tilde{K} = 1$ element each iteration. This variant will here be called regularised NNOMP (RNNOMP).

3.3. Joint sparsity methods

As described in Section 2.6.2 improvements could be made by using a method which uses not only information from a single voxel, but information of other voxels as well. This means that one tries to solve the problem (JNNLS), where a joint sparsity constraint could be added to retrieve a sparse solution for every voxel, using the same basis for all of the voxels. This section will discuss several methods to solve such a joint problem.

Algorithm 5 A combination of the SNNOMP and NNLS algorithm

INPUT: $\mathbf{x}$ - a real valued signal
D - a real valued dictionary
K - the maximum number of components
OUTPUT: $\mathbf{c}$ - a non-negative least square solution to (NNℓ_0a)

$\mathbf{h} \leftarrow \mathbf{0}$
$\mathcal{Z} \leftarrow \{1, ..., N\}$ ▷ The same a $\mathcal{S}^C$
$\mathcal{S} \leftarrow \emptyset$
$\mathbf{r} \leftarrow \mathbf{x}$
$\mathbf{a} \leftarrow D^T\mathbf{r}$
while $|\mathcal{Z}| > 0$ and $|\mathcal{S}| \leq K$ and $\exists i \in \mathcal{Z} : a_i > 0$ **do** ▷ Difference with the NNLS alg.
    $i^* \leftarrow \arg\max \mathbf{a}$
    $Z \leftarrow \mathcal{Z} \backslash i^*$
    $S \leftarrow \mathcal{S} \cup i^*$
    $\mathbf{c}_\mathcal{S} \leftarrow D_\mathcal{S}^\dagger \mathbf{x}$
    $\mathbf{c}_\mathcal{Z} \leftarrow 0$
    while $\exists j \in \mathcal{S} : c_j < 0$ **do**
        $\alpha \leftarrow \min_{k \in \mathcal{S}} h_k/(h_k - c_k)$
        $\mathbf{h} \leftarrow \mathbf{h} + \alpha(\mathbf{c} - \mathbf{h})$
        $\mathcal{Z} \leftarrow \{i | h_i = 0\}$ ▷ Support shrinkage step
        $\mathcal{S} \leftarrow \{i | h_i > 0\}$
        $\mathbf{c}_\mathcal{S} \leftarrow D_\mathcal{S}^\dagger \mathbf{x}$
        $\mathbf{c}_\mathcal{Z} \leftarrow 0$
    end while
    $\mathbf{h} \leftarrow \mathbf{c}$
    $\mathbf{r} \leftarrow \mathbf{x} - D\mathbf{h}$
    $\mathbf{a} \leftarrow D^T\mathbf{r}$
end while

In these approaches all the relevant signals can be considered at once in the matrix X or X might only contain one segment based on the segments as proposed in 3.1.3. After this segmentation it can be better to relax the boundaries between the segments as discussed in the next Section 3.3.3.

3.3.1. Joint sparsity using NNOMP

In the paper by Tropp et al. [42] the simultaneous orthogonal matching pursuit (S-OMP) algorithm is proposed. This algorithm is based on the OMP algorithm, however it tries to solve the problem (SSAa), which includes the joint sparsity constraint, using a small common basis for the representation of the measured signals. Just as with OMP the algorithm tries to form a basis consisting of a maximum number of elements which gives an optimal representation C of the measured signals X.
Each iteration the residual $R = X - DC$ is calculated, where C is the solution found in that iteration, which is initiated as a zero-matrix. The matrix R consists of the vectors representing the individual residuals, $\mathbf{r}^1, ..., \mathbf{r}^J$ of the individual signals $\mathbf{x}^1, ..., \mathbf{x}^J$ with sparse representations $\mathbf{c}^1, ..., \mathbf{c}^J$. Based on these residuals the basis $\mathcal{S}$ is expanded with the element

$$i^* = \underset{i \in \{1,...,N\}}{\arg\max} \|R^* \mathbf{d}^i\|_p^p, \tag{3.24}$$

where R^* denotes the complex conjugate of R and p is the order of the norm used for the selection. In the paper [42] $p = 1$ is used,this results in $\|R^* d_i\|_p^p = \sum_{j=1}^J \mathbf{r}^{jT} \dot{d}_i$, which will be taken here as well. When the basis is expanded, the orthogonal projector P onto the span of the selected dictionary atoms can be calculated. This projector can be used to determine the newest approximation to the sparse solution.
This S-OMP approach can be altered to a joint non-negative OMP (JNNOMP) algorithm, although this has not been done before for as far as we know. This algorithm is given in Algorithm 6. In the J-NNOMP the selection of the new atom is based on the positive values of A, the negative values are set to 0. To find the voxel-wise solutions the NNLS algorithm is used with a warm start, just as in the NNOMP algorithm. Including the support shrinkage as in the SNNOMP algorithm, could be done, but it is highly unlikely that for each voxel a previously selected atom is not used. The S-OMP was compared to a joint version of CoSaMP (see Section 3.2.5) called M-CoSaMP in [85] leading to the conclusion that for dictionaries with increased correlation the S-OMP algorithm is preferred. The main reason for the conclusion is the similarity in solutions and the poor convergence behaviour for the M-CoSaMP, where S-OMP stops after K iterations. However, these considerations were for methods allowing negative solutions.
A possible way to implement the M-CoSaMP is by stopping after a fixed number of iterations or when the found solution basis is not changed by adding the extra basis element. This modification, including non-negativity, to JNNOMP will thus be considered as well as Joint Regularised NNOMP (JRNNOMP), where the found basis of size K is expanded with the best $\tilde{K}$ (normally taken equal to 1) elements, leading to a new solution which is then restricted again to a basis of size K. This whole procedure is repeated at most T times. This non-negative version is proposed here in Algorithm 7 for

Algorithm 6 The joint NNOMP algorithm, which is a combination of the S-OMP and NNOMP algorithms as developed in this thesis.

$X = \begin{bmatrix}\mathbf{x}^1 & \dots & \mathbf{x}^J\end{bmatrix}$ - real valued signals
INPUT: D - a real valued dictionary
K - the maximal number of components
OUTPUT: C - the returned solution to (SSAa)

$\mathcal{S} \leftarrow \emptyset$ ▷ Support set of the solution
$R \leftarrow X$ ▷ Residual
$A \leftarrow D^T R$ ◇[3]
$A \leftarrow \max\{0, A\}$ ▷ Set all negative values of A to 0
while $|\mathcal{S}| < K$ and $\exists\, i \in \mathcal{S}^C, j \in \{1, ..., J\} : a_{i,j} > 0$ **do**
  $i^* \leftarrow \arg\max_{i \in \{1,\dots,M\}} \sum_{j=1}^{J} a_{ij}$ ▷ The element used for the expansion
  $\mathcal{S} \leftarrow \mathcal{S} \cup i^*$
  for j in `range`(J) **do**
    $\mathbf{z}^j \leftarrow \arg\min_{\mathbf{z} \geq 0} \|\mathbf{x}^j - D_{\mathcal{S}}\mathbf{z}\|_2^2$ ▷ Using the NNLS algorithm with a warm start for each signal ◇
  end for
  $Z = \begin{bmatrix}\mathbf{z}^1 & \dots & \mathbf{z}^J\end{bmatrix}$
  $R \leftarrow X - D_{\mathcal{S}} Z$ ◇
  $A \leftarrow D^T R$ ◇
  $A \leftarrow \max\{0, A\}$ ▷ A only contains non-negative values
end while
$C_{\mathcal{S}} S \leftarrow Z$
$C_{\mathcal{S}^C} \leftarrow \mathbf{0}$

the first time. Here $\arg\underset{K}{\text{partition}}\,\mathbf{v}$ returns the indices of the K largest elements in v.
The S-OMP algorithm has been compared in the paper [86] with the iterative support detection based joint sparsity algorithm (ISDJS). In this comparison the S-OMP algorithm is preferred for signals with a very small joint basis, which is the case in this project.

3.3.2. Sparsity Promoting Iterative Joint NNLS (SPIJN)

Another way to impose joint sparsity restrictions is by using a reweighting scheme as in Section 3.2.3. The first proposed method to do so is M-FOCUSS, based on FOCUSS [73] (FOCal Underdetermined System Solver) as mentioned before. The FOCUSS algorithm can be expanded to an implementation to solve the problem (2.20) with $p \leq 1$ and $q = 2$. This leads to the M-FOCUSS algorithm [41], dealing with multiple measurement vectors. This algorithm can be seen as a reweighted least squares solver. It uses a specific scheme of weights and is calculated iteratively. The algorithm for iteration k is summarised as

$$W_{k+1} = \text{diag}((\mathbf{w})_{k+1}^{1/2}) \tag{3.25a}$$

$$Q_{k+1} = D_{k+1}^{\dagger} X, \text{ where } D_{k+1} = DW_{k+1} \tag{3.25b}$$

$$C_{k+1} = W_{k+1} Q_{k+1}, \tag{3.25c}$$

where $(w_i)_{k+1}$ are the weights for each dictionary atom, which will be discussed later. The algorithm is terminated once a convergence criterion is satisfied, e.g.

$$\frac{\|C_{k+1} - C_k\|_F}{\|C_k\|_F} < \delta, \tag{3.26}$$

where δ is a chosen tolerance, in the introducing paper $\delta = 0.01$ was chosen.
The M-FOCUSS algorithm gives weight to the atoms which are used often and penalises the use of atoms which are not used often.
Different variants on this reweighting for the FOCUSS algorithm have been proposed [64, 87] which can also be applied to the M-FOCUSS algorithm.
The first variant [85] would be

$$(w_i)_{k+1} = \|C_{k,[i,:]}\|_2 + \epsilon_k, \tag{3.27}$$

where $\epsilon_k > 0$ (here has been used $\epsilon = 10^{-4}$) is a parameter used to prevent division by zero, which can be varied each iteration and is recommended to be decreasing, staying larger than zero. $C_{k,[:,i]}$ notates the ith row of the matrix C_k. This reweighing scheme will be referenced to as the first reweighting scheme from now on. The original M-FOCUSS algorithm sets $\epsilon = 0$. This second scheme will not be used often, since the division by zero is unwanted. A non-separable reweighting scheme [64], related to the second scheme proposed in [6] and given in Equation (3.18) is:

$$(w_i)_{k+1} = \|C_{k,[:,i]}\|_2^2 + (w_i)_k - ((w_i)_k)^2 \, \mathbf{d}^{iT} \left(\epsilon I + D\tilde{W}_k D^T\right)^{-1} \mathbf{d}^i, \tag{3.28}$$

Algorithm 7 The joint RNNOMP algorithm (JRNNOMP), which is a combination of the S-OMP, ROMP and NNOMP algorithms, as developed in this thesis.

	$X = \begin{bmatrix}\mathbf{x}^1 & \dots & \mathbf{x}^J\end{bmatrix}$	-	a real valued signals
	D	-	a real valued dictionary
INPUT:	K	-	the maximal number of components
	$\tilde{K}$	-	the number of added components (normally 1)
	T	-	the maximum number of iterations

OUTPUT: C - the returned solution to (SSAa)

$\mathcal{S} \leftarrow \emptyset$ ▷ Support set of the solution
$R \leftarrow X$ ▷ Residual
$A \leftarrow D^T R$ ◇
$A \leftarrow \max\{0, A\}$ ▷ Set all negative values of A to 0
$k \leftarrow 0$ ▷ Counter for the number of iterations
while $k < T$ and $\exists\, i \in \mathcal{S}^C, j \in \{1, ..., J\} : a_{i,j} > 0$ **do**
    $\mathcal{I} \leftarrow \arg\operatorname{partition}_{\tilde{K}} \sum_{j=1}^{J} a_{ij}$ ▷ The $\tilde{K}$ elements used for the expansion
    $\mathcal{S}_0 \leftarrow \mathcal{S}$; $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{I}$
    for j in `range`(J) **do**
        $\mathbf{z}^j \leftarrow \arg\min_{\mathbf{z} \geq 0} \|\mathbf{x}^j - D_{\mathcal{S}}\mathbf{z}\|_2^2$ ▷ Using the NNLS algorithm with a warm start for each signal ◇
    end for
    $Z \leftarrow \begin{bmatrix}\mathbf{z}^1 & \dots & \mathbf{z}^J\end{bmatrix}$
    if $\mathcal{S} > K$ **then**
        for p in `range`$(|S|)$ **do**
            $v_p \leftarrow \sum Z_{[:,p]}$ ▷ Taking the row sum of Z
        end for
        $\mathcal{T} \leftarrow \arg\operatorname{partition}_{K} \mathbf{v}$ ▷ The indices of the largest $\tilde{K}$ elements in $\mathbf{v}$ are determined
        $S \leftarrow S_{\mathcal{T}}$; $Z \leftarrow Z_{\mathcal{T}}$
        if $\mathcal{S} = \mathcal{S}_0$ **then**
            `Stop while loop`
        end if
    end if
    $R \leftarrow X - D_{\mathcal{S}} Z$ ◇
    $A \leftarrow \max\{0, D^T R\}$ ▷ A only contains non-negative values ◇
end while
for j in `range`(J) **do**
    $\mathbf{z}^j \leftarrow \arg\min_{\mathbf{z} \geq 0} \|\mathbf{x}^j - D_{\mathcal{S}}\mathbf{z}\|_2^2$ ◇
end for
$C_{\mathcal{S}} S \leftarrow Z$
$C_{\mathcal{S}^C} \leftarrow \mathbf{0}$

where $\tilde{W}_k$ is a diagonal matrix with the vector $\mathbf{w}_k$ on the diagonal. In this reweighting scheme the coherence between different dictionary elements is taken into consideration, punishing the selection of similar atoms less than the selection of non-similar atoms. To this reweighting scheme will be referenced as the third reweighting scheme.
The schemes as provided here are all based on general least squares solutions. It is however easy to change the algorithm to a non-negative algorithm, by replacing the pseudo-inverse with the NNLS solver. It was not possible to find such an application in literature, which is surprising since the change can be made rather easy.
To enable the reweighing schemes to deal with the noise present in the signals, it is recommended to use some form of regularisation. The most efficient regularisation is the ℓ_1^2 regularisation based on NNLSℓ_1^2. This regularisation method is less computationally intensive than ℓ_1 or ℓ_2 (Tikhonov) regularisation, since the ℓ_1^2 regularisation requires only a single added row to the dictionary. This regularisation leads a new algorithm, from now on called Sparsity Promoting Iterative Joint NNLS abbreviated as SPIJN. The SPIJN algorithm iterations can be summarised as:

$$W_{k+1} = \text{diag}((\mathbf{w})_{k+1}^{1/2}), \tag{3.29a}$$

$$\tilde{D}_{k+1} = \begin{bmatrix} DW_{k+1} \\ \lambda \mathbf{1}^T \end{bmatrix}, \tag{3.29b}$$

$$\tilde{X} = \begin{bmatrix} X_{k+1} \\ \mathbf{0}^T \end{bmatrix}, \tag{3.29c}$$

$$\tilde{C}_{k+1} = \text{JNNLS}(\tilde{X}, \tilde{D}),, \tag{3.29d}$$

$$C_{k+1} = W_{k+1}\tilde{C}_{k+1}, \tag{3.29e}$$

where JNNLS(X, D) gives the joint NNLS solution to the problem $\|X - DC\|_F^2$.
The discussed reweighting schemes are compared in the paper [64] leading to the conclusion that the non-separable scheme 3.28 shows the best results. In this comparison also computationally more demanding ℓ_1 methods [72] and the grouped Lasso method [88] are discussed and compared, showing no better results for basis pursuit problems. In these comparisons the S-OMP algorithm was not considered and the dictionaries were highly non-correlated, in contrast to the MRF-implementations. This thus leaves room for further research.
A method to speed up this algorithm would be by pruning the dictionary by removing elements which are unused. Dictionary atoms can be considered to be unused when $\|C_{k,[i,:]}\|_1 < \tilde{\delta}$, where $\tilde{\delta}$ is a parameter with a small value, here taken to be $10^{-8} \times J$ (J is the number of voxels considered). This step is justified if the unused dictionary atoms are not "reactivated" later on. The pruning can either be performed every iteration or after a fixed number of iterations. While testing this on data the pruning did not influence the end results when the first reweighting scheme was used and accelerated the computations significantly. For the third reweighting scheme it was not possible to draw immediate conclusions.
The complete SPIJN algorithm including the pruning step is given in Algorithm **??**.

Algorithm 8 The Sparsity Promoted Iterative Joint NNLS (SPIJN) algorithm to perform a multi-component analysis for an MRF measurement.

INPUT:
- X - a real valued signals
- D - a real valued dictionary
- T - maximum number of iterations
- $p, \tilde{\delta}$ - pruning takes place at this iteration, using this treshhold
- δ - convergence treshhold
- λ - regularisation parameter

OUTPUT: C - the matched solution

$k \leftarrow 1$ ▷ Counter for the number of iterations
$C_1 \leftarrow \arg\min_{C \in \mathbb{R}^{N \times J}_{\geq 0}} \|X - DC\|_F^2$ ▷ Initial solution ◇
$\mathbf{w}_1 \leftarrow \mathbf{1}$
$d \leftarrow 1$
while $k \leq T$ and $d > \delta$ **do**
    if $k = p$ **then** Prune D, C and $\mathbf{c}$ according to $\|C_{k,[i,:]}\|_1 < \tilde{\delta}$
    end if
    $w_{k+1,i} \leftarrow \|\mathbf{c}_k^i\|_2 + \epsilon \ \forall i \in \{1, ..., N\}$
    Calculate $\mathbf{w}_{k+1}$ from C and $\mathbf{w}_k$ with the desired scheme
    $\tilde{D}_{k+1} \leftarrow \begin{bmatrix} DW_{k+1} \\ \lambda \mathbf{1}^T \end{bmatrix}$
    $\tilde{X} \leftarrow \begin{bmatrix} X_{k+1} \\ \mathbf{0}^T \end{bmatrix}$
    $\tilde{C}_{k+1} \leftarrow \arg\min_{C \in \mathbb{R}^{N \times J}_{\geq 0}} \|\tilde{X} - \tilde{D}C\|_F^2$ ◇
    $C_{k+1} \leftarrow W_{k+1}\tilde{C}_{k+1}$
    $d \leftarrow \frac{\|C_{k+1} - C_k\|_F}{\|C_k\|_F}$
    $k \leftarrow k + 1$
end while
if $k > p$ **then**
    Fill C_k with zeros at pruned dictionary atoms.
end if
$C \leftarrow C_k$

3.3.3. Regrouped approach

When the JNNOMP or JRNNOMP algorithm are applied on the different segmented groups, hard boundaries between the segments exist. This is not a wanted behaviour since the boundary zones between tissue types are exactly the regions of interest.
When the JNNOMP or JRNNOMP is applied on a segmented image, for each group p ($p = 1, ..., P$) K indices are selected to form a basis $D_{\mathcal{S}_p}$. In total at most $K \cdot P$ atoms are selected to form these bases. These sets $\mathcal{S}_p$ can be combined to one set $\mathcal{S} = \cup_{p=1}^{P} \mathcal{S}_p$. This combined set can be used to form the basis $D_{\mathcal{S}}$, which can be used to solve the (JNNLS) problem. Since the basis is expanded, the solution can only be improved, with the drawback that the basis is larger.
This regrouping leads to a solution where the basis can be found easily and the boundaries between the segments disappear.

3.3.4. Search around

In some of the previously proposed methods it was attempted to restrict the large dictionary to only a few dictionary atoms, forming a small basis, which hopefully represents the present tissues. In these approaches it was assumed that the tissue types have the same T_1 and T_2 combination in each voxel. This is not necessarily the case, which makes this framework possibly a bit too rigid.
A way to improve the results when such a small basis $\mathcal{S}_0 = \{s_1, ..., s_K\}$ with weights $\mathbf{c}_S$ has been found, is by searching around these T_1 and T_2 combinations. In this approach the dictionary atoms $\mathcal{C}$ with comparable relaxation times to the values of atom s_i are selected and used to form the dictionary $D_{\mathcal{C}}$. In Figure 3.5 an example of the considered dictionary atoms is shown. This small dictionary $\tilde{D}_{s_i}$ is used to find the best atom to replace the original atom s_i in the set $\mathcal{S}$.
The reproduced signal is given by

$$\tilde{\mathbf{x}} = \sum_{s \in \mathcal{S}} c_s \mathbf{d}^s. \tag{3.30}$$

The approximation error is thus given by

$$\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}}. \tag{3.31}$$

The goal is to find a better set of atoms S^* such that the error decreases. This can be done by trying to improve each of the atoms of $\mathcal{S}$. By performing a single component matching based on dictionary $\tilde{D}_{s_i}$ with the signal

$$\bar{\mathbf{y}}_i = c_{\mathbf{s}_i} \mathbf{d}^{s_i} + \mathbf{e} \tag{3.32}$$

a new atom s_i^* can be found and added to $\mathcal{S}^*$. This set $\mathcal{S}^*$ can be used to find a new NNLS solution. When this is done the error can be determined again and this procedure can be repeated for all the elements in $\mathcal{S}$. In this method it matters however in which order the atoms in S are updated. A logical choice would be to start with updating

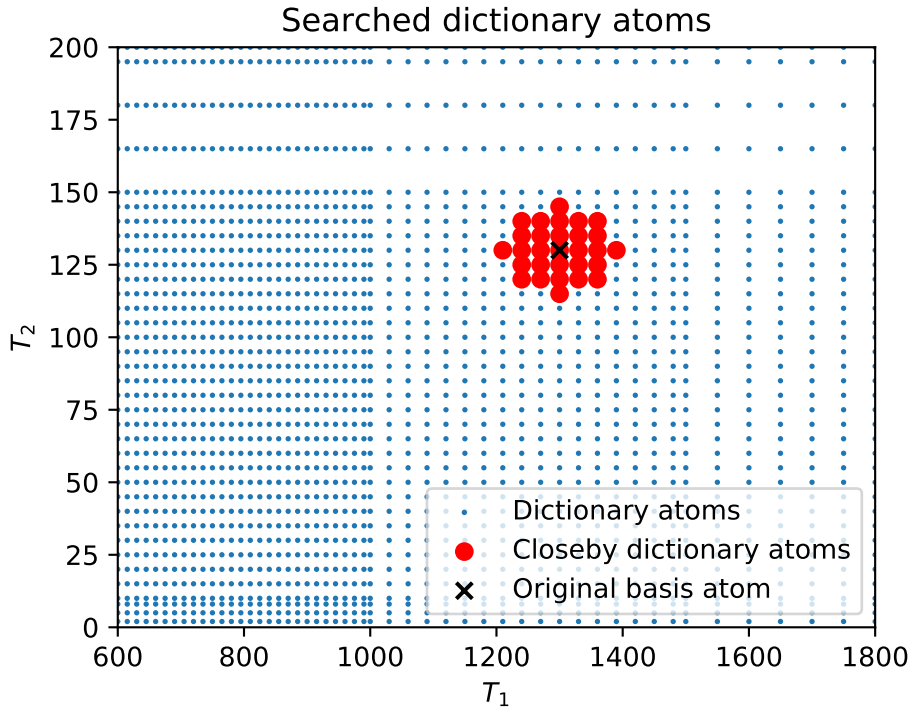


Figure 3.5.: The considered dictionary atoms to replace the originally selected atom. The small dictionary contains the atoms at a distance of at most 3 on the grid formed by the dictionary atoms.

s_1 trying to compensate for error $\mathbf{e}_1$, then s_2 compensating error $\mathbf{e}_2$ etc. However, one can also choose to start with s_2 compensating error $\tilde{\mathbf{e}}_1$ and updating s_1 in the next step based on $\tilde{\mathbf{e}}_2$. In this second step $\mathbf{e}_1 \neq \tilde{\mathbf{e}}_2$ and thus probably leads to a different s_1^*.
To prevent this order dependency, it is preferred to first update all the elements in $\mathcal{S}$, which then leads to a new non-negative least squares problem. Updating all the elements of $\mathcal{S}$ can again be done by performing a single component matching with dictionary $\tilde{D}_{s_i}$ on the signal $\bar{\mathbf{y}}_i$ as in Equation 3.32, this could however lead to unwanted behaviour as well. When several of the new indices s_i^* are able to completely remove the error on their own, the combination of them would lead to a solution which is compensating too much in the direction of the error. To prevent this behaviour the signal

$$\mathbf{y}_i = c_{s_i}\mathbf{d}^{s_i} + \frac{\mathbf{e}}{K} \tag{3.33}$$

is used for the single component matching. This whole procedure can be repeated several times, until the set $\mathcal{S}$ is stable or when a maximum number of iterations is reached. This leads to algorithm 9, where $\text{NNLS}(\mathbf{x}, D)$ solves the problem NNLS with the NNLS algorithm as given before.

Algorithm 9 An algorithm to search around an initial basis to find a better basis.

INPUT:
- $\mathbf{x}$ - a real valued signal
- D - a real valued dictionary
- $\mathcal{S}$ - the indices of the initial basis
- T - number of iterations

OUTPUT:
- $\mathbf{c}$ - the improved solution
- $\mathcal{S}^*$ - the improved basis

$K \leftarrow |\mathcal{S}|$
$k \leftarrow 0$ ▷ Counter for the number of iterations
while $k < T$ **do**
  $\mathbf{c}_{\mathcal{S}^*} \leftarrow \mathbf{0}$
  $\mathbf{c}_{\mathcal{S}} \leftarrow \text{NNLS}(\mathbf{x}, D_{\mathcal{S}})$
  $S^* \leftarrow \emptyset$
  $\mathbf{e} \leftarrow \mathbf{x} - \sum_{s \in \mathcal{S}} c_s \mathbf{d}^s$
  for $s \in \mathcal{S}$ **do**
    Determine neighbouring dictionary atoms $\mathcal{C}$ of atom s
    $\mathbf{y} \leftarrow \mathbf{c}_s \mathbf{d}^s + \frac{1}{K}\mathbf{e}$
    $s^* \leftarrow \arg\max_{i \in \mathcal{C}} \mathbf{d}^{iT}\mathbf{y}$
    $\mathcal{S}^* \leftarrow \mathcal{S}^* \cup s^*$
  end for
  $k \leftarrow k + 1$
  $\mathbf{c}_{\mathcal{S}^*} \leftarrow \mathbf{0}$
  $\mathbf{c}_{\mathcal{S}} \leftarrow \text{NNLS}(\mathbf{x}, D_{\mathcal{S}^*})$
  $\mathcal{S} \leftarrow \mathcal{S}^*$
end while
$\mathbf{c}^*_{\mathcal{S}} \leftarrow \mathbf{z}$
$\mathbf{c}_{\mathcal{S}^{*C}} \leftarrow \mathbf{0}$

3.3.5. Remarks

Disadvantages of the joint approach might be that it is more likely to miss tissue types which are only present in a small number of voxels and would therefore be suppressed by the restriction to have a small number of components. In the SPIJN algorithm this would be caused by choosing a high regularisation parameter. The J(R)NNOMP have a similar problem because the number of tissue has to be set on forehand, making it likely to identify non-existing tissues or mixing tissues into one.
With the introduction of the joint sparsity constraint, the masking becomes more important. When for example muscle on the outer part of the head is included and not masked, it will be necessary to include a component in the matching representing this tissue.

3.4. Fixed parameters for large dictionaries

In the previous sections it was assumed that the signal only depends on the relaxation times T_1 and T_2 and that the dictionary only contained these combinations of parameters. However, to obtain better result it can be necessary to compensate for other unwanted effects such as inhomogeneities in the B_1 field by introducing an extra parameter in the signal calculation. In this section, it is discussed how to include such extra parameters in the methods discussed before. Here B_1 inhomogeneities are considered, but other voxel dependent system parameters can be included as well.
The relative error of the B_1 field can be taken into account when calculating the dictionary containing the signal behaviour. This thus leads to a three dimensional parameter space, where the first two dimensions are formed by the T_1 and T_2 values and the third by the extra parameter, in this case relative error in the B_1 field. When a single component matching is performed, it will lead to a certain combination of the three parameters, where the relaxation times are giving information about the tissue and the third parameter only about a property of the voxel, but not about the tissues contained by the voxel.
When this dictionary containing the extra parameter values is used for the multi-component analysis, it would be possible to find multiple off-resonance values for one voxel, which seems incorrect, especially when the values are not close to each other. To make the preceding algorithms useful it is necessary to restrict the parameter space to one system parameter value for each voxel and determine the restricted dictionary from this for each voxel.
The relative error in the B_1 field for each voxel can either be determined from a separate measurement [89] or approximated from the measured MRF signal, for example using the single component matching as discussed before.
For the voxel-by-voxel methods this fixed parameter can be implemented relatively easily by restricting the dictionary to the determined parameter value without changing the algorithms any further. When the joint methods are used, this cannot be done that easily. At each step where the dictionary is used in combination with the signals or their residuals, the dictionary needs to be restricted to the voxel depending fixed parameter. Assuming

that the T_1 and T_2 values are ordered in the same way for different fixed parameter values, the row sums used for the atom selection in JNNOMP and JRNNOMP algorithms and for the reweighting in the SPIJN can still be used. In the SPIJN, JNNOMP and JRNNOMP algorithms the steps which need to be changed are marked with $\diamond$.
In the SPIJN algorithm the simple, separable reweighting scheme can easily be adapted, but the non-separable reweighting scheme from Equation (3.28) can not be changed to the fixed parameter setting.

4. Experiments and results

In this chapter the different methods as discussed in the previous chapter are analysed. The chapter starts with presenting results for the voxel-by-voxel methods using simulated signals with different noise levels in Section 4.1. This analysis is done for different sequence types: an MRF sequence and exponential signals.
Afterwards the results from the joint methods applied to a numerical phantom with two components are shown in Section 4.2. The most promising methods are then tested with a numerical phantom containing three components in Section 4.3. A real phantom phantom is then used to test the effects of undersampling in Section 4.4, this is followed by the results of the application of the methods to in vivo measurements in Section 4.5.2. Section 4.5.2 shows the effects of undersampling for a different in vivo measurement with the found decomposition. The chapter ends with some side-results in Section 4.6.

4.1. Simulations voxel-by-voxel methods

An image consisting of 25 voxels each containing two components with equal weights (50%-50%) is analysed using 12 different methods. Tables 4.1 and 4.2 show the results for this analysis using the MRF500 sequence and a multi-echo spin echo T_2 mapping sequence. The different subtables show the results for different noise levels. The combination of relaxation times $T_1 = 305$ ms,$T_2 = 50.3$ ms and $T_1 = 1005$ ms,$T_2 = 205$ ms are used as a test case. For different relaxation time combinations similar results were obtained.
The different tested methods are as follows:

1. the non-negative least squares algorithm (NNLS)(Section 3.2.1);
2. the support shrinkage non-negative orthogonal matching pursuit (SNNOMP) algorithm (Section 3.2.4);
3. the regularised non-negative orthogonal matching pursuit (RNNOMP) algorithm (Section 3.2.5)
4. the Bayesian method (Section 3.2.2)
5. a reweighted least squares method from equation 3.15 for comparison with the Bayesian method;
6. the ℓ_1 reweighted method with two reweighting schemes:
   a) from Equation 3.17 noted as Reweighted ℓ_1 1;
   b) from Equation 3.18 noted as Reweighted Reweighted ℓ_1 2;
7. a reweighted NNLS methods with two reweighting schemes:
   a) based on Equation 3.27 noted as reweighted NNLS 1;
   b) based on Equation 3.28 noted reweighted NNLS 2;

8. the sparsity promoting iterative joint NNLS (SPIJN) algorithm with two reweighting schemes:
   a) the simple, first scheme based on Eq. (3.27)
   b) the third, non-separable reweighting scheme from Eq. (3.28);
9. the joint regularised non-negative orthogonal matching pursuit (JRNNOMP) algorithm.

The analysed signals consist of two tissue types with different T_1 and T_2 values, which are not part of the used dictionary. To this signal Gaussian complex noise is added, where the signal to noise (SNR) ratio is related to the standard deviation of the noise according to

$$\sigma = \frac{M}{SNR}, \tag{4.1}$$

where M is the maximum absolute value of the simulated signal. When a method uses a regularisation parameter, it is set equal to the standard deviation. When there is no noise the regularisation parameter is set to 10^{-5}.

In the case that the matched decomposition consists of more than two components, the components are considered to belong to the closest simulated component. For the exponential signal decoding only T_2 this is done with $T_2 = 125$ as the boundary between the two true components. For the MRF signal a component is considered to belong to the first component with $T_1 = 305, T_2 = 50.3$ if the matched T_1 is smaller than 650 and T_2 is smaller than 125, if the matched component has relaxation times larger than 650ms and 125ms it is considered to belong to the second component. If the matched component is not part of these two ranges, it is considered as an outlier. After the grouping the weighted average is used to determine the matched relaxation times.

The dictionaries were formed as follows. The MRF dictionary contained 6280 T_1/T_2 combinations, with T_1 ranging from 10ms to 2950ms (10 to 90 with stepsize 5, to 1000 with stepsize 10, to 1500 with stepsize 20 and to 2950 with stepsize 50) and T_2 ranging from 2ms to 500 ms (2 to 10 with stepsize 2, to 150 with stepsize 5, to 200 with stepsize 10 and to 500 with stepsize 50). The T_2 relaxation dictionary contained the same 43 T_2 values, without T_1. The T_2 simulation uses 32 echoes. The relaxation times of the simulated components are not contained in this dictionary.

Table 4.1.: The solving behaviour of the different algorithms for 25 voxels containing two equally mixed components with different noise levels for the MRF500 sequence. Every entry represents the mean value and its standard deviation - when applicable - between brackets.

(a) SNR = 1000

Method	**Weights: 0.5,0.5**	T_1 **[ms]: 305, 1005**	T_2 **[ms]: 50.3, 205.0**	**Number of components**	**One comp. [%]**	**Relative error**	**Time per voxel [s]**
NNLS	0.50 (0.01) 0.50 (0.01)	303.79 (1.13) 1005.77 (5.49)	50.07 (0.76) 203.91 (0.87)	4.80 (0.98)	0	1.40e-03 (8.24e-05)	4.805e-02
SNNOMP	0.50 (0.01) 0.50 (0.01)	303.77 (1.19) 1005.73 (5.30)	50.03 (0.78) 203.89 (0.85)	4.48 (1.14)	0	1.38e-03 (4.55e-05)	2.281e-01
RNNOMP	0.50 (0.00) 0.50 (0.00)	303.77 (1.16) 1005.59 (6.15)	50.15 (0.65) 203.68 (0.86)	4.56 (0.98)	0	1.38e-03 (4.57e-05)	3.028e-01
Bayesian	0.50 (0.04) 0.50 (0.04)	303.08 (8.99) 1001.52 (55.26)	51.53 (5.63) 198.22 (3.66)	4.84 (0.37)	0	3.47e-02 (8.36e-02)	8.115e-02
Reweighted LS	0.50 (0.00) 0.50 (0.00)	300.00 (0.00) 1043.91 (32.48)	51.19 (1.07) 202.00 (4.20)	2.76 (0.76)	0	4.07e-03 (3.33e-03)	2.485e-01
Reweighted ℓ_1 1	0.49 (0.00) 0.51 (0.00)	300.00 (0.00) 1000.00 (4.90)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	1.73e-03 (1.72e-04)	2.890e-01
Reweighted ℓ_1 2	0.49 (0.00) 0.51 (0.00)	300.00 (0.00) 993.42 (4.64)	50.00 (0.00) 200.00 (0.00)	2.04 (0.20)	0	1.76e-03 (7.15e-05)	4.275e-01
Reweighted NNLS 1	0.49 (0.00) 0.51 (0.00)	301.20 (3.25) 977.20 (7.76)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	2.30e-03 (4.56e-04)	5.493e-02
Reweighted NNLS 2	0.37 (0.00) 0.63 (0.00)	278.40 (0.50) 870.56 (3.49)	40.49 (0.15) 183.70 (0.40)	2.00 (0.00)	0	3.68e-03 (9.40e-05)	1.797e-01
SPIJN 1 $\lambda = 0.10$	0.49 (0.00) 0.51 (0.00)	300.00 (0.00) 1000.00 (0.00)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	3.27e-03 (2.49e-05)	9.328e-02
SPIJN 3 $\lambda = 0.45$	0.49 (0.00) 0.51 (0.00)	300.00 (0.00) 1000.00 (0.00)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	1.75e-03 (4.67e-05)	6.936e-02
JRNNOMP	0.50 (0.00) 0.50 (0.00)	304.84 (0.63) 1009.43 (2.82)	50.18 (0.39) 203.40 (0.67)	3.00 (0.00)	0	1.39e-03 (4.59e-05)	5.722e-02

(b) SNR = 50

Method	**Weights: 0.5,0.5**	T_1 **[ms]: 305, 1005**	T_2 **[ms]: 50.3, 205.0**	**Number of components**	**One comp. [%]**	**Relative error**	**Time per voxel [s]**
NNLS	0.49 (0.06) 0.48 (0.06)	291.56 (19.05) 1047.08 (77.02)	46.88 (6.62) 190.21 (11.42)	5.76 (1.84)	0	2.73e-02 (9.88e-04)	3.822e-02
SNNOMP	0.49 (0.06) 0.48 (0.06)	292.60 (16.55) 1050.77 (77.67)	47.63 (6.65) 190.18 (11.71)	5.88 (1.70)	0	2.72e-02 (8.42e-04)	1.794e-01
RNNOMP	0.49 (0.06) 0.48 (0.06)	292.10 (19.18) 1044.93 (76.92)	47.20 (6.92) 190.76 (10.77)	5.24 (1.30)	0	2.72e-02 (8.43e-04)	2.798e-01
Bayesian	0.50 (0.17) 0.50 (0.17)	291.52 (28.33) 954.47 (118.14)	49.66 (10.11) 184.22 (14.31)	5.00 (0.00)	16	1.59e-01 (4.22e-01)	9.436e-02
Reweighted LS	0.50 (0.07) 0.45 (0.06)	299.69 (32.11) 1193.24 (178.16)	51.73 (7.29) 210.99 (30.78)	3.56 (0.98)	0	4.23e-01 (1.72e+00)	2.919e-01
Reweighted ℓ_1 1	0.45 (0.04) 0.55 (0.04)	269.20 (16.95) 995.20 (69.40)	45.20 (4.58) 194.80 (9.00)	2.00 (0.00)	0	2.82e-02 (1.24e-03)	2.879e-01
Reweighted ℓ_1 2	0.44 (0.04) 0.56 (0.04)	275.60 (14.99) 950.80 (73.26)	46.40 (4.36) 186.00 (10.20)	2.00 (0.00)	0	2.76e-02 (8.70e-04)	3.382e-01
Reweighted NNLS 1	0.48 (0.06) 0.50 (0.07)	290.90 (17.14) 1035.94 (80.29)	48.36 (6.79) 190.84 (11.96)	2.56 (0.75)	0	2.73e-02 (8.36e-04)	6.430e-02
Reweighted NNLS 2	0.39 (0.06) 0.61 (0.06)	281.01 (18.24) 881.95 (57.87)	41.91 (5.59) 186.24 (9.71)	2.08 (0.27)	0	2.75e-02 (8.58e-04)	1.678e-01
SPIJN 1 $\lambda = 0.15$	0.52 (0.00) 0.48 (0.00)	300.00 (0.00) 1100.00 (0.00)	55.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	2.82e-02 (7.67e-04)	3.713e-02
SPIJN 3 $\lambda = 1.01$	0.52 (0.00) 0.48 (0.00)	300.00 (0.00) 1100.00 (0.00)	55.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	2.76e-02 (7.85e-04)	5.095e-02
JRNNOMP	0.53 (0.07) 0.45 (0.06)	311.48 (35.21) 1068.35 (37.23)	55.17 (9.26) 191.21 (9.82)	5.36 (1.02)	0	2.73e-02 (8.37e-04)	3.083e-02

(c) SNR = 10

Method	**Weights: 0.5,0.5**	T_1 **[ms]: 305, 1005**	T_2 **[ms]: 50.3, 205.0**	**Number of components**	**One comp. [%]**	**Relative error**	**Time per voxel [s]**
NNLS	0.50 (0.15) 0.38 (0.16)	290.82 (74.91) 1137.33 (470.85)	36.87 (22.91) 180.63 (28.24)	5.64 (1.20)	0	1.39e-01 (5.07e-03)	3.472e-02
SNNOMP	0.50 (0.15) 0.37 (0.16)	290.39 (76.50) 1137.33 (470.85)	36.85 (22.95) 180.63 (28.24)	5.60 (1.17)	0	1.39e-01 (5.07e-03)	1.415e-01
RNNOMP	0.49 (0.14) 0.39 (0.16)	290.86 (76.02) 1142.02 (467.45)	38.25 (22.93) 180.50 (28.17)	5.12 (1.14)	0	1.39e-01 (5.07e-03)	1.985e-01
Bayesian	0.72 (0.31) 0.09 (0.20)	395.15 (126.14) 938.19 (192.27)	84.68 (24.21) 192.45 (43.46)	4.94 (0.24)	32	1.68e-01 (3.44e-02)	7.611e-02
Reweighted LS	0.28 (0.17) 0.15 (0.12)	273.89 (115.13) 2166.97 (685.92)	26.58 (17.38) 266.59 (87.67)	7.21 (1.97)	44	5.01e+00 (5.39e+00)	1.008e+00
Reweighted ℓ_1 1	0.76 (0.27) 0.22 (0.27)	252.38 (59.67) 978.89 (423.88)	34.52 (24.51) 215.56 (61.30)	1.48 (0.50)	16	2.09e-01 (7.07e-02)	2.866e-01
Reweighted ℓ_1 2	0.45 (0.28) 0.41 (0.30)	318.60 (120.83) 858.24 (264.60)	47.77 (37.83) 178.53 (58.05)	1.88 (0.52)	0	1.41e-01 (5.09e-03)	3.235e-01
Reweighted NNLS 1	0.50 (0.16) 0.38 (0.16)	292.44 (78.19) 1129.34 (474.01)	37.62 (23.61) 180.21 (29.01)	2.96 (0.72)	0	1.39e-01 (5.07e-03)	6.017e-02
Reweighted NNLS 2	0.54 (0.19) 0.33 (0.23)	310.77 (81.68) 1541.31 (799.67)	40.98 (27.05) 240.44 (90.71)	2.52 (0.64)	0	1.39e-01 (5.06e-03)	2.365e-01
SPIJN 1 $\lambda = 0.15$	0.50 (0.01) 0.50 (0.01)	310.00 (0.00) 1000.00 (0.00)	60.00 (0.00) 170.00 (0.00)	2.00 (0.00)	0	1.40e-01 (5.36e-03)	3.084e-02
SPIJN 3 $\lambda = 1.01$	0.49 (0.01) 0.51 (0.01)	310.00 (0.00) 1000.00 (0.00)	60.00 (0.00) 170.00 (0.00)	2.00 (0.00)	0	1.40e-01 (5.36e-03)	4.884e-02
JRNNOMP	0.67 (0.14) 0.26 (0.10)	350.84 (53.60) 1293.41 (396.74)	60.48 (22.37) 204.63 (28.43)	5.00 (1.02)	0	1.39e-01 (5.08e-03)	1.887e-02

(d) SNR = 5

Method	**Weights: 0.5,0.5**	T_1 **[ms]: 305, 1005**	T_2 **[ms]: 50.3, 205.0**	**Number of components**	**One comp. [%]**	**Relative error**	**Time per voxel [s]**
NNLS	0.45 (0.19) 0.26 (0.21)	202.17 (131.50) 1442.20 (768.97)	22.24 (24.60) 229.96 (108.96)	4.80 (1.10)	0	2.94e-01 (2.45e-02)	3.249e-02
SNNOMP	0.46 (0.20) 0.26 (0.21)	202.15 (131.53) 1442.20 (768.97)	22.17 (24.66) 229.96 (108.96)	4.80 (1.10)	0	2.94e-01 (2.45e-02)	1.107e-01
RNNOMP	0.43 (0.18) 0.31 (0.23)	198.51 (127.91) 1420.68 (757.19)	23.26 (24.10) 225.45 (108.81)	4.64 (1.29)	0	2.94e-01 (2.45e-02)	1.636e-01
Bayesian	0.60 (0.31) 0.19 (0.26)	363.17 (203.25) 901.02 (204.85)	59.15 (50.89) 171.86 (47.49)	5.00 (0.00)	56	3.06e-01 (2.30e-02)	7.436e-02
Reweighted LS	0.48 (0.22) 0.11 (0.10)	236.58 (61.22) 2187.65 (208.71)	16.23 (4.97) 300.63 (22.13)	9.00 (0.71)	84	1.29e+01 (2.60e+00)	2.742e+00
Reweighted ℓ_1 1	0.57 (0.26) 0.32 (0.29)	188.80 (163.13) 1250.00 (667.53)	29.76 (39.95) 268.12 (124.36)	1.84 (0.37)	0	3.11e-01 (4.11e-02)	4.050e-01
Reweighted ℓ_1 2	0.48 (0.27) 0.35 (0.31)	258.14 (171.14) 1144.12 (612.32)	36.40 (43.07) 233.53 (109.11)	2.36 (0.74)	0	2.95e-01 (2.45e-02)	4.722e-01
Reweighted NNLS 1	0.44 (0.20) 0.29 (0.24)	199.52 (131.47) 1435.74 (771.34)	21.90 (24.92) 229.91 (109.24)	2.96 (0.72)	0	2.94e-01 (2.45e-02)	6.567e-02
Reweighted NNLS 2	0.49 (0.21) 0.32 (0.26)	228.03 (141.99) 1439.82 (828.81)	22.11 (25.74) 240.86 (109.80)	2.64 (0.62)	0	2.94e-01 (2.44e-02)	2.392e-01
SPIJN 1 $\lambda = 0.15$	0.54 (0.05) 0.46 (0.05)	370.00 (0.00) 940.00 (0.00)	75.00 (0.00) 180.00 (0.00)	2.00 (0.00)	0	2.96e-01 (2.45e-02)	3.163e-02
SPIJN 3 $\lambda = 1.01$	0.54 (0.05) 0.46 (0.05)	370.00 (0.00) 940.00 (0.00)	75.00 (0.00) 180.00 (0.00)	2.00 (0.00)	0	2.96e-01 (2.45e-02)	4.761e-02
JRNNOMP	0.66 (0.13) 0.20 (0.08)	229.12 (124.69) 1494.88 (665.22)	36.12 (28.74) 210.86 (106.46)	4.20 (1.06)	0	2.94e-01 (2.44e-02)	1.511e-02

Table 4.2.: The solving behaviour of the different algorithms for 25 voxels containing two equally mixed components with different noise levels for the T_2 sequence. The mean values and between brackets the standard deviation are shown when applicable.

(a) SNR = 1000

Method	**Weights: 0.5,0.5**	T_2 **[ms]: 50.3, 205.0**	**Number of components**	**One comp. [%]**	**Relative error**	**Time per voxel [s]**
NNLS	0.54 (0.06) 0.46 (0.06)	44.28 (8.17) 205.77 (2.33)	3.52 (1.30)	0	1.61e-03 (2.23e-04)	2.146e-03
SNNOMP	0.54 (0.06) 0.46 (0.06)	44.28 (8.17) 205.77 (2.33)	3.52 (1.30)	0	1.61e-03 (2.23e-04)	3.460e-02
RNNOMP	0.54 (0.06) 0.46 (0.06)	44.28 (8.17) 205.77 (2.33)	3.52 (1.30)	0	1.61e-03 (2.23e-04)	2.034e-02
Bayesian	0.30 (0.07) 0.70 (0.07)	37.27 (2.49) 168.15 (22.16)	4.00 (0.00)	56	1.15e-01 (1.02e-01)	1.687e-02
Reweighted LS	0.50 (0.00) 0.50 (0.00)	49.83 (0.43) 203.73 (1.24)	2.16 (0.37)	0	1.95e-03 (1.22e-03)	1.617e-01
Reweighted ℓ_1 1	0.48 (0.00) 0.52 (0.00)	47.60 (2.50) 200.00 (0.00)	2.00 (0.00)	0	4.89e-03 (2.14e-03)	9.309e-02
Reweighted ℓ_1 2	0.49 (0.00) 0.51 (0.00)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	2.97e-03 (2.99e-04)	4.341e-01
Reweighted NNLS 1	0.49 (0.00) 0.51 (0.00)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	2.95e-03 (3.00e-04)	3.917e-01
Reweighted NNLS 2	0.49 (0.00) 0.51 (0.00)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	3.00e-03 (2.95e-04)	1.582e-02
SPIJN 1 $\lambda = 0.45$	0.41 (0.00) 0.59 (0.00)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	7.12e-02 (2.16e-05)	6.680e-01
SPIJN 3 $\lambda = 0.45$	0.48 (0.00) 0.52 (0.00)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	4.16e-03 (2.17e-04)	8.737e-04
JRNNOMP	0.54 (0.06) 0.46 (0.06)	43.56 (8.27) 205.18 (1.49)	3.56 (1.27)	0	1.61e-03 (2.23e-04)	1.943e-02

(b) SNR = 100

Method	**Weights: 0.5,0.5**	T_2 **[ms]: 50.3, 205.0**	**Number of components**	**One component [%]**	**Relative error**	**Time per voxel [s]**
NNLS	0.65 (0.18) 0.35 (0.18)	36.05 (18.61) 222.60 (48.55)	3.24 (0.91)	0	1.60e-02 (2.24e-03)	2.006e-04
SNNOMP	0.65 (0.18) 0.35 (0.18)	36.05 (18.61) 222.60 (48.55)	3.24 (0.91)	0	1.60e-02 (2.24e-03)	1.159e-02
RNNOMP	0.65 (0.18) 0.35 (0.18)	36.05 (18.61) 222.60 (48.55)	3.24 (0.91)	0	1.60e-02 (2.24e-03)	7.420e-03
Bayesian	0.87 (0.00) 0.13 (0.00)	103.88 (0.07) 450.00 (0.00)	4.00 (0.00)	16	4.68e-02 (3.99e-03)	5.495e-03
Reweighted LS	0.88 (0.16) 0.12 (0.16)	12.20 (13.64) 244.19 (78.36)	2.35 (1.46)	8	1.22e+00 (1.34e+00)	9.638e-02
Reweighted ℓ_1 1	0.59 (0.07) 0.41 (0.07)	49.00 (6.63) 262.00 (53.44)	2.00 (0.00)	0	3.11e-02 (7.31e-03)	2.910e-02
Reweighted ℓ_1 2	0.47 (0.08) 0.53 (0.08)	49.53 (7.57) 201.77 (39.32)	2.28 (0.45)	0	1.84e-02 (2.77e-03)	3.453e-02
Reweighted NNLS 1	0.64 (0.18) 0.36 (0.18)	37.86 (18.72) 220.82 (41.36)	2.56 (0.64)	0	1.62e-02 (2.28e-03)	2.948e-03
Reweighted NNLS 2	0.50 (0.04) 0.50 (0.04)	51.40 (5.39) 203.60 (17.86)	2.00 (0.00)	0	1.76e-02 (2.69e-03)	6.379e-03
SPIJN 1 $\lambda = 0.45$	0.41 (0.01) 0.59 (0.01)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	7.32e-02 (5.87e-04)	6.818e-04
SPIJN 3 $\lambda = 0.45$	0.48 (0.01) 0.52 (0.01)	50.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	1.74e-02 (2.31e-03)	6.618e-04
JRNNOMP	0.65 (0.18) 0.35 (0.18)	35.79 (19.26) 214.72 (16.22)	3.32 (0.84)	0	1.61e-02 (2.23e-03)	8.342e-03

(c) SNR = 50

Method	**Weights: 0.5,0.5**	**T_2 [ms]: 50.3, 205.0**	**Number of components**	**One component [%]**	**Relative error**	**Time per voxel [s]**
NNLS	0.69 (0.20) 0.31 (0.20)	34.89 (21.65) 241.82 (59.67)	2.92 (1.23)	0	3.21e-02 (4.46e-03)	2.406e-04
SNNOMP	0.69 (0.20) 0.31 (0.20)	34.89 (21.65) 241.82 (59.67)	2.92 (1.23)	0	3.21e-02 (4.46e-03)	1.009e-02
RNNOMP	0.69 (0.20) 0.31 (0.20)	34.89 (21.65) 241.82 (59.67)	2.92 (1.23)	0	3.21e-02 (4.46e-03)	7.179e-03
Bayesian	0.41 (0.09) 0.59 (0.09)	101.21 (1.21) 141.25 (1.28)	4.00 (0.00)	92	7.90e-02 (9.83e-04)	5.314e-03
Reweighted LS	0.96 (0.07) 0.04 (0.07)	6.88 (5.38) 291.89 (107.61)	2.54 (1.34)	48	4.67e+00 (3.89e+00)	2.339e-01
Reweighted ℓ_1 1	0.61 (0.09) 0.39 (0.09)	51.20 (10.70) 285.60 (74.68)	2.00 (0.00)	0	4.22e-02 (6.14e-03)	2.932e-02
Reweighted ℓ_1 2	0.50 (0.15) 0.50 (0.15)	53.66 (13.05) 227.88 (75.32)	2.12 (0.32)	0	3.55e-02 (5.04e-03)	3.632e-02
Reweighted NNLS 1	0.68 (0.21) 0.32 (0.21)	37.04 (22.67) 240.94 (60.10)	2.20 (0.57)	0	3.22e-02 (4.46e-03)	2.286e-03
Reweighted NNLS 2	0.54 (0.10) 0.46 (0.10)	56.00 (11.66) 224.80 (49.49)	2.00 (0.00)	0	3.37e-02 (4.08e-03)	7.680e-03
SPIJN 1 $\lambda = 0.50$	0.45 (0.01) 0.55 (0.01)	70.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	9.42e-02 (3.27e-03)	1.181e-03
SPIJN 3 $\lambda = 0.80$	0.52 (0.01) 0.48 (0.01)	70.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	4.53e-02 (6.41e-03)	3.389e-03
JRNNOMP	0.70 (0.20) 0.30 (0.20)	35.10 (22.30) 244.92 (61.86)	2.92 (1.23)	0	3.21e-02 (4.45e-03)	9.224e-03

(d) SNR = 25

Method	**Weights: 0.5,0.5**	T_2 **[ms]: 50.3, 205.0**	**Number of components**	**One component [%]**	**Relative error**	**Time per voxel [s]**
NNLS	0.73 (0.21) 0.27 (0.21)	37.58 (27.49) 295.16 (107.49)	2.68 (1.38)	0	6.41e-02 (8.81e-03)	1.811e-04
SNNOMP	0.73 (0.21) 0.27 (0.21)	37.58 (27.49) 295.16 (107.49)	2.68 (1.38)	0	6.41e-02 (8.81e-03)	9.305e-03
RNNOMP	0.73 (0.21) 0.27 (0.21)	37.58 (27.49) 295.16 (107.49)	2.68 (1.38)	0	6.41e-02 (8.81e-03)	6.658e-03
Bayesian	0.87 (0.19) 0.13 (0.19)	106.38 (2.01) 143.76 (1.24)	4.00 (0.00)	76	1.15e-01 (6.54e-03)	5.317e-03
Reweighted LS	Only one components was matched in this case					
Reweighted ℓ_1 1	0.65 (0.13) 0.35 (0.13)	53.00 (18.75) 312.08 (100.58)	1.96 (0.20)	4	7.15e-02 (1.24e-02)	3.040e-02
Reweighted ℓ_1 2	0.54 (0.25) 0.46 (0.25)	53.59 (23.31) 245.20 (96.59)	1.88 (0.32)	0	6.84e-02 (9.67e-03)	3.349e-02
Reweighted NNLS 1	0.72 (0.22) 0.28 (0.22)	37.73 (27.32) 295.73 (109.41)	2.00 (0.69)	0	6.42e-02 (8.81e-03)	2.025e-03
Reweighted NNLS 2	0.61 (0.15) 0.39 (0.15)	59.72 (19.60) 278.00 (105.49)	2.08 (0.27)	0	6.55e-02 (8.82e-03)	1.019e-02
SPIJN 1 $\lambda = 0.40$	0.58 (0.03) 0.42 (0.03)	85.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	9.42e-02 (1.01e-02)	7.018e-04
SPIJN 3 $\lambda = 0.80$	0.59 (0.03) 0.41 (0.03)	85.00 (0.00) 200.00 (0.00)	2.00 (0.00)	0	8.06e-02 (1.17e-02)	5.855e-04
JRNNOMP	0.72 (0.23) 0.28 (0.23)	37.91 (27.92) 290.74 (105.61)	2.68 (1.26)	0	6.43e-02 (8.82e-03)	4.793e-03

In general, the Bayesian algorithm and the reweighted least squares show the largest errors and return most often only one component, both of these methods do not include the non-negativity constraint. Another general behaviour is that the reweighting used for the NNLS does not improve the multi-component matchings compared to the unweighted NNLS.
Most methods give accurate and precise results for SNR=1000. The Bayesian method and reweighted LS show relatively high standard deviations in the relaxation times. For the T_2 sequence the matched relaxation times show the largest error for the Bayesian method. In general the variation is larger for the smaller component than for the second component at SNR=1000.
With decreasing SNR the voxel-by-voxel methods including non-negativity show similar behaviour. When the MRF sequence is used with a low SNR the SPIJN algorithm gives the best results. For the T_2 sequence the SPIJN algorithm gives the best results as well, although the relaxation times of the short component is for SNR=50 and 25 overestimated. The longer component is estimated correctly.
It can be seen here that the T_2 measurement is more sensitive to noise than the MRF500 sequence. This can partly be larger number of echoes used for the MRF sequence (500 echoes) compared to the T_2 measurement (32 echoes).

4.2. Simulations joint methods

In order to evaluate the effectiveness of the joint methods it is necessary to consider a group of pixels. In this section a set of 17×7 pixels is considered, which all contain one or two components. The eleven columns in the centre contain different combinations of the two components. The three outer columns on both sides contain only one component. The distribution of the two components is shown in Figure 4.1. The same dictionaries as in the previous section were used.

For each of the pixels the corresponding mixture of signals is calculated and a certain level of noise is added, just as in the previous section. In Figure 4.2 the result of a single component match is shown with an SNR of 5 using sequence MRF500. It can be seen that the matched intensity is approximately 1 throughout the matching. In the centre mixed area the matching results in an averaged value of the true relaxation times.
In Figures 4.3, 4.4 and 4.5 the results for SNR values of 100,10 and 3 respectively are shown using the SPIJN method with the reweighting scheme from Equation 3.27 and the JRNNOMP algorithm using two segments using the MRF500 sequence. The figures show the absolute error in the matched weights for the two simulated components and also shown the matched relaxation times.
The SPIJN algorithm finds the two closest components with each noise level. The SPIJN algorithm has also been tried with the reweighting scheme of Equation 3.28 with $\lambda = 2$, which leads to similar results.
The JRNNOMP algorithm has also been tried without the segmentation, the offset in the matched relaxation times as well as the matched weights is larger for the method without segmentation. When using the JRNNOMP algorithm the maximum number of components was set to 4.
In Section A.1 of the Appendix the results of applying the joint algorithms for T_2 relaxation signals on the same true distributions as for the MRF500 signal are shown. The differences between the two methods are smaller than for the MRF500 sequence. In general the SPIJN algorithm gives a better approximation of the relaxation times.
Something which can not be seen from the shown images, but was observed in the simulations, is that the JRNNOMP selects smaller T_2 values for both components for the

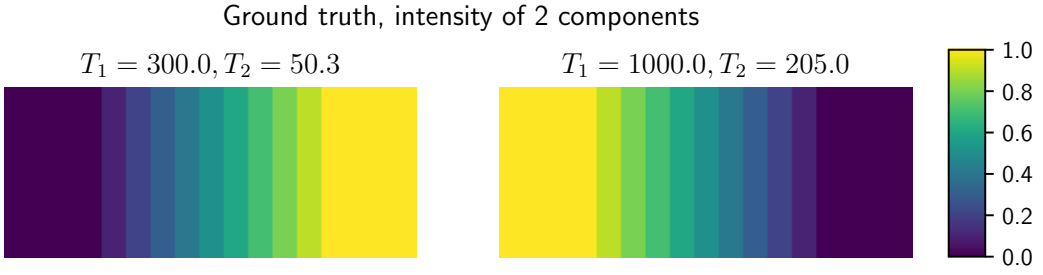
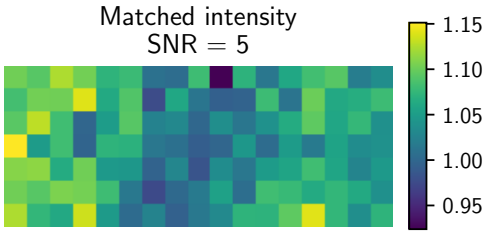
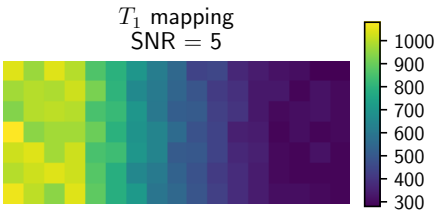


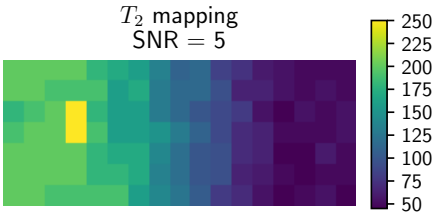
Figure 4.1.: The distribution of the two components in the simulated image.



(a) The matched scales



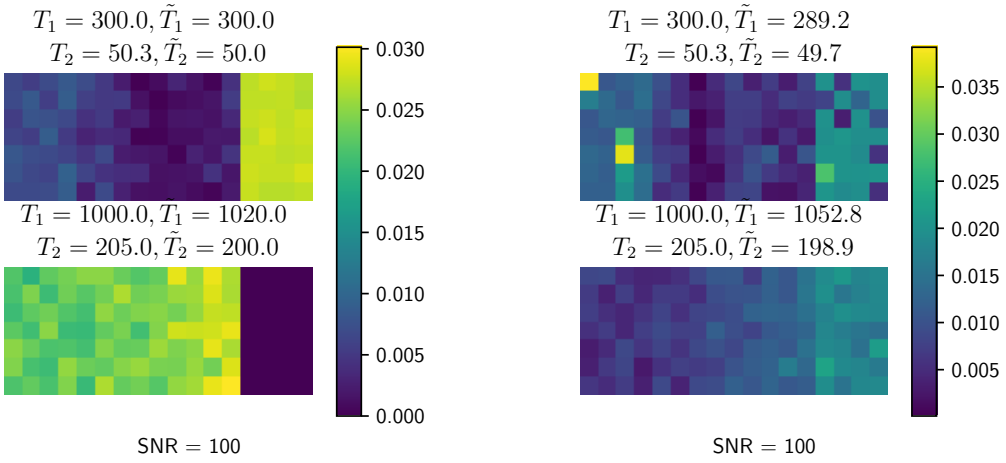
(b) The matched T_1 values



(c) The matched T_2 values.

Figure 4.2.: The result of a single component MRF500 matching with a signal to noise ratio of 5.

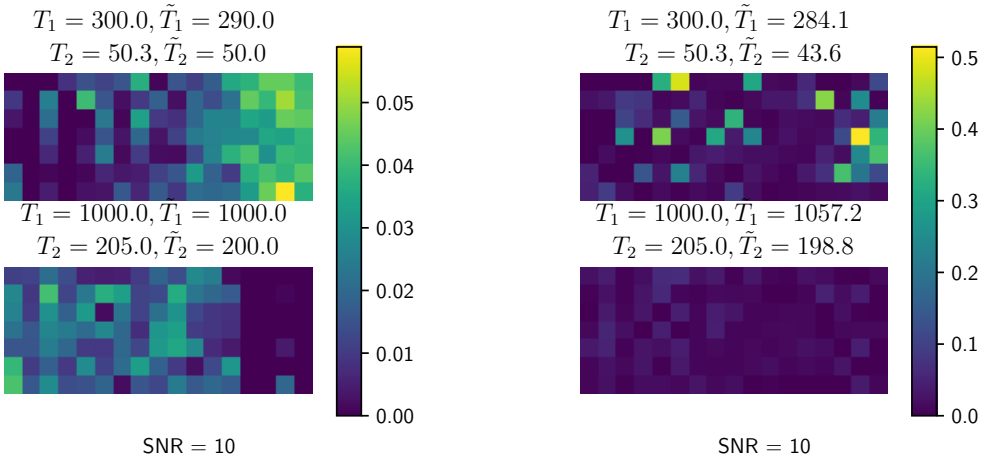
signals with more noise. The SPIJN algorithm does not show this behaviour. Following on this the shorter T_2 values are given a relatively high weight, leading to a sum of weights above 1.



(a) Using the SPIJN algorithm the reweighting scheme from Equation 3.27 with $\lambda = 0.25$.

(b) Using the JRNNOMP algorithm with two segments.

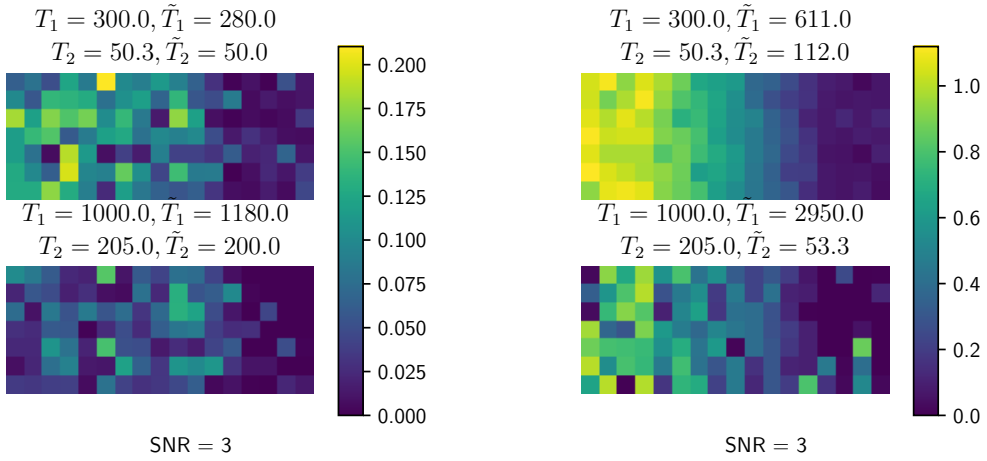
Figure 4.3.: The multi-component matching for the simulated image using the MRF500 sequence with SNR=100. The true T_1 and T_2 values and matched T_1 and T_2 values ($\tilde{T}_1$, $\tilde{T}_2$) are given. In the images the absolute difference of the matched weights compared to the true weights as shown in Figure 4.1 are visualised. Notice that the two colourbars have different scales.



(a) Using the SPIJN algorithm the reweighting scheme from Equation 3.27 with $\lambda = 0.25$.

(b) Using the JRNNOMP algorithm with two segments.

Figure 4.4.: The multi-component matching for the simulated image using the MRF500 sequence with SNR=10. The true T_1 and T_2 values and matched T_1 and T_2 values ($\tilde{T}_1$, $\tilde{T}_2$) are given. In the images the absolute difference of the matched weights compared to the true weights as shown in Figure 4.1 are visualised. Notice that the two colourbars have different scales.



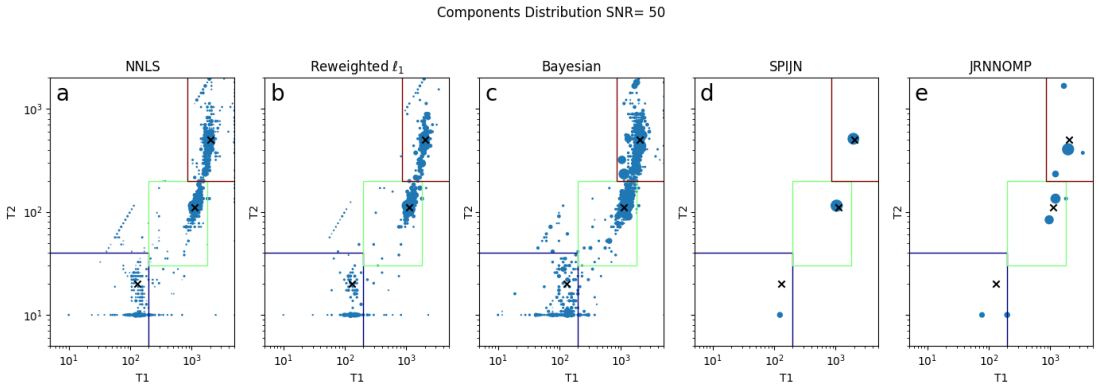
(a) Using the SPIJN algorithm the reweighting scheme from Equation 3.27 with $\lambda = 0.25$.

(b) Using the JRNNOMP algorithm with two segments.

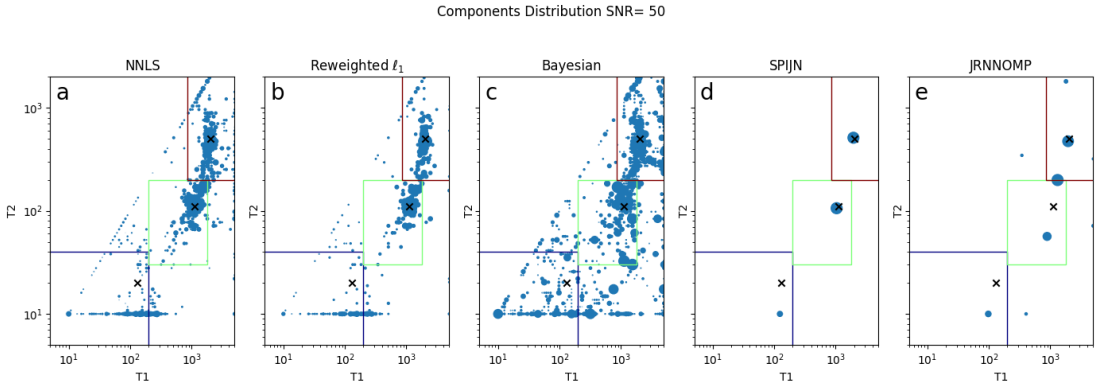
Figure 4.5.: The multi-component matching for the simulated image using the MRF500 sequence with SNR=3. The true T_1 and T_2 values and matched T_1 and T_2 values ($\tilde{T}_1$, $\tilde{T}_2$) are given. In the images the absolute difference of the matched weights compared to the true weights as shown in Figure 4.1 are visualised. The two plots use different colour scales.

4.3. Numerical simulations with three components

In order to analyse the effectiveness of the joint sparsity constraint, the NNLS algorithm, reweighted ℓ_1 algorithm and Bayesian approach are compared to the SPIJN and JRNNOMP algorithms. These numerical simulations now including a myelin water like component. For these simulations the MRF500, MRF200 and T2 exponential sequences were used. In both the reweighed ℓ_1 and the SPIJN algorithm the simple reweighting was used. The numerical simulations assume three T_1/T_2 combinations to be present; a short component T_1=110ms, T_2=20ms present in every signal with 10 percent related to myelin water, a mid-range component T_1=1100ms, T_2=130ms related to intra and extracellular water and a long component T_1=2000ms, T_2=500ms related to free water. The last two components vary from 0% to 90% in 10 steps. To the simulated MRF signals Gaussian noise with SNR=50 was added, for the T_2 exponential signals Gaussian noise with SNR=100 was added.
A logarithmically spaced dictionary was used in this section. T_1 and T_2 ranged both from 10 ms to 5 seconds in 80 logarithmic steps, with the restriction $T_2 \leq T_1$.
Matched components are assumed to belong to the first component when $T_1 < 200$ms and $T_2 < 30$ms, to the second component when 200ms$< T_1 < 1800$ms, 30ms$< T_2 < 200$ms and to the third component when $T_1 > 850$ms and $T_2 > 200$ms. The matched components and the grouping for the different algorithms are visualised in Figure 4.6 for the MRF sequences. In Figure 4.7, Figure 4.8 and Figure 4.9 the matched weights are shown for the different methods and the three sequence types.
From these distributions it can be seen that the distributions from the MRF500 sequence are more concentrated around the true components than the MRF200 distributions. The Bayesian approach shows a broader spread in the matched components and this results in a larger error in the grouped, matched weights. The NNLS and reweighted ℓ_1 algorithms give very similar results. For each of the methods the short component is most difficult to estimate.
The results of the algorithms including the joint sparsity constraints show less variation in the matched components. The SPIJN algorithm gives a better result than the JRNNOMP algorithm, especially for the short component.
The approximation for the T_2 signal show an increased sensitivity to noise, as observed in the single voxel simulations as well. The Bayesian method does not give a meaningful MW-map any more, the other methods give reasonable results in which the SPIJN algorithm is closest to the true solution.



(a) MRF500



(b) MRF200

Figure 4.6.: The distribution of the matched components for the numerical phantom with the different algorithms for the two MRF sequences. The boxes indicate how the components are grouped. The blue box is the short component, the green box the middle component and the red box the long component. The size of the dots corresponds to the relative abundance the different components in the multi-component matching. The three crosses give the locations of the true components.

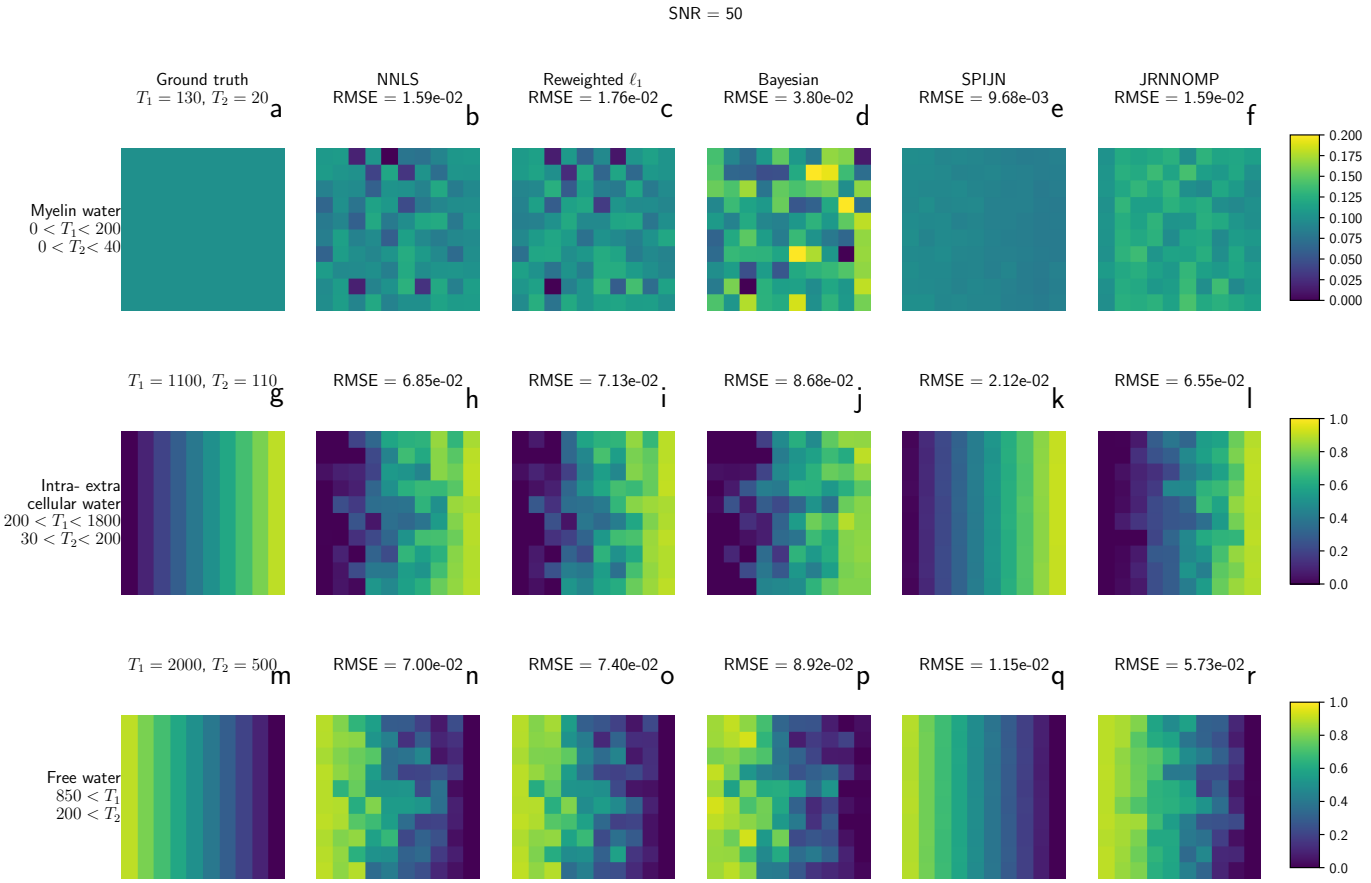


Figure 4.7.: A numerical phantom containing three different components was simulated with an SNR of 50 for sequence MRF500. The numerical phantom consists of 100 pixels, the first component is present in each pixel with ten percent, the other two components are range in the horizontal direction from 0 to 90% in 10 steps. The first column shows the ground truth for the distribution of the weights for the different components and the other columns show the retrieved matchings with the different algorithms. The first row has a different colour scale than the other rows.

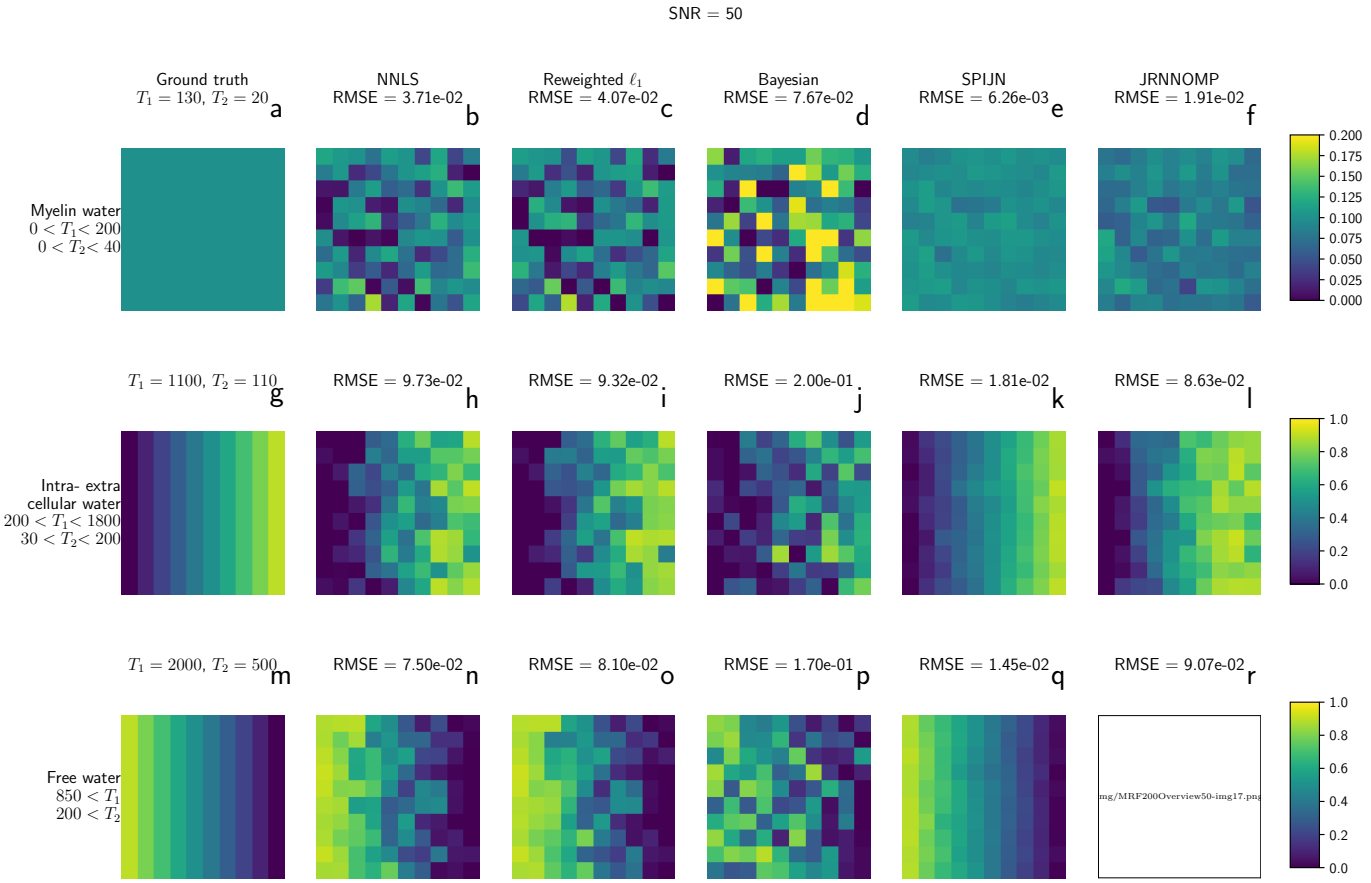


Figure 4.8.: A numerical phantom containing three different components was simulated with an SNR of 50 for sequence MRF200. The numerical phantom consists of 100 pixels, the first component is present in each pixel with ten percent, the other two components are range in the horizontal direction from 0 to 90% in 10 steps. The first column shows the ground truth for the distribution of the weights for the different components and the other columns show the retrieved matchings with the different algorithms. The first row has a different colour scale than the other rows.

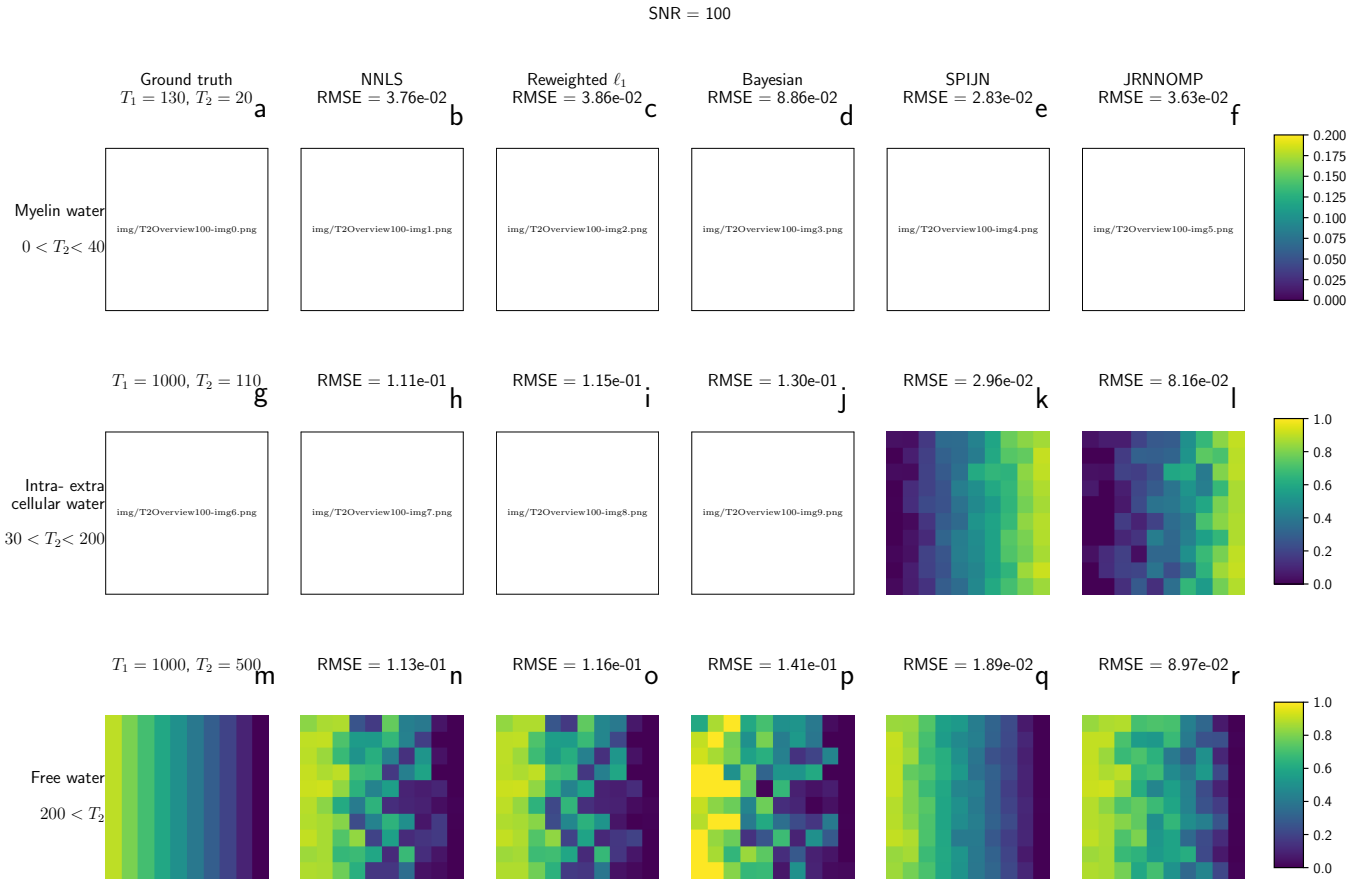


Figure 4.9.: A numerical phantom containing three different components was simulated with an SNR of 100 for the T_2 exponential signal. The numerical phantom consists of 100 pixels, the first component is present in each pixel with ten percent, the other two components are range in the horizontal direction from 0 to 90% in 10 steps. The first column shows the ground truth for the distribution of the weights for the different components and the other columns show the retrieved matchings with the different algorithms. The first row has a different colour scale than the other rows.

4.4. Phantom measurements

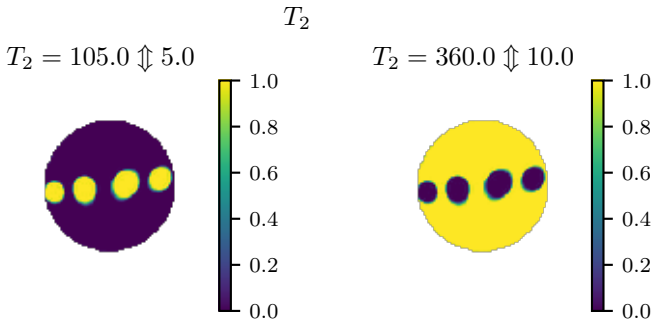
To analyse the proposed methods a phantom has been used. The phantom was made of a vessel containing a solution of $CuSo_4$ in which a plastic glove filled with water with a different concentration of $CuSo_4$ was placed. The glove contained a higher concentration of $CuSo_4$ leading to shorter relaxation times [90] and was used as thin barrier between the two compartments allowing partial volume effects.
Different measurements were performed on this phantom:

1. T_2 relaxation with and echo time TE=10ms, repetition time TR=2.5s and echo train length of 60 echoes;

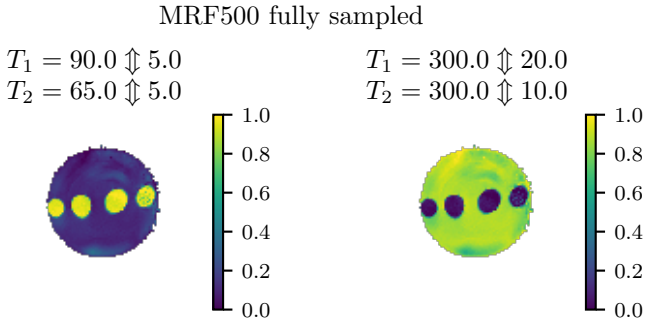
2. The MRF500 sequence, fully sampled and undersampled (1 out of 34 spiral interleaves).

The measurement were performed on a 1.5T MR-Scanner with the following parameters: FOV $= 240 \times 240 \times 5\text{mm}^3$, resolution $= 1.875 \times 1.875 \times 5\text{mm}^3$.
Measuring fully- and undersampled data makes it possible to compare the effects of undersampling on the quality of the matching. Undersampling leads to a large decrease in measurement time and is therefore necessary when an in vivo measurement is performed. It would therefore be good to know how this affects the results.
These MRF measurements lead to a simultaneous multi-parametric matching, measurement 1 measures T_2 in a more classical way. This gives a reference measurement and at the same time it is interesting to know what the performance is of the SPIJN method. A single component matching was used to form a B_1 map, the B_1 values ranged from 0.75 to 1.25, T_1 values ranged from 5ms to 1s with step size 5 up to 150, 10 up to 300 and 30 up to 1000. The T_2 values ranged from 4ms to 500ms with step size 2 up to 10, 5 up to 150 and 10 up to 500, including the restriction $T_2 \leq T_1$.
In Figure 4.10 the components matched with the SPIJN algorithm are shown for the three different measurements. The matched relaxation times are given above the corresponding image. The $\Updownarrow$ notation gives the distance from the matched relaxation time to the closest relaxation times and thus indicating the step size in this part of the parameter space. When the highest or lowest relaxation time is used, $\Downarrow$ and $\Uparrow$ are used respectively.
As assumed, the T_2 SPIJN matching results in exactly two components. Between the two components there is a smooth transition, which can be seen in the green transition zone between the fingers of the glove and the surrounding fluid.
The fully sampled MRF data also results in two components, with lower T_2 relaxation times. The fully sampled MRF data results in mixtures of the two components in the surrounding fluid, which does not correspond to the phantom. A fraction of 0.1 to 0.2 of the short component is matched in the surrounding fluid. This might also explain why the matched relaxation times are lower than in the T_2 measurement, combined with the experience that relaxation times measured with MRF are typically lower than relaxation times measured with multi-echo spin-echo measurements.
The undersampled data results in a noisier matching than the fully-sampled measurement and the relaxation times are slightly different as well. The third component could not

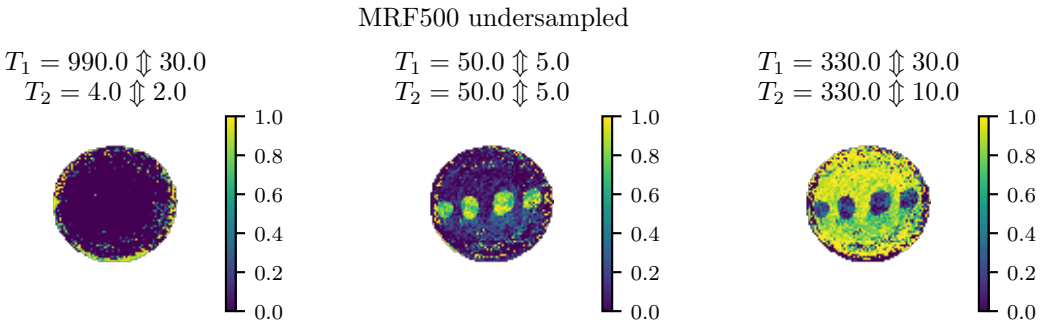
be suppressed by increasing the regularisation without changing the other components. This resulted in combining the first and component component.



(a) The components as determined with the SPIJN algorithm for the T_2 phantom measurement.



(b) The components as determined with the SPIJN algorithm for the fully sampled MRF500 phantom measurement.



(c) The components as determined with the SPIJN algorithm for the undersampled MRF500 phantom measurement.

Figure 4.10.: The multi-component matching for the glove phantom measurement with three different measurement types. The multi-component matching has been performed with the SPIJN algorithm.

4.5. In vivo measurements

In this section the results of the here proposed algorithm are shown for different available datasets, compared to published methods. For multi-echo spin-echo T_2 measurements the SPIJN algorithm is compared to results of the NNLS algorithm and the algorithm proposed by Prasloski et al. [10]. For the MRF measurement the NNLS, Bayesian and reweighted-ℓ_1 algorithms are compared to the SPIJN, JRNNOMP and search around algorithms as was done in Section 4.3.

4.5.1. In vivo T_2 measurements

Measurements including B_1 compensation for a healthy brain

In this section the proposed SPIJN algorithm is used for multi-echo spin-echo T_2 measurements to produce a myelin water fraction map. The results shown here are for 3 out of 9 slices of one measurement of a healthy volunteer. Different healthy datasets are examined as well and they produce similar results, with similar decompositions.
To obtain a MWF map, an estimation of the flip angle inhomogeneities (FAI) caused by B_1 inhomogeneities is needed. To obtain an estimation of the FAI map, a single component matching is performed as explained in Section 2.5.1. This FAI map is used as fixed parameter map for the NNLS algorithm and the SPIJN algorithm. These results are compared to the the B_1-compensated regularised NNLS method, regNNLS for short, as proposed in [10]. The dataset was acquired on a 3T MR scanner (Ingenia 3.0T, Philips, Best, The Netherlands) with the following parameters: 48 echoes, ΔTE=8 ms, TR=1200 ms, FOV=$240 \times 205 \times 72$ mm^3, resolution=$1.25 \times 1.25 \times 8$ mm^3 and 9 slices.
Figures 4.11 and 4.12 show the estimated flip angle inhomogeneity (FAI) maps and MWF maps for the 3T data for three slices. NNLS took 10 seconds per slice, regNNLS 22 minutes, and SPIJN 45 seconds on a simple laptop (IntelCore i5-6300U CPU @2.40GHz 2 cores, 4 threads). The used T_2 threshold for myelin water is $T_2 = 40$ms. The NNLS and SPIJN algorithm use a pre-computed dictionary, this calculation took a few minutes and can be reused for other measurements. The used dictionary contained B_1 values ranging from 0.6 to 1 in 50 steps and T_2 ranging from 10ms to 5s in 141 on a logarithmic scale. The regNNLS uses a T_2 values ranging from 10ms to 2s in 101 steps.

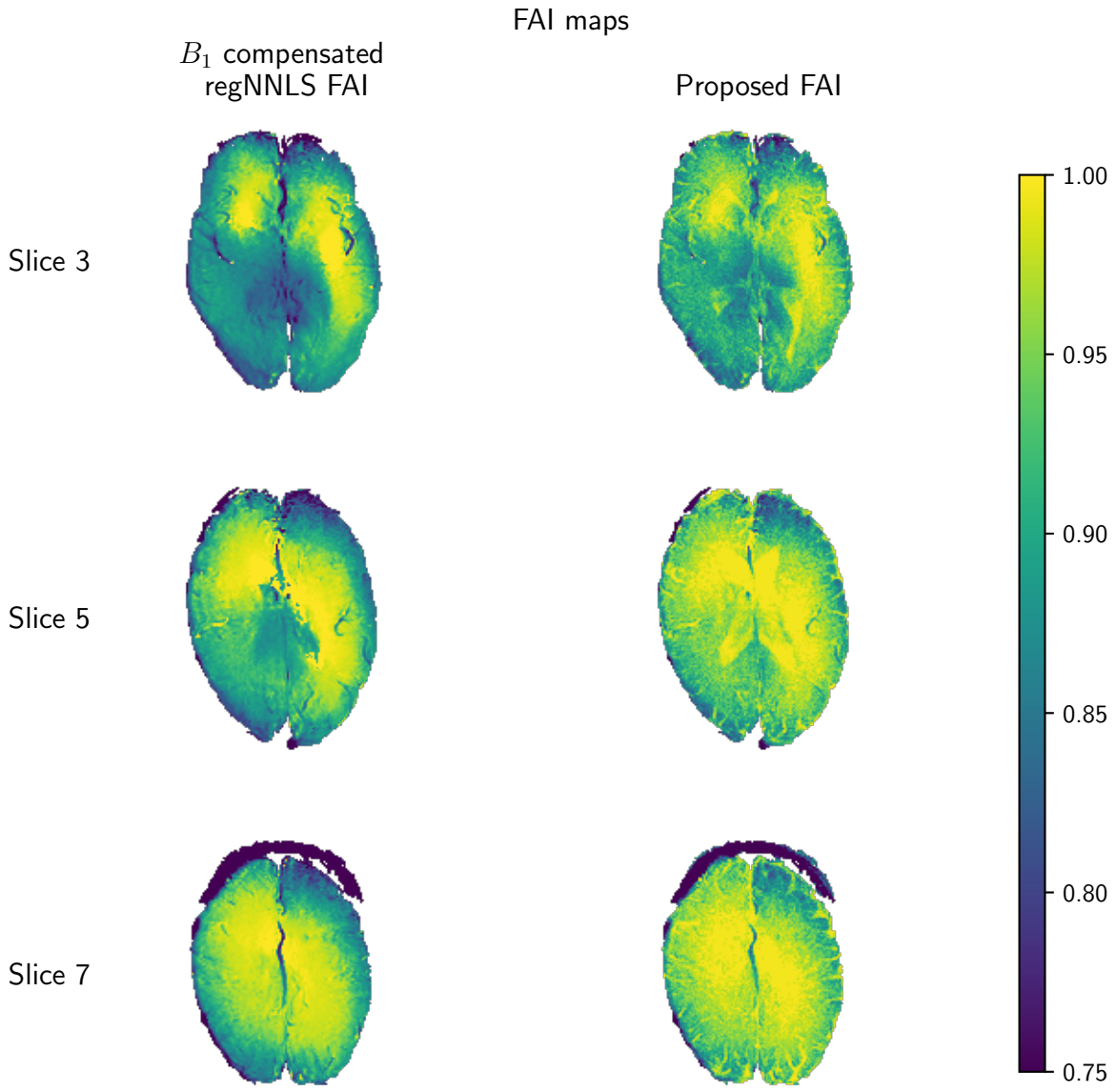


Figure 4.11.: The FAI maps as determined with the B_1-compensated regularised NNLS method [10] and the single component matching for multi-echo spin-echo T_2 data at 3T from a healthy volunteer.

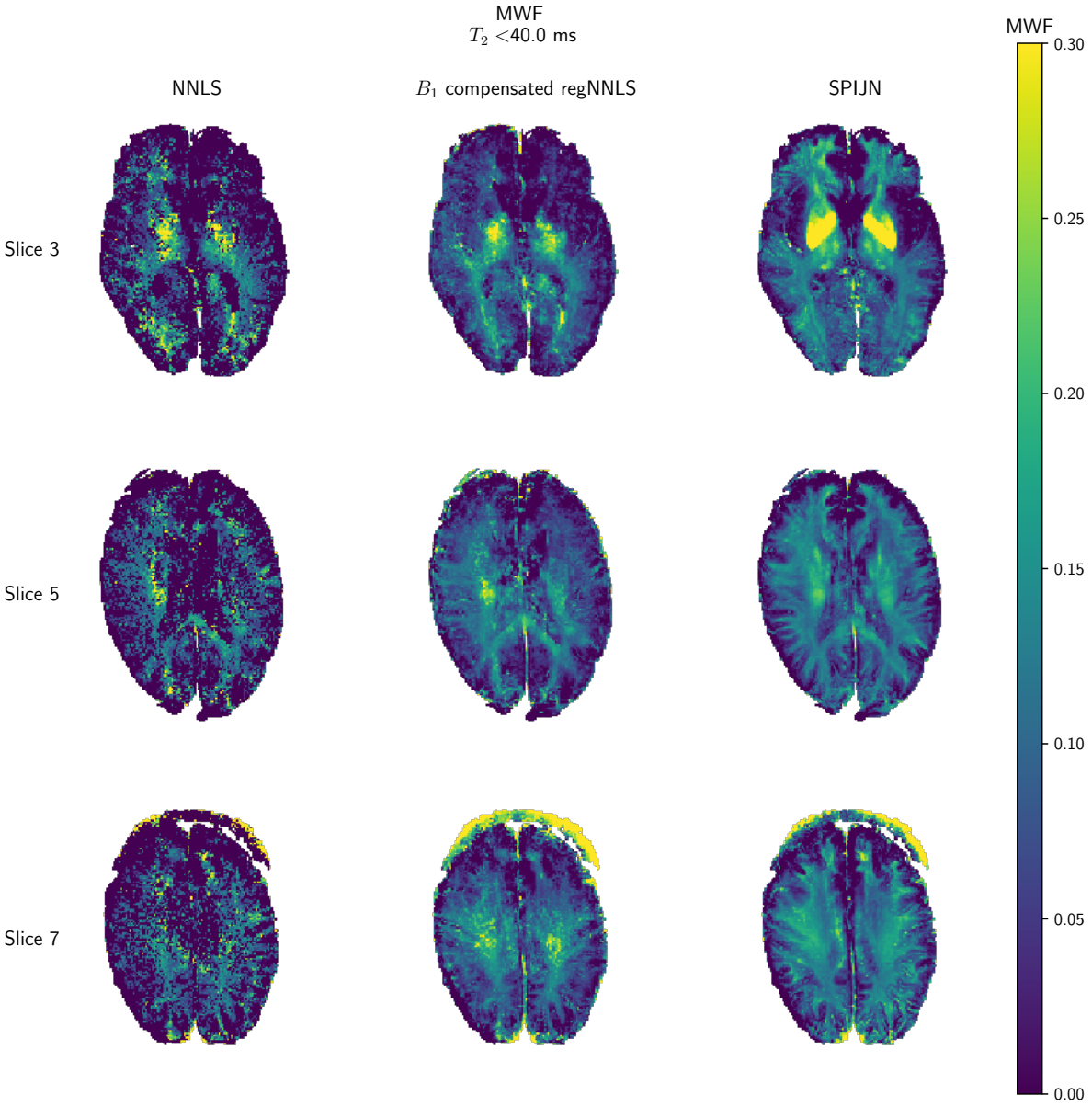


Figure 4.12.: The MWF maps calculated with the NNLS algorithm, the B_1-compensated regularised NNLS method [10] and the SPIJN algorithm for multi-echo spin-echo T_2 data at 3T from a healthy volunteer. The NNLS and SPIJN algorithm use the FAI-map from the single component matching.

Although the FAI maps of both methods look similar, the proposed FAI maps show some brain structure. This could be a drawback of this relatively simple approach assuming a single component. In the ventricles such B_1 offset has been observed before [91]. To prevent this, especially in the outer area, spatial smoothing could be applied to the FAI map.
The NNLS MWF maps show noise amplification, which is reduced by the regNNLS method, with the drawback of longer computation times. The SPIJN MWF maps show better left-right symmetry than the regNNLS maps and a smoother behaviour in general. It seems that the structures in the FAI map do not influence the MWF maps.
The components as resulting from the SPIJN algorithm are shown in Figure 4.13 for the fifth slice. The first component seems related to MW, the second component belongs to the intra- and extracellular water and the fourth component resembles free water. The third component does not seem to resemble a specific tissue, but is mainly located in locations were CSF is present. As said before, the decompositions for different slices or measurements are similar. This matching was also performed for a measurement with less masking and were a part of the muscle outside of the skull was included. This lead to 5 components, the myelin water component was separated in 2 components at 10ms and 29ms. However, this did not change the MWF map.

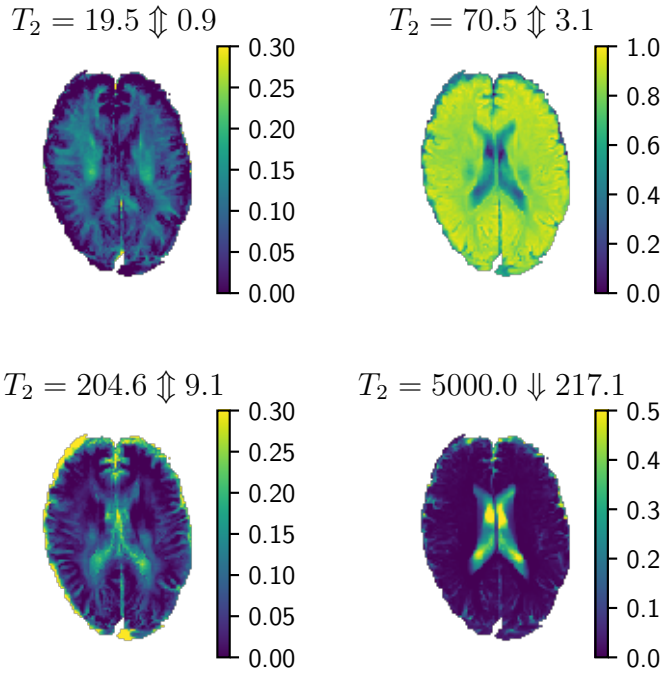


Figure 4.13.: The components as matched with the SPIJN algorithm for a 3T multi-echo spin-echo measurement, slice 5 is shown here.

Multi-echo T_2 measurement including B_1 compensation on a patient with Krabbe disease

In this section the results for a multi-echo spin-echo measurement for a patient with Krabbe disease are presented. Krabbe disease is a lysosomal disorder that affects the white matter of the central and peripheral nervous systems [92], the effects of the Krabbe disease are showing in the myelin water fraction [93]. The MWF for a brain from a patient with Krabbe disease shows a wide-spread lower MWF.
The dataset from the last paper is used to test the SPIJN algorithm. The dataset was obtained with a 3DT_2 Gradient Echo Spin Echo sequence (32 echoes, TR = 1.200 s, TE = 10 ms, resolution = $1.0 \times 1.0 \times 5$ mm^3, slices = 10). In the mentioned paper the MWF is determined using the B_1 compensated regNNLS method. This method includes an estimation of the FAI map. In this thesis the FAI map is obtained through the single component matching including T_2 and B_1 which is then used for the different algorithms. The FAI maps obtained with these two methods are shown in Figure 4.14. The same dictionary as in the previous section was used.
The FAI maps show some differences especially at the outer parts of the brain.
To identify how the differences in the FAI map influence the SPIJN algorithm Figure 4.15 shows the results from the B_1 compensated regNNLS algorithm, the results from the SPIJN algorithm with the FAI map from the regNNLS algorithm and with the FAI map from the single component matching.

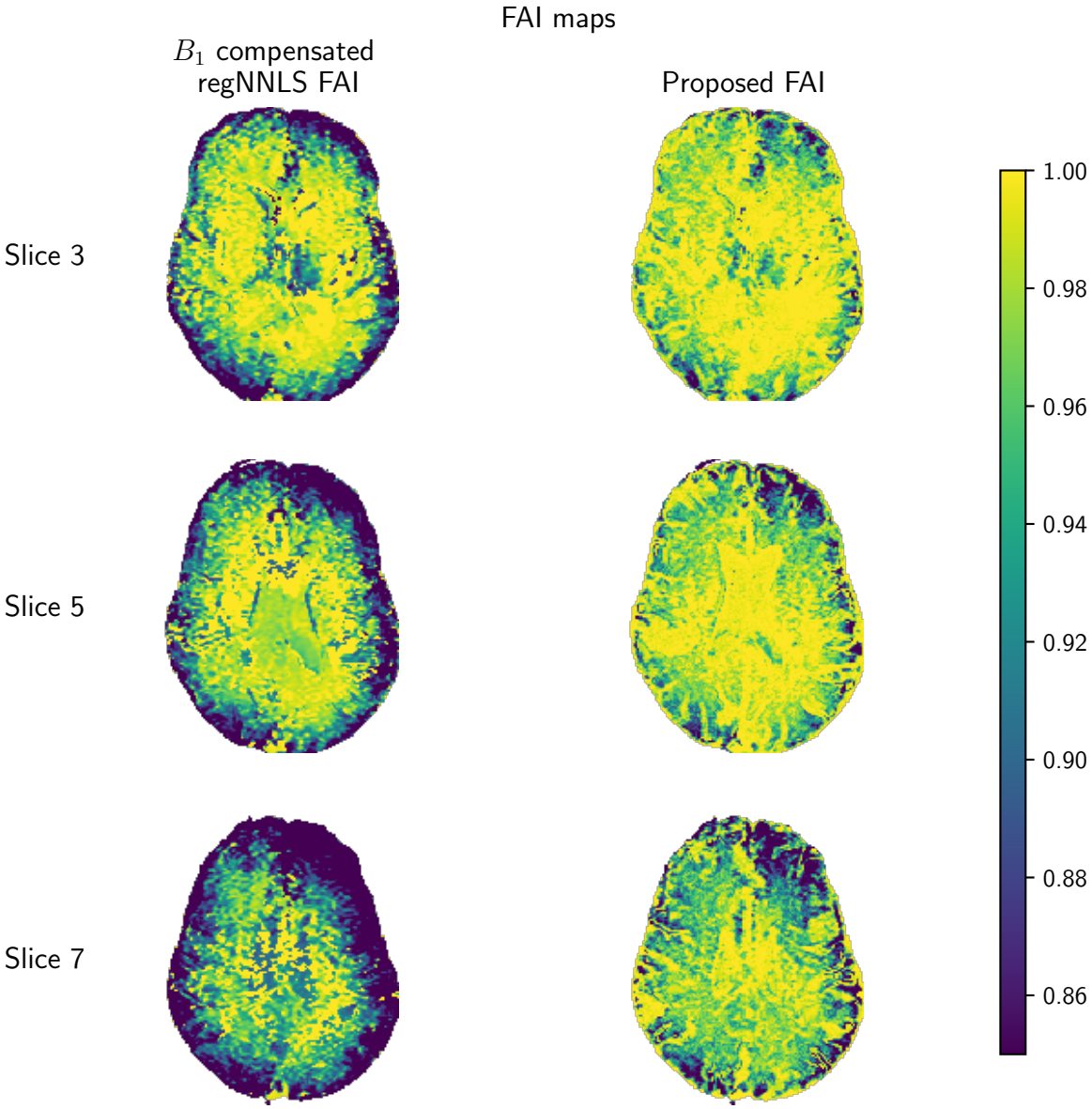


Figure 4.14.: The FAI maps determined the B_1-compensated regularised NNLS method [10] and the single component matching including B_1 and T_2 algorithm for multi-echo spin-echo T_2 data at 3T for a patient with Krabbe disease.

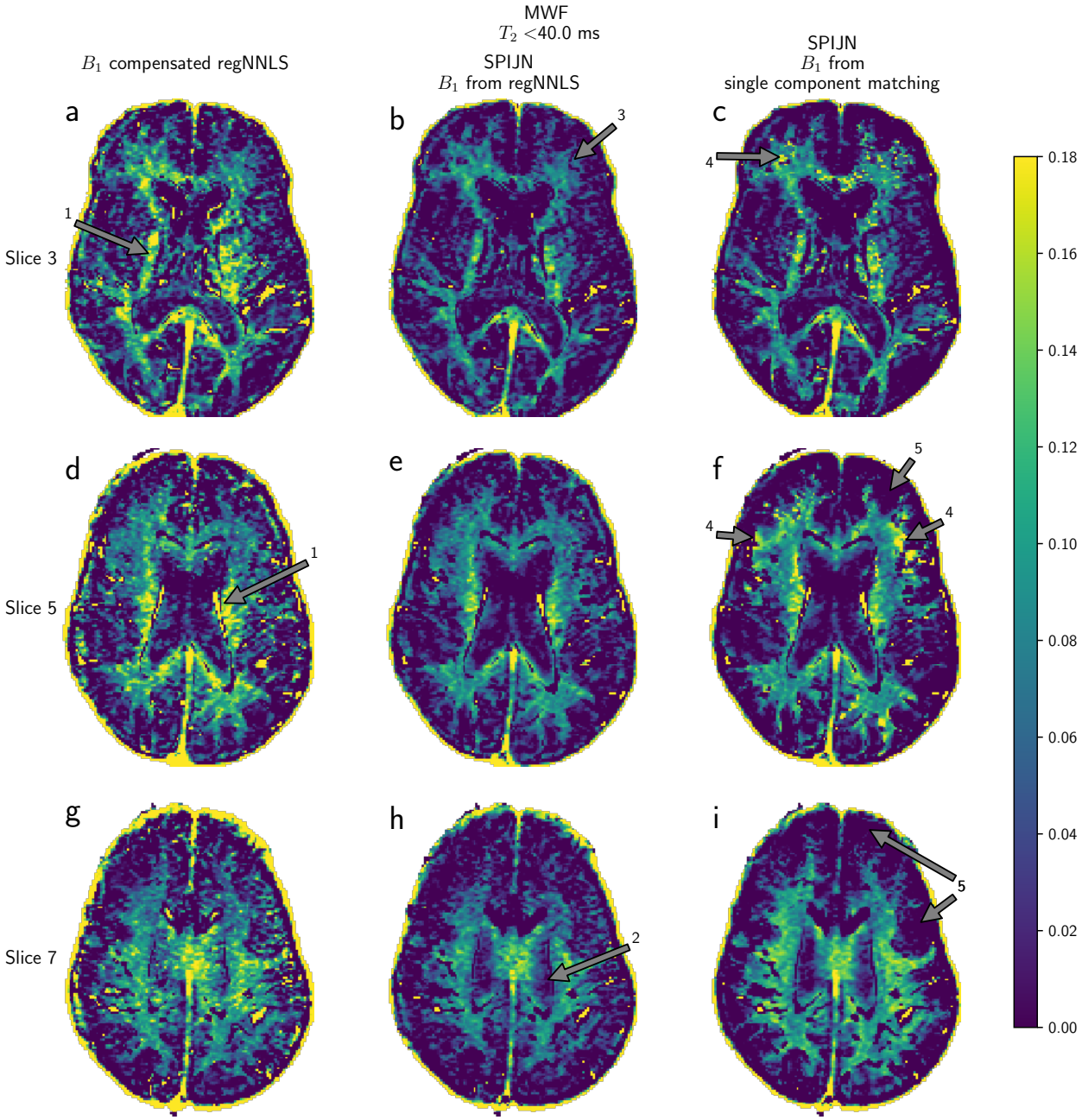


Figure 4.15.: The MWF maps for a patient with Krabbe disease. The Krabbe disease leads to a decrease in myelin water. The plots on the right use the FAI map from the single component matching. The other plots use the FAI map from the B_1-compensated regNNLS. The MWF maps are calculated with the following algorithms from left to right (using the mentioned FAI maps): B_1-compensated regularised NNLS, SPIJN and SPIJN.

These MWF maps show several interesting effects, which are annotated and numbered in the figure, starting with the MWF maps based on the regNNLS FAI map. In the centre of the MWF maps with the regNNLS annotated at **1**, the MWF shows higher intensities which are not shown in the SPIJN maps. Furthermore the SPIJN maps show a more smooth MWF for example at annotation **2** and makes it easier to recognise structures as annotated by **3**. The remark must be made that it is not known whether these structures represent something meaningful. In general the SPIJN MWF map seems a bit calmer than the regNNLS map, although no spatial smoothing is used.
There are also differences between the two SPIJN MWF maps. These differences are mainly at the outer part of the brain. At some place (annotated as **4**) the MWF is high in the single component-based map, which could be considered as unexpected. The opposite effect is also observed and annotated with **5**, where the regNNLS based maps show myelin water, the MWF from the single component matching does not show this myelin water.

4.5.2. In vivo MRF measurements

Comparison of algorithms

A fully sampled MR measurement with sequence MRF200 has been performed. The measurement was performed at 1.5T with the following settings: FOV = $300 \times 300 \times 8$ mm^3, resolution = $2.01 \times 2.01 \times 8$ mm^3, 1 slice. This measurement is used in this section to compare the results obtained with the following 6 algorithms: the NNLS algorithm, the Bayesian approach, the reweighted ℓ_1 method, the SPIJN algorithm, the search around method based on the components found by SPIJN and the JRNNMOP algorithm, one time without segmentation instructed to find 7 components and one time with the segmentation in 3 segments from Section 3.1.3. A logarithmically spaced dictionary was used with T_1 ranging from 10ms to 4s in 100 steps and T_2 ranging from 4ms to 2s in 100 steps, with the restriction $T_2 \leq T_1$ leading to a total of 6219 dictionary elements.
The NNLS algorithm took 26 seconds per slice, the reweighted-ℓ_1 method 4384 seconds, the Bayesian 1324 seconds, the SPIJN algorithm 33 seconds, the search around 104 seconds and the JRNNOMP 23 and 21 (with segmentation) seconds, all performed on a simple laptop.
Figure 4.16 shows how the matched components are distributed for each of the algorithms and also shows how they are grouped. The results for NNLS and reweighted-ℓ_1 are exactly similar for this multi-component analysis. The Bayesian approach shows a broad spread, although still 3 groups, present in the other methods, could be recognised.
The JRNNOMP without segmentation and SPIJN algorithm both return 7 components in similar areas, but these components have different relaxation times. The components matched by SPIJN can directly be related to the distributions of components in the voxel-by-voxel methods. The component at $T_1 = 780$ms, $T_2 = 104$ms, matched by the JRNNOMP, is not matched by any of the voxel-by-voxel methods. The JRNNOMP with segmentation returns 9 components, with most interestingly no component around T_1=10ms and T_2=10ms and two components with very small T_2 and large T_1. The search around as performed on the components found by the SPIJN algorithm shows that most voxels still used the initial components and some are matched to the surrounding components. This spreading around mainly occurs in the T_2 values and less in the T_1.
Figure 4.17 shows the results of the multi-component matching when the matched components are grouped in the same way as described in Section 4.3. As an indication of the quality of the matching the normalised relative mean square error (NRMSE) is given. The Bayesian method and the joint methods (SPIJN and JRNNOMP) show a higher NRMSE than the voxel-by-voxel methods. For the Bayesian method this might be caused by the lack of the non-negativity constraint, which will be discussed in more detail in Section 5.1. For the joint method this is probably an effect of the introduction of the joint sparsity constraint. Introducing the segmentation and the regrouping afterwards increased the NRMSE even more.
For the intra- and extracellular water and free water the differences between the algorithms are minimal. Only the SPIJN algorithm, the search around and the segmented JRNNOMP show a structure in the myelin water.

The 7 components found by the SPIJN algorithm are shown in Figure 4.18. The second component is the component grouped to myelin water. The third and fourth component seem to resemble white and grey matter, although the T_1 relaxation times are shorter than expected. The first component does not seem meaningful, the fifth could be related to the third component in the T_2 joint sparsity matching as shown in Figure 4.13. The sixth and seventh component both seem to represent CSF.
The search around algorithm was applied to the five main components of the SPIJN results the matched components show a small spread as mentioned before. As a result of this the grouping for myelin water in Figure 4.17 shows less structure compared to the SPIJN result. The result for the other two groups do not seem to be changed. The components previously matched to the third (white matter) and the fourth component (grey matter) are however moving towards each other and the white and grey components do not show the structures any more which were observed before.
The 7 components matched by the JRNNOMP algorithm are shown in Figure 4.19. Compared to the components from the SPIJN algorithm these matches show more voxels containing a mixture of components, also at places in the white or grey matter where a more dominant tissue would be expected. The second, short component related to myelin water does not show structures like the matching of the SPIJN algorithm. The fourth component shows a white matter structure, but different from the white matter component from the SPIJN algorithm.
Figure 4.20 shows the 9 components as matched with the JRNNOMP algorithm, using a segmentation into 3 segments and 3 components per segment. After finding the nine components, the regrouping as discussed in Section 3.3.3 was performed. These components are more difficult to interpret. The fourth component seems to include all the intra- and extracellular water. Components 5, 7, 8 and 9 are located in the centre of the brain and could be related to free water, just as component 2. Component 3 seems to be related to white matter or myelin water, which could also be said for component 1.

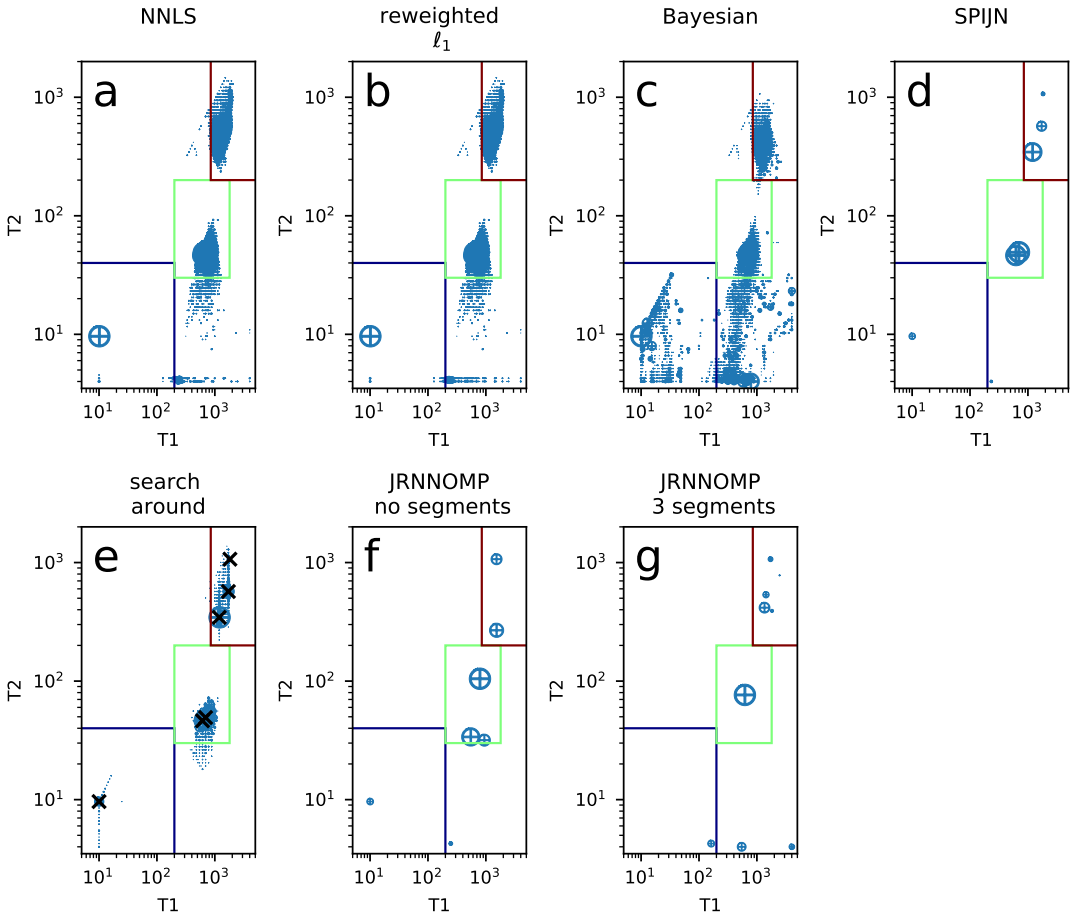


Figure 4.16.: The distribution of the matched components for the different algorithms is shown. The blue box is the short component (myelin water), the green box the middle component (white and grey matter) and the red box the long component (CSF). The size of the dots corresponds to the relative abundance the T_1/T_2 components in the multi-component matching. For the search around the initial T_1/T_2 combinations are shown with black crosses.

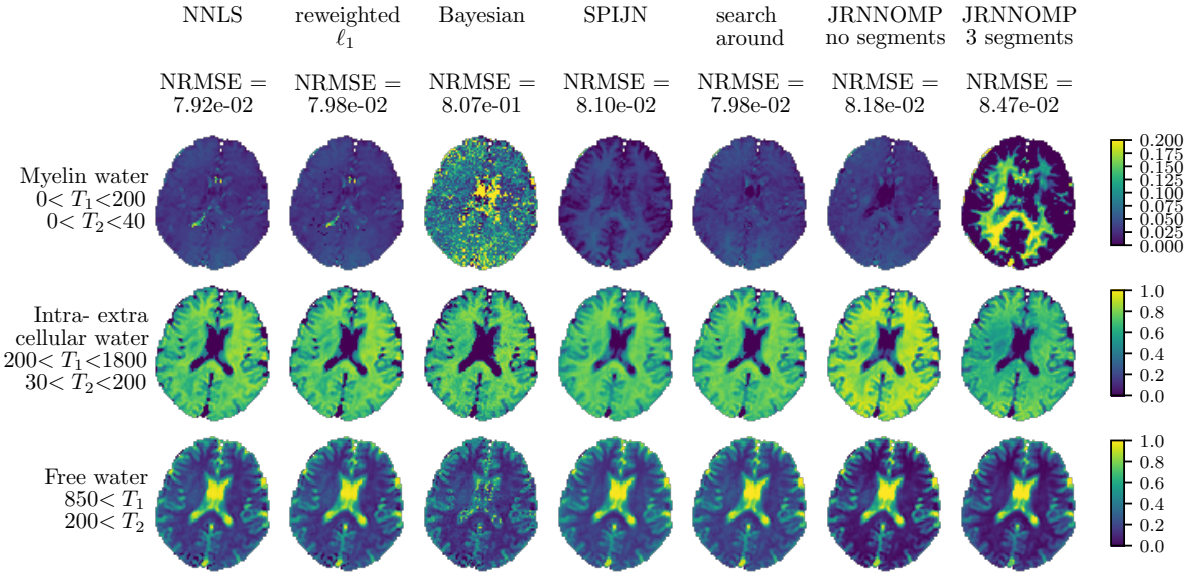


Figure 4.17.: The results of the multi component matching for an in vivo brain MRF measurement. The shown intensities represent the fraction of the component. The rows correspond to the different grouped components and the columns to the different algorithms. The first row has a different colour scale than the lower two rows.

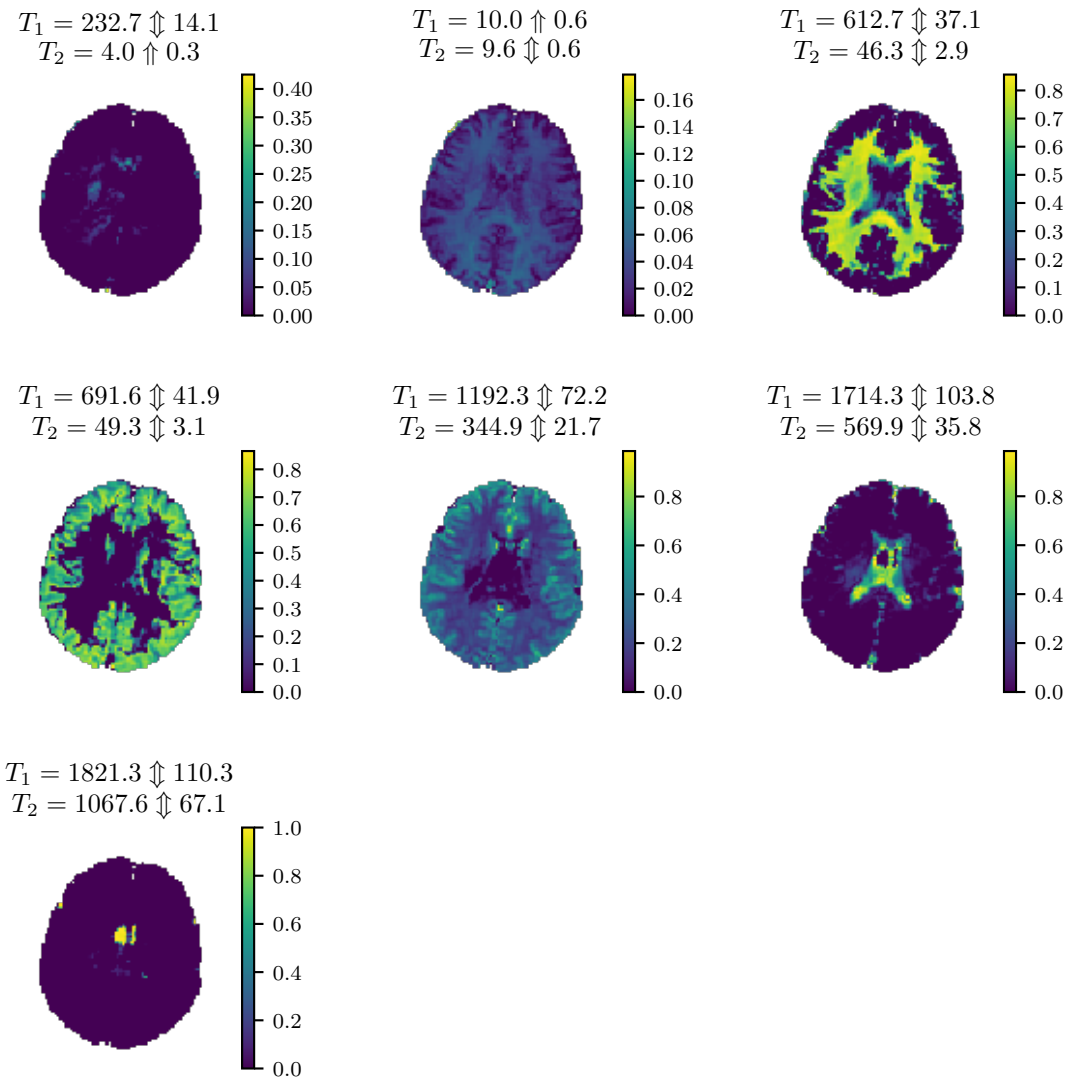


Figure 4.18.: The results of the multi component matching for an in vivo brain MRF measurement with the SPIJN algorithm.

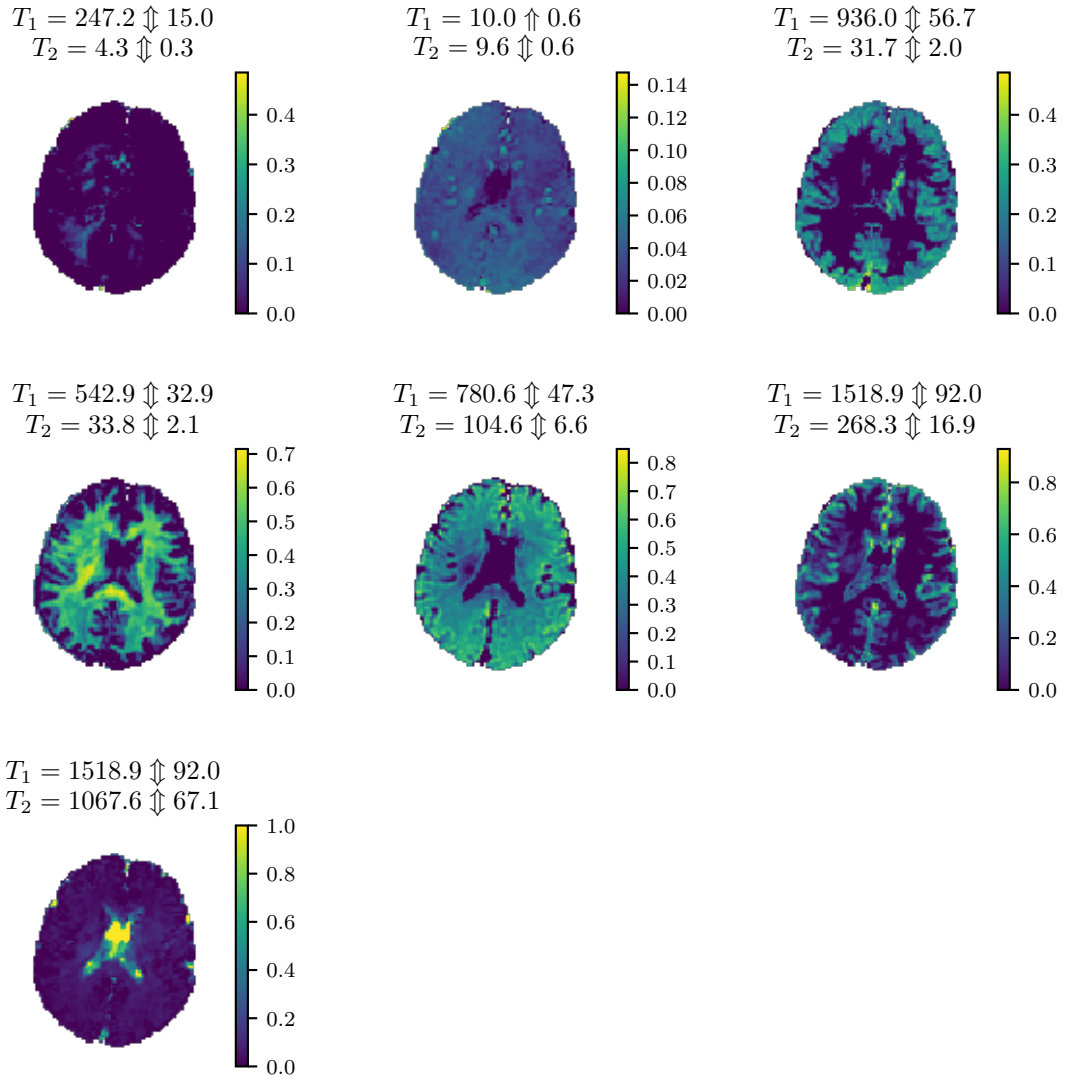


Figure 4.19.: The results of the multi-component matching for an in vivo brain MRF measurement with the JRNNOMP algorithm, where the desired number of components was set to 7.

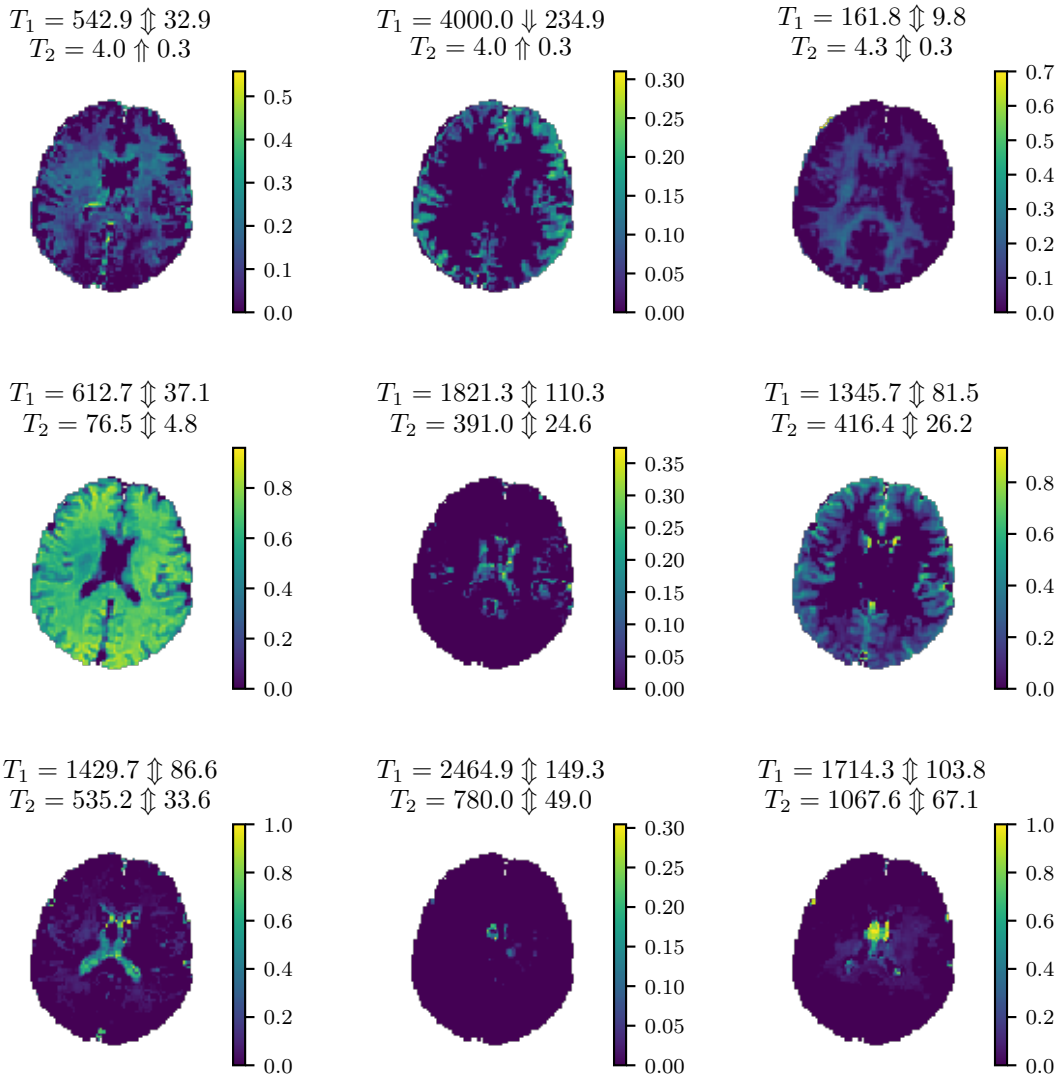


Figure 4.20.: The results of the multi-component matching for an in vivo brain MRF measurement with the JRNNOMP algorithm, where the brain was segmented in 3 segments and in each segment 3 component were sought. The 9 found components were then regrouped to remove the separation of the different segments.

Effects of undersampling

To evaluate the effect of undersampling on the multi-component analysis for in vivo measurements, several undersampled and a fully sampled measurement were performed on a healthy volunteer. A fully sampled measurement with the MRF200 sequence was performed, with the following parameters: FOV = $240 \times 240 \times 15\text{mm}^3$, resolution = $1 \times 1 \times 5\text{mm}^3$ and 3 slices. Beside this fully sampled measurement several undersampled measurements were performed. The measurements were performed with a spiral trajectory, out of the 34 spiral interleaves needed to fully sample the k-space, only 1 or 2 interleaves were sampled. The undersampled measurement were performed with the MRF200 and MRF500 sequence, using a repetition time of 15ms and a relaxation interval of 15ms. The acquisition times were as follows:

- MRF200 fully sampled: 359 seconds;
- MRF500 1 out of 34: 88 seconds;
- MRF500 2 out of 34: 175 seconds;
- MRF200 1 out of 34: 65 seconds;
- MRF200 2 out of 34: 130 seconds.

A logarithmically spaced dictionary was used with T_1 ranging from 10ms to 4s in 100 steps and T_2 from 4ms to 2s in 80 steps, with the restriction $T_2 \leq T_1$ this leads to 4974 dictionary atoms. The components from the matching using the SPIJN method for the fully sampled data are shown in Figure 4.21. The multi-component matchings for the undersampled MRF200 data showed severe spiral artefacts. The results from the MRF500 sequence were better and 2 interleaves show less artefacts than 1 interleave. Figure 4.22 shows the components found with the SPIJN algorithm for the MRF500 sequence with 2 out of 34 interleaves.

The first component matched to $(T_1, T_2) = (10, 9.5)$ in the fully sampled data could be related to myelin water and a similar component is found in the undersampled data at (42.7,42.4). The second component at (691,54) seems related to white matter and is found in the undersampled data at (829,68). The third component at (936,63) is related to the grey matter and is partly reflected in the undersampled data at (1122,93). The fourth component at (1429,383) seems to be complementary to the first component and might be related to free water. This component seems also related to the fifth component in Figure 4.18.The fifth component at (1821,841) would be related to free water and seems related to component matched to the longest component (4000,2000) in the undersampled data.

The first two components in the undersampled data can not be directly related to the components in the fully sampled data. The first component (262,4) seems only present in the white matter and could partly explain the higher relaxation times for the white matter in the undersampled data. The second component (4000,4) matched in the undersampled data shows a similar effect as observed in Section 4.4 and resembles an unrealistic combination of relaxation times.

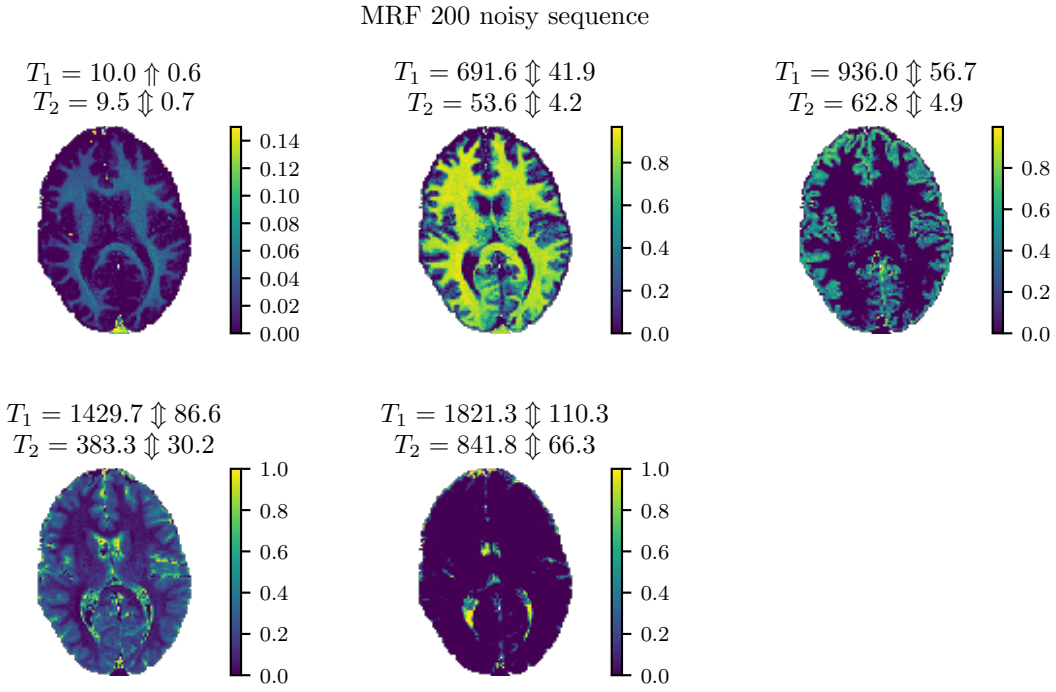


Figure 4.21.: The components from the multi-component matching using the SPIJN algorithm for fully sampled data with the MRF200 sequence. For each component the matched relaxation times are given in milliseconds, including the distance to the closest relaxation times.

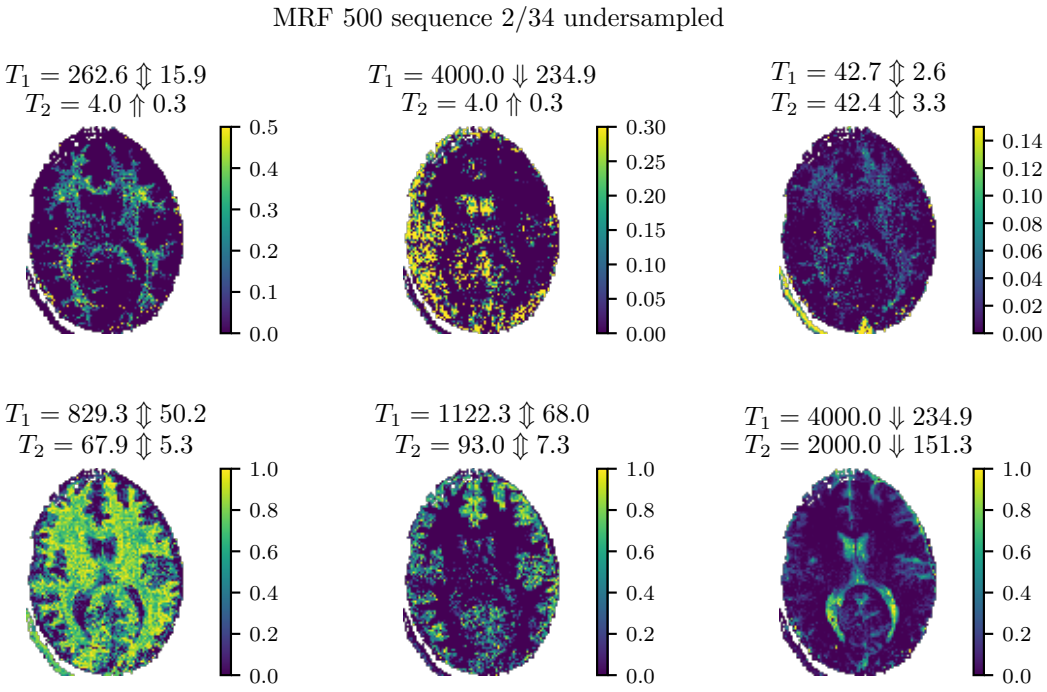


Figure 4.22.: The components from the multi-component matching using the SPIJN algorithm for undersampled data with the MRF500 sequence. 2 out of 34 interleaves were used in the sampling. For each component the matched relaxation times are given in milliseconds, including the distance to the closest relaxation times.

4.6. Further remarks and results on the joint algorithms

Beside the main results as given in the sections before, some additional results and remarks will be given in this section.

4.6.1. Speed up the algorithms

Several strategies to decrease the computation times of the joint algorithms are proposed and were investigated in simple test cases, as discussed briefly in this section. The SPIJN and JRNNOMP algorithms come down to finding a common basis selected from the dictionary atoms which represent the present tissues. If this basis would be known, the dictionary can be restricted and the best solution would be found using the simple NNLS algorithm with these restricted dictionaries, which is rather fast. One could think of several ways to find this restricted dictionary, which will be discussed below.

Fourier transforming the image

One option to speed up the basis selection would be by performing an FFT on the images. The idea would be that the central points in k-space contain the general behaviour of the data. However, the resulting Fourier transform is a complex signal, which can not be projected to the real axes as done before, since the Fourier signal is truly complex and not only a rotated real signal. This makes it therefore impossible to obtain useful results from the Fourier transformed data.

Downscaling the image

Another method is to downscale the images, so by reducing the number of voxels by taking the average over blocks of voxels. As a result of this a smaller number of voxels is considered, making the SPIJN faster and improving the SNR. With the selected basis, the NNLS algorithm can be ran for each voxel. For a simulated image consisting of 3 components this gives faster results and there does not seem to be a negative effect on the solution. This has been tested when the groups of voxels which where combined had size 4 and for SNR$\geq$ 50.
As a variant on this it would also be possible to combine this with the idea from the previous section. By taking the inverse Fourier transform from the central part of the k-space as and using low resolution images it is possible to obtain similar results.

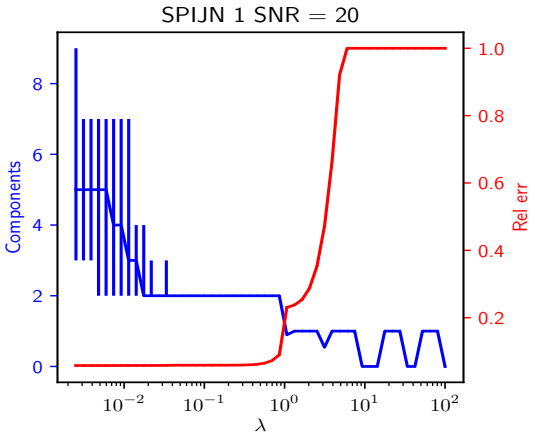
Reuse the basis from different slices

Another interesting option to use the information obtained in different slices as an initial guess for the basis. A drawback of this would be that tissues present in one slice but not in another could be missed.

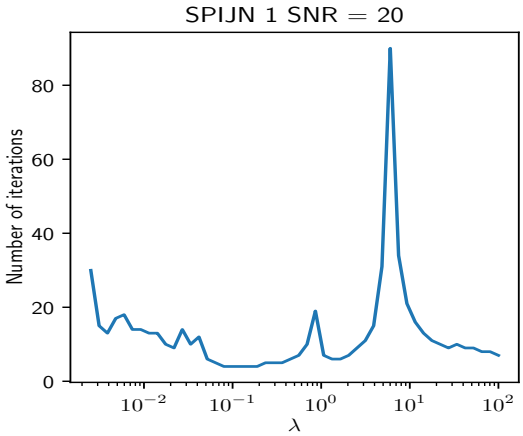
4.6.2. Regularisation parameter of the SPIJN algorithm

The behaviour of the SPIJN algorithm has been analysed with respect to the regularisation parameter λ in a simulation with 2 components. Multiple conclusions can be drawn from these results. The used regularisation value does not influence the matched relaxation times in general. A second observation is that the relative error compared to the regularisation parameter has a critical point at which the relative error drastically increases to a plateau. After a short increase of λ the relative error rises further to one. This behaviour is related to the number of components. It can be seen that the point where the error drastically increases is the point where the number of components decreases from two to one. This increase in the error makes sense, since one component is not enough to estimate the true signal. The desired value of λ lies somewhere before the point where the relative error starts to increase.
Figures 4.23b and 4.23d show the number of iterations needed to reach the convergence tolerance as in Equation 3.26. When the regularisation parameter leads to the true number of components, the convergence is faster than when the regularisation parameter is too large or too small. The number of iterations is the smallest at the value of λ when the number of components has just dropped to the true number of components. In the optimal case only 5 iterations are needed to reach convergence.
In Figures 4.23 and 4.24 the two different reweighting schemes have been compared with the MRF500 sequence and the T_2 exponential decay sequence. When the regularisation is too strong, it can happen that zero components are matched, thus the zero solution is returned. From these plots it can be seen that the behaviour of the different reweighting schemes is similar, but the λ value where the relative error starts to increase is larger for the third scheme than for the first scheme. Another difference is that the third reweighting scheme needs more iterations for convergence when the regularisation parameter is not in the right range.
In Figure 4.25 the behaviour of the SPIJN algorithm with the first reweighting scheme, using different numbers of voxels and different regularisation parameters has been shown. It can be seen that the behaviour is similar for different numbers of voxels. The only difference is that the curve is shifted to the right with a constant spacing in this log-log plot. This constant shift value on a log scale indicates a constant factor of $\log J$ where J is the number of voxels. In Figure 4.26 the relative error against different values of $\bar{\lambda}$ with different number of voxels is shown. The value $\bar{\lambda}$ is the scaled version of the regularisation parameter $\lambda = \bar{\lambda}/\log J$. In this figure it can be seen that the rescaling of the regularisation parameter results in a similar behaviour for each number of voxels. This would make it possible to test different regularisation parameters on a small number of voxels, which can be done rather fast and then to rescale the found regularisation parameter to the larger number of voxels. By doing so, it is not necessary to do the computation for the large number of voxels several times.
For the third reweighting scheme there is not such a clear relation between the regularisation behaviour for different numbers of pixels.
In Figure 4.27 the regularisation behaviour for different SNR levels is shown. For all the different noise levels the relative errors show a same behaviour without a shift as in

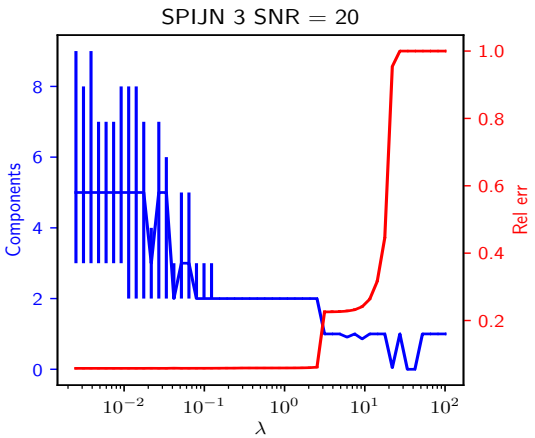
the case for different numbers of voxels. The behaviour of the number of components found is however different. The plateau where the number of components is correct, 2 in this case, is larger for the larger SNR values, which could be expected, since it would be easier to recognise the true components in the less noisy images. The third reweighting scheme shows exactly the same behaviour.



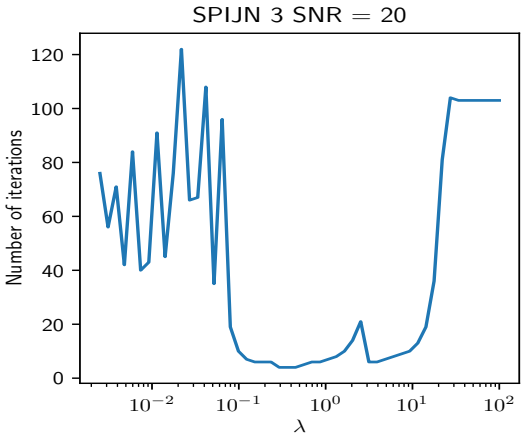
(a) The behaviour for the mean number of components and relative error with error bars.



(b) The number of iterations for different values of λ. The method is stopped for $\delta = 10^{-3}$ in Equation 3.26 or the maximum number of 100 iterations.

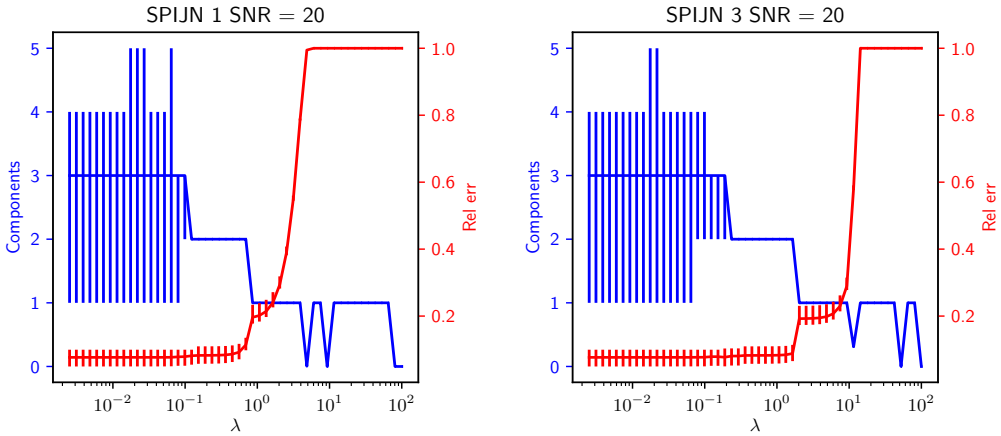


(c) The behaviour for the mean number of components and relative error with error bars.



(d) The number of iterations for different values of λ. The method is stopped for $\delta = 10^{-3}$ in Equation 3.26 or the maximum number of 100 iterations.

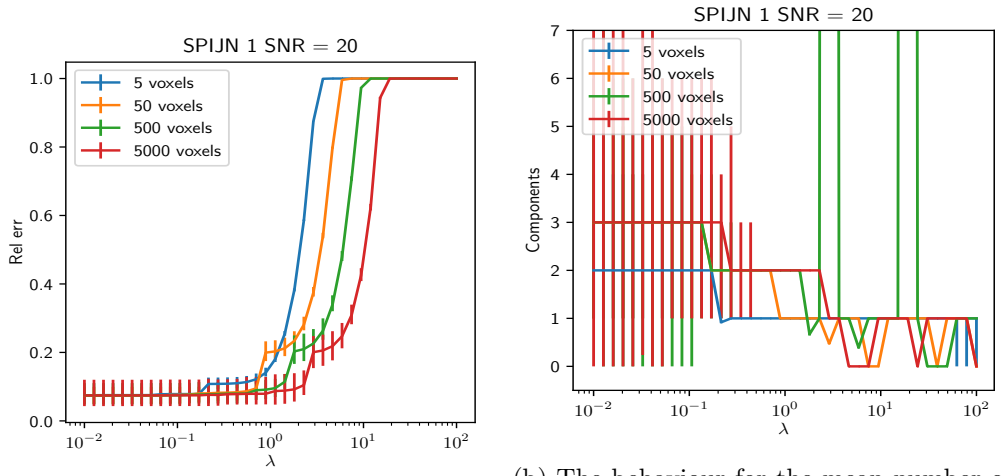
Figure 4.23.: The SPIJN algorithm with reweighting schemes 1 and 3 has been used for different values of the regularization parameter λ for the MRF500 sequence. An image consisting of 25 voxels, containing 2 components with SNR = 20 has been analysed.



(a) The behaviour for the mean number of components and relative error with error bars.

(b) The behaviour for the mean number of components and relative error with error bars.

Figure 4.24.: The SPIJN algorithm with reweighting scheme 1 and 3 has been used for different values of the regularization parameter λ for the T_2 sequence. An image consisting of 25 voxels, containing 2 components with SNR = 20 has been analysed.



(a) The behaviour for the relative error values with error bars.

(b) The behaviour for the mean number of components with error bars. For small λ the number of components varies a lot.

Figure 4.25.: The SPIJN algorithm with reweighting scheme 1 has been used for different values of the regularization parameter λ for the T_2 sequence. Different numbers of voxels have been tried to analyse the shift of the optimal λ value.

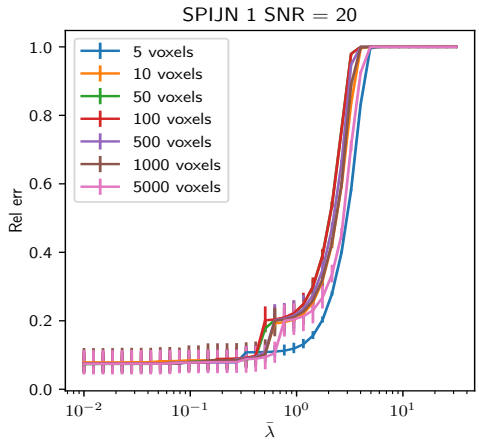
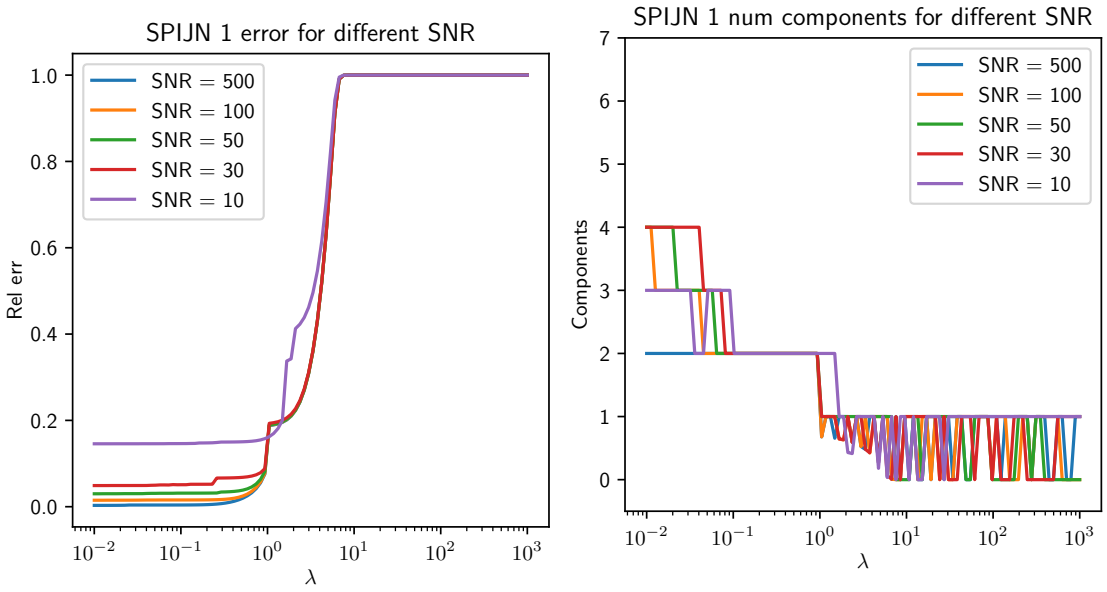


Figure 4.26.: The behaviour for the relative error values with error bars. The SPIJN algorithm with reweighting scheme 1 has been used with regularization parameter $\bar{\lambda}$ for the T_2 sequence. The regularization parameter used in the algorithm is defined as $\lambda = \bar{\lambda}/J$, where J is the number of voxels.



(a) The behaviour for the relative error values.

(b) The behaviour for the mean number of components.

Figure 4.27.: The SPIJN algorithm with reweighting scheme 1 has been used for different values of the regularization parameter λ for the T_2 sequence. The behaviour for different SNR values is shown.

4.6.3. Finding a joint basis with unrestricted basis signals

To illustrate the special properties of the joint-sparsity problem with a given dictionary, a short experiment has been performed as described in this section.
In the multi-component matching as considered in this project a highly coherent dictionary with a non-negativity constraint is used to reconstruct the signal. When the dictionary is allowed to contain any kind of signals, a small common basis would be sufficient to represent the signal. This has been tested for the dataset as discussed in Section 4.5.2
Using the singular value decomposition (SVD) of the matrix containing all the measured signals, can lead to such an unrestricted dictionary. The singular values are sorted on their size and the right singular vectors corresponding to the largest singular values are used as the joint basis signals. With this unrestricted, joint basis an approximation of the original signals can be made. The norm of the difference between the approximated signal and the measured signal, the residual, for each voxel is taken as a measure of the matching quality.
The NNLS algorithm has been used to obtain a multi-component matching as well. The relative residuals of this NNLS approximation with the fixed dictionary are compared to the residuals for the SVD approximation when only 2 joint basis signals are used in Figure 4.28. The relative residual of the NNLS approximation is 7.9%, the SVD approximation 5.6%, although this large difference is mainly caused by the CSF. It can be seen from the

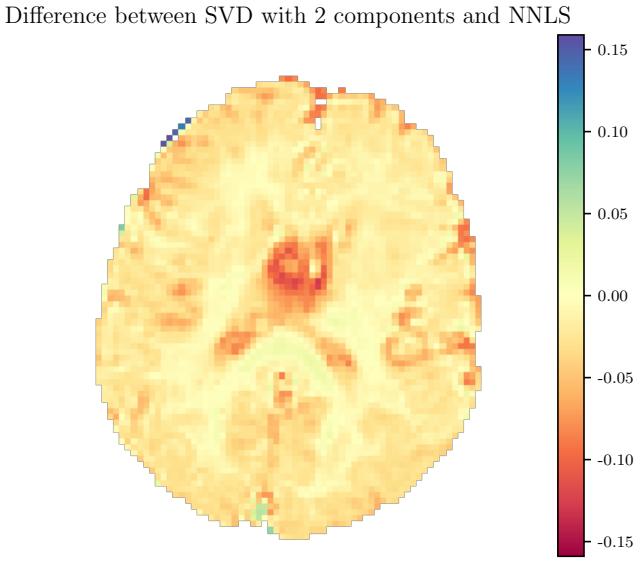


Figure 4.28.: The difference between the residual from an SVD based approximation using two joint basis signals and the NNLS algorithm without joint sparsity restrictions. A negative value (red) indicates that the SVD approximation is better than the multi-component approximation with a realistic dictionary.

figure that the SVD approximation, even with only two singular vectors, gives a better

approximation of the measured signals. The areas where the NNLS approximation is better are mainly the areas where the signal is very different from the rest of the image, mainly in the blood vessel in the lower part of the image.
Using only two signals as a common basis gives very good approximation of the measured signal. Therefore, this test gives an indication of how similar the measured MRF signals are and how difficult it can be to find the different components.

5. Discussion

In this chapter the methods and results as described and shown in the previous chapters will be discussed. This chapter starts with the discussion of the voxel-by-voxel methods, followed by the discussion of the joint methods and their results for simulated, phantom and in vivo measurements.

5.1. Voxel-by-voxel methods

Non-negative least squares as by Lawson and Hanson The standard method as used in the field of T_2 multi-component analysis leads to good approximations in both simulations and measurements. The NNLS algorithm intrinsically finds a sparse solution because of the active set method of the algorithm. Since the NNLS algorithm does not require any prior information besides the dictionary, the NNLS algorithm can easily be used.
Another benefit of the NNLS algorithm is the fast FORTRAN implementation of the algorithm available in the SciPy package as used in this project. This implementation uses a QR decomposition and Householder transformations to perform this as efficient as possible. The Matlab implementation of the NNLS algorithm (`lsnonneg`) relies on the backslash operator for the calculation of the least square solution, which is possibly less efficient than the Fortran implementation of this algorithm.
In Section 4.1 reweighted forms of the NNLS algorithm were considered as well. These modifications lead often to 2 components instead of the 4 components on average without regularisation. The differences in the relaxation times are minimal. A drawback of imposing this regularisation is the need to determine a regularisation parameter and the regularisation can possibly lead to a bias.

Bayesian method The Bayesian algorithm as published in [5] was the first proposal for a multi-component method in MRF and is therefore taken into consideration in most comparisons. The Bayesian method is effectively a reweighted least squares algorithm and therefore comparing this method with a LS algorithm with a simple reweighting in Section 4.1 was interesting. The Bayesian approach could be considered as a special reweighting scheme and for the SNR values of 5 and 10 the Bayesian method gave better results than the reweighted least squares implementation.
In this approach it is assumed that each weight for each voxel is independent and is normally distributed with zero mean and a variance taken from a Gaussian distribution. The assumption of a normal distribution with zero mean leads to negative components and many small components. The assumption that all weights are independent from each other and from other voxels could be doubted as well.

The Bayesian method has three parameters which should be chosen, the shape parameters α and β and the regularisation parameters μ. The parameters as recommended in the original paper [5] were used as a starting point, but it was difficult to obtain a sparse solution by increasing the regularisation. The main way to obtain a sparse solution was the pruning of the dictionary. That the ℓ_2 Tikhonov regularisation is not very effective to obtain a sparse solution is a known problem that occurs here as well.
The method does not take non-negativity into account, this also leads to some disadvantages as can be seen in the in vivo measurement in Figure 4.17 in the myelin water component. Components with short relaxation times are matched to the area of the brain which consists of CSF, which is unexpected since CSF is known to have long relaxation times. This is caused by the lack of a non-negativity constraint, which can be seen from the matching results. In Figure 5.1 the signal evolution for very short and very long relaxation times are shown. This is illustrative for what can happen when the non-negativity is not taken into account. The signal of CSF with long relaxation times looks similar to the component with very short relaxation times, with the difference of a factor -1.

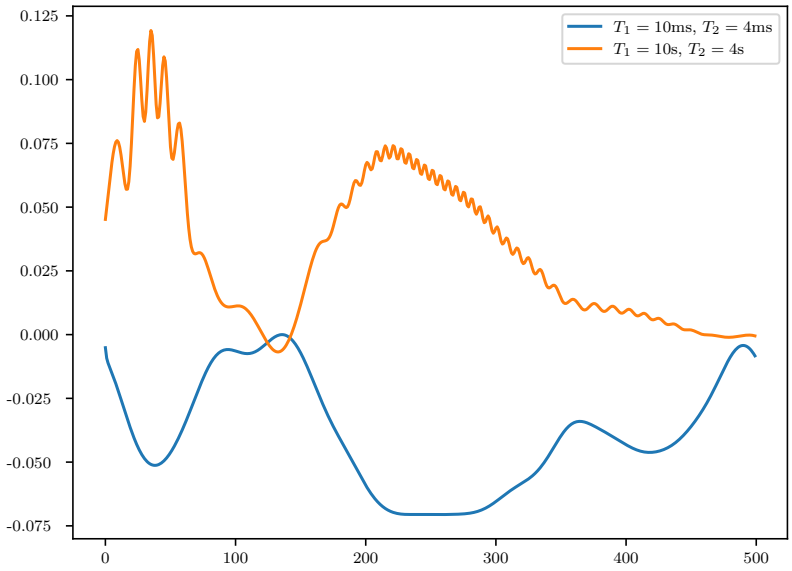


Figure 5.1.: The signal evolution for short relaxation times and long relaxation times.

It is interesting that the algorithm as implemented here is a factor of 63 faster for the in vivo MRF measurement than the computation times as mentioned in the original paper (both were performed on a desktop computer). This could partly be explained by the use of compression of the measured signals, in the original paper it is not clear whether this

is used. Besides this, it could be that the Scipy sparsity and Numpy packages as used are faster than the standard Matlab tools used in the original implementation. Just as in the original paper a conjugate gradient least squares implementation was used and this CGLS solving step takes approximately 50 percent of the computation time, combined with the large number of iterations needed (> 100) because of the slow convergence to a sparse solution, the method is not as fast as could be expected from a reweighted least squares method.
The Bayesian approach gives an interesting first try, but shows several drawbacks. In the field of T_2 multi-exponential analysis more sophisticated Bayesian approaches were proposed [52, 94] and show the possibilities a Bayesian approach can offer.

Reweighted-ℓ_1 algorithm The reweighted ℓ_1-norm regularised method as proposed in [6] implements both a non-negativity constraint and ℓ_1-norm regularisation. The implemented ℓ_1-norm regularisation is an effective way to obtain sparse solutions and as seen in the simulations in Section 4.3 the results are reasonable. The non-separable reweighting scheme shows small differences with the simple reweighting scheme, as can be seen in the results in Section 4.1. The original paper does not conclude which reweighting scheme is preferable and this lack of clarity is understandable from the here performed simulations.
The most interesting outcome is that results from this specifically designed algorithm are very similar to the general purpose NNLS algorithm. An advantage over the NNLS is that the regularisation offers a way to improve the sparsity of the solution, with the drawback that this regularisation parameter has to be determined, probably leading to increased computation times. However, this small advantage does not outweigh the increase of computation time with a factor 168 (from the computation times as in Section 4.5.2).
The original paper does not explicitly mention the computation times, but from a figure comparing different inversion methods the computation time per voxel is between 0.1 and 1 second, which is comparable to the computation times measured here. The reweighting in this algorithm seems very effective, only 3 or 4 outer iterations are needed on average, however, many inner iterations are needed for the interior point solver. This solver needs to balance between the log-barrier keeping the solution away from negative values and the sparsity which tries to push the small components back to 0. To improve the calculation times, it would probably be necessary to find an algorithm which includes the non-negativity and the ℓ_1-regularisation in a more effective way. The Woodburry inversion as proposed in the original paper gave a serious improvement in the computation times. The weight calculation of the non-separable reweighting scheme takes approximately 8 percent of the computation time, which is rather high.

(Regularised) Support shrinkage Non-Negative OMP SNNOMP provides a different approach to find a multi-component matching, making it possible to directly set the number of components. OMP-like algorithms are in general designed for uncorrelated dictionaries and this leads to certain disadvantages with the highly correlated MRF-

dictionaries as used here.
SNNOMP finds a basis consisting a small number of dictionary atoms, which is then used to find a NNLS solution with this basis. SNNOMP expands the basis every iteration by picking the atom which improves the approximation as most as possible. The effect of this is that the first basis element is the result of the single component matching, the components picked afterwards try to compensate for the errors by this rough approximation by picking component with more extreme relaxation times than necessary. An effect of this is that the SNNOMP uses the components with the largest and smallest relaxation times more often than the other algorithms. When the maximum number of components is very small this effect occurs more often. By switching to the RNNOMP algorithm this effect is partly tackled, with the drawback of increased computation times, since more iterations are needed.
When the maximum number of components is increased the results of the SNNOMP and RNNOMP become the same as the results of the NNLS algorithm on which the SNNOMP relies heavily. When the number of components was set to exactly 2 in the single voxel simulations, the results were very different from the results as shown in Section 4.1 where the maximum number of components was set to 7.

5.2. Joint methods

The core of this project is the introduction of the joint sparsity constraint in the field of multi-component analysis. Both in T_2-relaxation and MRF multi-component analysis this was not done before. In section 4.3 it can be seen that the introduction of the joint sparsity constraint gave an improvement in the noise-robustness. Two different methods to impose this joint sparsity constraint were considered in this project, which will be discussed below.

5.2.1. JNNOMP and JRNNOMP

The JNNOMP and JRNNOMP algorithms, developed in this project, implement the joint sparsity constraint such that the number of components can be set on forehand. While performing the first experiment with the JNNOMP and JRNNOMP algorithm, it turned out that the regularised version (JRNNOMP) gave improved results just as observed for the SNNOMP and RNNOMP algorithms. However, in all simulations the JRNNOMP algorithm was outperformed by the SPIJN algorithm.
A drawback by the JRNNOMP method is that the algorithm will always find the number of components as set, even if the true number of components is lower. It tends to pick extreme values, a behaviour which was already observed with the RNNOMP algorithm, to form a better approximation of the signal. An effect of this is that many voxels contain a mixture of components, as can be seen in the results for in vivo data in Figure 4.19. The component as found for the measured MRF data in Figure 4.16 shows that the components found by the JRNNOMP do not always correspond from the components as found by the single-voxel methods.

The segmentation and regrouping gave improvements in the simulations and had a clear, but questionable effect on the in vivo data. This lead to several components with unrealistic combinations of relaxation times (e.g. $T_1 = 4$s and $T_2 = 4$ms) and again gave mixtures in most voxels.
The introduction of the joint sparsity constraint in the RNNOMP algorithm gave a small improvement in the computation times, which is a beneficial effect. The S-CoSAMP (without non-negativity) or JRNNOMP might be effective for incoherent dictionaries, but for the coherent dictionaries as required in MRF the approximation effects of the orthogonal matching pursuit lead to clear drawbacks for measured data.

5.2.2. Sparsity promoting iterative joint non-negative least squares

The SPIJN algorithm as introduced in this project combines the fast and robust NNLS algorithm with the joint sparsity constraint. The performance in simulations and for the in vivo data will be discussed here, followed by a discussion of the implementation.

Simulations In the comparison with the voxel-by-voxel methods, the SPIJN algorithm showed to be more precise and more accurate than the voxel-by-voxel methods. The SPIJN algorithm shows to give am even better high noise performance in the simulations of Section 4.2. This improved performance might be caused by the variance of the weights of the two components. An interesting effect can be seen in Figure 4.3a where the largest error in the matched weight, around 1 percent, is in the part where the first component is the only component. With a lower SNR this effect can still be seen and indicates that the SPIJN algorithm does not always recognise a pure tissue.
The comparison of Section 4.3 show as clear advantage of the SPIJN algorithm over the voxel-by-voxel methods and the JRNNOMP algorithm. The SPIJN solution showed a smaller error compared to the ground truth and a smaller variation in the myelin water like component. From Figure 4.6 it can be seen that the SPIJN algorithm groups the matched components from the individual voxels, which thus results in a lower error.

Fully sampled MRF data The results from the fully sampled MRF data in Section 4.5.2 confirm the observation from the simulations that the NNLS or reweighted ℓ_1 algorithms give similar grouped results to the SPIJN algorithm. All the methods find a component with short relaxation times, which could be related to myelin water. However, in the MW-like component from the SPIJN algorithm certain structures seem to be present, which was not recognised with the other algorithms.
From Figure 4.16, showing the components for the MRF brain measurement for different methods, it can be seen that it is difficult to distinguish the white and grey matter in the intra- and extracellular water from the component matched by the NNLS algorithm. The decomposition from SPIJN between white and grey matter as shown in Figure 4.18 is therefore an interesting result. The matched relaxation times are much shorter than the relaxation times as expected from literature, although this effect has been observed earlier for single component matching in MRF. Another problem is that not all of the

matched components can be related to known tissues. The distribution of the found components are similar to the components as found by [53], however the relaxation times mentioned there are very different.
The results from the second MRF measurement in Section 4.5.2 show a similar decomposition as in the first MRF measurement. The matched relaxation times are close to the relaxation times from the previous measurement. Both of these measurements used the same MRF-sequence and it would be good to investigate the effect of different sequences on the matched relaxation times.
The importance of the correctness of the matched relaxation times can however be questioned. The value of the relaxation times does not reveal specific information about the tissue and could therefore be considered as just a tool to distinguish different tissues from each-other, not to recognise a tissue based on the relaxation times.

Undersampled MRF data The undersampling of the data resulted in different relaxation times and more noise in the weight maps. Out of the four undersampled measurements performed, only the measurement with the highest SNR resulted a decomposition which could be related to the fully sampled measurement. The weights maps show more noise than the fully sampled measurements and the relaxation times are slightly different, although this could also be caused by the different MRF sequences used here.
The use of undersampled data thus probably requires some pre-processing of the data, for example iterative reconstruction to remove the undersampling artefacts, before the SPIJN algorithm can effectively be applied. When the voxel-by-voxel methods were used the results were not interpretable or groupable.

MWF from multi-echo spin echo T_2 data The MWF maps as obtained using the SPIJN algorithm for the multi-echo spin echo measurements as shown in Section 4.5.1 show more symmetry than the state of the art regNNLS method, which could be considered as an improvement. For the Krabbe data the differences were smaller and the greatest improvement by the SPIJN algorithm is the reduced computation time. The SPIJN algorithm resulted in MWF maps which were in general smooth, although no spatial smoothing was involved.
The matched components are more difficult to interpret and unfortunately the white and grey matter can not be distinguished from their T_2 relaxation times. A component with a relaxation time around 204 ms is found as well, which is not expected from literature. This component is mainly present in the areas related to grey matter and CSF, but attempting to assign a certain meaning would only be guessing.

Implementation of SPIJN The SPIJN algorithm stands out compared to the other algorithms because of the short computation times. Only the NNLS algorithm is faster. The speed of the algorithm can mainly explained by the speed of the used NNLS algorithm and the small number of iterations (less than 20) needed for convergence. The pruning which is implemented also gives a drastic increase in the computation times, since for large dictionaries more than 90 percent of the components can be pruned after 2 iterations. A

last novelty which helps for the fast computation time is the use of the ℓ_1^2-regularisation as explained in Eq. (2.19), leading to the use of only 1 extra row instead of N, where N is the number of components. The adding of these thousands of rows would increase the computation times drastically and make the benefits from compression negligible.
The speed of the SPIJN algorithm could even be further improved when the NNLS step would be performed in parallel and when the results from certain calculations would be exchanged between the NNLS solve for different voxels. Steps like this were tried to be implemented in Python, but the NNLS as implemented in FORTRAN and used was still much faster.
An advantage of the SPIJN algorithm over the voxel-by-voxel methods as implemented here is that certain preparation steps can be performed for the complete image at once, instead of voxel by voxel. The NNLS algorithm using the same voxel-by-voxel framework as the Bayesian and reweighted-ℓ_1 method took approximately 21 seconds for a complete image, when the SPIJN algorithm with only 1 iteration was used it took 10 seconds, showing a speed up of 11 seconds. For the Bayesian and ℓ_1-reweighted algorithm these 11 seconds are however negligible.
In the current use case the regularisation parameter has to be chosen manually, which is a drawback when many datasets are considered. However, the results are not that sensitive to the regularisation parameter and the same regularisation can be used for different slices when the parameter is scaled with respect to the number of pixels. In section 4.6.1 some strategies to further speed up the algorithm were mentioned.

Reweighting Two reweighting schemes were proposed and compared. The first was a simple scheme and the other a non-separable scheme which included the similarity between the different dictionary signals. The non-separable scheme only showed small improvements and was unfortunately not applicable to problems were a fixed parameter map was considered. Another advantage of the simple scheme was the option to include the pruning and the easier interpretable behaviour with respect to the regularisation parameter.

5.3. Joint sparsity constraint or a smooth spectrum

The SPIJN algorithm heavily relies on the joint sparsity constraint. The main premise for the joint sparsity constraint is that the tissues throughout the region of interest can be represented by a small group of T_1/T_2 combinations. A similar assumption was made in [53]. This constraint could be considered as the opposite of a smooth spectrum as used in T_2 multi-component analysis.
To test the effect of the joint-sparsity constraint the search around method (Algorithm 9) was used to see how this influenced the found solution, as showed in Section 4.5.2. Loosening this condition did not improve the solution. Not all structures as found by the the SPIJN matching are recognisable after the application of the search around algorithm.
The assumption that the T_2 distribution is smooth as used in myelin water mapping from T_2 data was introduced in [8] and confirmed in [24] as an effective way to address the

ill-posed nature of the inversion problem. The smooth spectrum is found through this regularisation and no publications could be found indicating that a smooth spectrum has a physical explanation. A smooth spectrum only seems to occur when smoothness improving methods are used. That these results are more reliable can be explained from introduction of additional assumptions and the regularisation as an effect of this.
The proposed joint sparsity constraint is another form of introducing extra assumptions and lead to similar results for the MW-mapping. It raises, however, questions how some of the found components should be interpreted, since they do not always seem to represent known structures. At the same time there is only a little knowledge on the relaxation times in multi-component matching and which assumptions can be made or not. The observation that similar structures were observed for different slices and measurement, has an affirmative effect.
Since each voxel influences the outcome in the joint sparsity constraint, the masking becomes more important. The method as proposed in this project gives decent results, but more advanced skull-stripping methods could be considered.

5.4. MRF compared to T_2 relaxation

Using the varying flip angle as proposed by MRF was expected to lead to several positive effects in the multi-component analysis compared to T_2 multi-component analysis.
From the multi-component decomposition as showed in Figure 4.13 and Figure 4.18 it can be seen that more tissue types can be recognised because of the sensitivity for T_1. Using MRF it becomes possible to distinguish white and grey matter.
From the simulations it can be observed that the MRF500 signal is less sensitive to noise than the MRF200 signal which is again less sensitive to noise than the T_2 signal. There might be several reasons for this behaviour. The MRF500 sequence contains more measurements and is therefore less sensitive to noise than the MRF200 sequence or the T_2 measurements using 32 up to 60 echoes. Another explanation is the larger variation in the signals for the MRF dictionary than for an exponential dictionary.
The phantom measurements show a slightly different effect, namely that the SPIJN algorithm was more effective to correctly determine the pure components for the T_2 data then for the MRF500 measurements. Where the voxels consisted of a pure component it was estimated as an approximately 90-10% mixture. This behaviour was not expected based on the simulations, it could be that there were additional effects present in this measurement that were not explained by the signal model used in the dictionary computation.
The in vivo MRF measurements and their multi-component matching (Figures 4.18 and 4.21) resulted in a component with short relaxation times as expected from myelin water, however the MWF of this component (around 5 %) is surprisingly low. More measurement would be necessary to obtain a proper comparison between the MWF-map obtained by MRF and a reference MWF-map obtained by multi-echo multi-spin T_2 measurements.

6. Conclusions

This project was aimed at the of sparsity promoting algorithms for multi-component parameter mapping, for data obtained with magnetic resonance fingerprinting and multi-echo spin-echo T_2 measurements. Multi-component parameter mapping makes it possible to measure the concentration of different components or tissues in a voxel or a certain region. By using a sparsity constraint only a small number of components are found for each voxel and the hope is that these components can be related to different tissues. Two methods to perform such a multi-component analysis in MRF have been published recently [5, 6] and the NNLS algorithm [8, 60], as used for T_2 measurements, incorporates this restriction.

These methods perform the analysis on a voxel-wise basis and do not take information about the rest of the region of interest, for example a human brain, in consideration. As an effect of this, it can be that hundreds of different components are matched throughout the region of interest. To include the notion that only a few number of tissues are present in the region of interest, the joint sparsity constraint is introduced. This joint sparsity constraint enforces to use the same, small set of components (related to different tissues) for each voxel. The Joint Regularised Non-Negative OMP (JRNNOMP) algorithm and the Sparsity Promoting Iterative Joint NNLS (SPIJN) algorithm were developed to implement this joint sparsity constraint including non-negativity for the matched components.

In this project the Bayesian approach from [5] and the reweighted ℓ_1-norm algorithm from [6] were compared to the NNLS algorithm [60, 8] and the two new algorithms with the joint sparsity constraint. The inclusion of the non-negativity constraint as in the reweighted-ℓ_1 and NNLS algorithms showed an improvement compared to the Bayesian approach. The NNLS algorithm as commonly used in T_2 multi-component analysis leads to results comparable to the reweighted-ℓ_1 algorithm with improved computation times. The voxel-by-voxel methods, however, require grouping of the matched components to obtain insights over a larger region of interest.

The joint-sparsity constraint was introduced to implement the notion that only a few tissues are present in the region of interest, making grouping of the matched component easier or even unnecessary. In simulations both algorithms with the joint sparsity constraint showed improvements compared to the voxel-by-voxel methods. For measured data the results from the SPIJN algorithm showed more meaningful structures than the JRNNOMP algorithm and is therefore preferred. The joint sparsity constraint as used by the SPIJN algorithm leads to a small number of components, making grouping of the component, which would require extra assumptions and processing steps, unnecessary.

The SPIJN algorithm was used for the calculation of the myelin water fraction (MWF) for multi-echo spin-echo T_2 measurements and showed similar or improved results compared

to state-of-the-art methods in this field. Combined with the proposed approximation of the flip angle inhomogeneity map from the single component matching, the SPIJN algorithm was a factor 50 faster than the B_1 compensated regNNLS algorithm. A conference abstract about this new algorithm, used for the calculation of myelin water fractions, has been submitted for the 2019 conference of ISMRM (International Society for Magnetic Resonance in Medicine) [95]. A paper, showing a more exhaustive analysis for the results for MWF-mapping, is planned on this topic as well.
The introduction of the joint sparsity constraint in the MRF multi-component analysis leads to easy interpretable results without further grouping of the components. The matched components from the SPIJN algorithm show structures which can be related to myelin water, white matter, grey matter and CSF. It is possible to distinguish white and grey matter, which could not be done easily with the single component methods. An abstract on these results has been submitted for the ISMRM 2019 conference as well[96]. There are, however, also components found which do not directly represent a known tissue. It could be that the joint sparsity constraint makes it more difficult to detect small anomalies, which are repressed by the joint sparsity constraint. It would be very interesting to perform a multi-component analysis on a measurement set in which such anomalies are present.
The SPIJN algorithm requires a regularisation parameter and the choice of this parameter determines the number of components as found by the algorithm. Some first tests have been performed how the regularisation parameter can be chosen, implementing algorithms to optimise the choice of this parameter will, however, lead to increased computation times. Some ideas to do this efficiently were proposed, but not fully tested.
The multi-component analysis by the SPIJN algorithm is not as robust to undersampling effects as the single component matching and therefore probably needs pre-processing of the data to make the use of undersampled data possible.
To find whether the proposed multi-component MRF analysis gives meaningful and consistent results more comparisons should be made. Especially comparisons to the MWF as obtained from multi-echo spin-echo measurements would be useful. A paper, giving more insight in the implementation of the SPIJN algorithm and how it is used for MRF multi-component analysis, including some first results, is planned.
The joint sparsity constraint implemented through the SPIJN algorithm as proposed in this thesis and used for multi-component analysis both in MRF and T_2 relaxation leads to easy interpretable results with meaningful structures within short computation times. Since the computations can be performed in less than a minute on a normal computer it would be possible to use the multi-component analysis in a clinical setting and show the results within a feasible time frame. Where a myelin water fraction map from T_2-relaxation is currently only available after hours of calculations, the proposed SPIJN algorithm would make it possible to calculate these maps for a full brain scan in the time of getting a cup of coffee. This will hopefully lead to improved clinical usage, just as the application of the SPIJN algorithm with multi-component MRF hopefully leads to other new insights.

Bibliography

[1] A. Mackay, K. Whittall, J. Adler, D. Li, D. Paty, and D. Graeb, "In vivo visualization of myelin water in brain by magnetic resonance," *Magnetic Resonance in Medicine*, vol. 31, pp. 673–677, Jun. 1994.

[2] D. H. Miller, R. I. Grossman, S. C. Reingold, and H. F. McFarland, "The role of magnetic resonance techniques in understanding and managing multiple sclerosis." *Brain*, vol. 121, no. 1, pp. 3–24, Jan. 1998.

[3] C. Laule, I. M. Vavasour, G. R. W. Moore, J. Oger, D. K. B. Li, D. W. Paty, and A. L. MacKay, "Water content and myelin water fraction in multiple sclerosis," *Journal of Neurology*, vol. 251, no. 3, pp. 284–293, Mar. 2004.

[4] A. MacKay, C. Laule, I. Vavasour, T. Bjarnason, S. Kolind, and B. Mädler, "Insights into brain microstructure from the T2 distribution," *Magnetic Resonance Imaging*, vol. 24, no. 4, pp. 515–525, May 2006.

[5] D. McGivney, A. Deshmane, Y. Jiang, D. Ma, C. Badve, A. Sloan, V. Gulani, and M. Griswold, "Bayesian estimation of multicomponent relaxation parameters in magnetic resonance fingerprinting: Bayesian MRF," *Magnetic Resonance in Medicine*, Nov. 2017.

[6] S. Tang, C. Fernandez-Granda, S. Lannuzel, B. Bernstein, R. Lattanzi, M. Cloos, F. Knoll, and J. Assländer, "Multicompartment magnetic resonance fingerprinting," *Inverse Problems*, vol. 34, no. 9, p. 094005, Sep. 2018.

[7] D. McGivney, Y. Jiang, D. Ma, C. Badve, V. Gulani, and M. Griswold, "Segmentation of Brain Tissues using a Bayesian Estimation of Multicomponent Relaxation Values in Magnetic Resonance Fingerprinting," in *Proc. Intl. Soc. Mag. Reson. Med. 26*, Paris, 2018, p. 1022.

[8] K. P. Whittall and A. L. MacKay, "Quantitative interpretation of NMR relaxation data," *Journal of Magnetic Resonance (1969)*, vol. 84, no. 1, pp. 134–152, Aug. 1989.

[9] D. Hwang and Y. P. Du, "Improved myelin water quantification using spatially regularized non-negative least squares algorithm," *Journal of Magnetic Resonance Imaging*, vol. 30, no. 1, pp. 203–208, Jun. 2009.

[10] T. Prasloski, B. Mädler, Q.-S. Xiang, A. MacKay, and C. Jones, "Applications of stimulated echo correction to multicomponent T2 analysis," *Magnetic Resonance in Medicine*, vol. 67, no. 6, pp. 1803–1814, Jun. 2012.

[11] A. Raj, S. Pandya, X. Shen, E. LoCastro, T. D. Nguyen, and S. A. Gauthier, "Multi-Compartment T2 Relaxometry Using a Spatially Constrained Multi-Gaussian Model," *PLoS ONE*, vol. 9, no. 6, p. e98391, Jun. 2014.

[12] D. Kumar, S. Siemonsen, C. Heesen, J. Fiehler, and J. Sedlacik, "Noise robust spatially regularized myelin water fraction mapping with the intrinsic B1-error correction based on the linearized version of the extended phase graph model," *Journal of Magnetic Resonance Imaging*, vol. 43, no. 4, pp. 800–817, Apr. 2016.

[13] D. G. Nishimura, *Principles of Magnetic Resonance Imaging.* Stanford University, 1996.

[14] M. H. Levitt, *Spin Dynamics Basics of Nuclear Magnetic Resonance, Second Edition*, ser. Spin dynamics basics of nuclear magnetic resonance, second edition. John Wiley & Sons Ltd, 2008.

[15] M. Hammer, "MRI Physics: MRI Pulse Sequences - XRayPhysics," http://xrayphysics.com/sequences.html#se, 2014.

[16] K. G. Hollingsworth, "Reducing acquisition time in clinical MRI by data undersampling and compressed sensing reconstruction," *Physics in Medicine & Biology*, vol. 60, no. 21, p. R297, 2015.

[17] R. M. Stassart, W. Möbius, K.-A. Nave, and J. M. Edgar, "The Axon-Myelin Unit in Development and Degenerative Disease," *Frontiers in Neuroscience*, vol. 12, 2018.

[18] P. Hasgall, F. Di Gennaro, C. Baumgartner, E. Neufeld, B. Lloyd, M. Gosselin, D. Payne, A. Klingenböck, and N. Kuster, "IT'IS Database for thermal and electromagnetic parameters of biological tissues," https://itis.swiss/virtual-population/tissue-properties/, Version 4.0, May 15, 2018.

[19] M. Warntjes, M. Engström, A. Tisell, and P. Lundberg, "Modeling the Presence of Myelin and Edema in the Brain Based on Multi-Parametric Quantitative MRI," *Frontiers in Neurology*, vol. 7, 2016.

[20] E. Alonso-Ortiz, I. R. Levesque, and G. B. Pike, "MRI-based myelin water imaging: A technical review," *Magnetic Resonance in Medicine*, vol. 73, no. 1, pp. 70–81, Jan. 2015.

[21] P. A. Bottomley, C. J. Hardy, R. E. Argersinger, and G. Allen-Moore, "A review of 1 H nuclear magnetic resonance relaxation in pathology: Are T1 and T2 diagnostic? : Review Article: Proton NMR relaxation in pathology," *Medical Physics*, vol. 14, no. 1, pp. 1–37, Jan. 1987.

[22] P. Schmitt, M. A. Griswold, P. M. Jakob, M. Kotas, V. Gulani, M. Flentje, and A. Haase, "Inversion recovery TrueFISP: Quantification of T1,T2, and spin density," *Magnetic Resonance in Medicine*, vol. 51, no. 4, pp. 661–667, Apr. 2004.

[23] K. P. Whittall, A. L. MacKay, and D. K. Li, "Are mono-exponential fits to a few echoes sufficient to determine T2 relaxation for in vivo human brain?" *Magnetic Resonance in Medicine*, vol. 41, no. 6, pp. 1255–1257, Jun. 1999.

[24] R. M. Kroeker and R. Mark Henkelman, "Analysis of biological NMR relaxation data with continuous distributions of relaxation times," *Journal of Magnetic Resonance (1969)*, vol. 69, no. 2, pp. 218–235, Sep. 1986.

[25] A. A. Istratov and O. F. Vyvenko, "Exponential analysis in physical phenomena," *Review of Scientific Instruments*, vol. 70, no. 2, pp. 1233–1257, Feb. 1999.

[26] S. J. Graham, P. L. Stanchev, and M. J. Bronskill, "Criteria for analysis of multicomponent tissue T2 relaxation data," *Magnetic Resonance in Medicine*, vol. 35, no. 3, pp. 370–378, Mar. 1996.

[27] D. Ma, V. Gulani, N. Seiberlich, K. Liu, J. L. Sunshine, J. L. Duerk, and M. A. Griswold, "Magnetic resonance fingerprinting," *Nature*, vol. 495, no. 7440, pp. 187–192, Mar. 2013.

[28] M. Doneva, P. Börnert, H. Eggers, C. Stehning, J. Sénégas, and A. Mertins, "Compressed sensing reconstruction for magnetic resonance parameter mapping," *Magnetic Resonance in Medicine*, vol. 64, no. 4, pp. 1114–1120, Oct. 2010.

[29] M. Doneva, T. Amthor, P. Koken, K. Sommer, and P. Börnert, "Matrix completion-based reconstruction for undersampled magnetic resonance fingerprinting data," *Magnetic Resonance Imaging*, vol. 41, pp. 41–52, Sep. 2017.

[30] A. Panda, B. B. Mehta, S. Coppo, Y. Jiang, D. Ma, N. Seiberlich, M. A. Griswold, and V. Gulani, "Magnetic resonance fingerprinting – An overview," *Current Opinion in Biomedical Engineering*, vol. 3, pp. 56–66, Sep. 2017.

[31] D. F. McGivney, E. Pierre, D. Ma, Y. Jiang, H. Saybasili, V. Gulani, and M. A. Griswold, "SVD Compression for Magnetic Resonance Fingerprinting in the Time Domain," *IEEE Transactions on Medical Imaging*, vol. 33, no. 12, pp. 2311–2322, Dec. 2014.

[32] A. Oppelt, R. Graumann, H. Barfuss, H. Fischer, W. Hartl, and W. Schajor, "FISP, a novel, fast pulse sequence for nuclear magnetic resonance imaging," *Electromedica*, vol. 54, no. 1, pp. 15–18, 1986.

[33] E. M. Haacke, P. A. Wielopolski, J. A. Tkach, and M. T. Modic, "Steady-state free precession imaging in the presence of motion: Application for improved visualization of the cerebrospinal fluid." *Radiology*, vol. 175, no. 2, pp. 545–552, May 1990.

[34] J. Pfister, M. Blaimer, P. M. Jakob, and F. A. Breuer, "Simultaneous T1/T2 measurements in combination with PCA-SENSE reconstruction (T1* shuffling) and multicomponent analysis," in *Proc. Intl. Soc. Mag. Reson. Med. 25*, 2017, p. 0452.

[35] M. Nagtegaal, T. Amthor, P. Koken, and M. Doneva, "The need of a varying flip angle in multi-component analysis with IR-bSSFP sequences." submitted to Proc. Intl. Soc. Mag. Reson. Med. 27.

[36] B. K. Natarajan, "Sparse Approximate Solutions to Linear Systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, Apr. 1995.

[37] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[38] T. A. Bjarnason, C. R. McCreary, J. F. Dunn, and J. R. Mitchell, "Quantitative t_2 analysis: The effects of noise, regularization, and multivoxel approaches," *Magnetic Resonance in Medicine*, pp. NA–NA, 2009.

[39] S. Foucart and D. Koslicki, "Sparse Recovery by Means of Nonnegative Least Squares," *IEEE Signal Processing Letters*, vol. 21, no. 4, pp. 498–502, Apr. 2014.

[40] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, "Distributed Compressed Sensing of Jointly Sparse Signals," in *Proceedings of the 2005 Asilomar Conference on Signals, Systems, and Computers.* IEEE, 2005, pp. 1537–1541.

[41] S. Cotter, B. Rao, Kjersti Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.

[42] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, Mar. 2006.

[43] J. A. Tropp, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, Mar. 2006.

[44] J. D. Blanchard, M. Cermak, D. Hanle, and Y. Jing, "Greedy Algorithms for Joint Sparse Recovery," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1694–1704, Apr. 2014.

[45] J. Oh, E. T. Han, D. Pelletier, and S. J. Nelson, "Measurement of in vivo multi-component T2 relaxation times for brain tissue using multi-slice T2 prep at 1.5 and 3 T," *Magnetic Resonance Imaging*, vol. 24, pp. 33–43, Jan. 2006.

[46] I. R. Levesque, P. S. Giacomini, S. Narayanan, L. T. Ribeiro, J. G. Sled, D. L. Arnold, and G. B. Pike, "Quantitative magnetization transfer and myelin water imaging of the evolution of acute multiple sclerosis lesions," *Magnetic Resonance in Medicine*, vol. 63, pp. 633–640, Mar. 2010.

[47] R. J. Brown, "Information available and unavailable from multiexponential relaxation data," *Journal of Magnetic Resonance (1969)*, vol. 82, no. 3, pp. 539–561, 1989.

[48] D. Kim, J. L. Wisnowski, C. T. Nguyen, and J. P. Haldar, "Relaxation-Relaxation Correlation Spectroscopic Imaging (RR-CSI): Leveraging the Blessings of Dimensionality to Map In Vivo Microstructure," *arXiv:1806.05752 [eess]*, Jun. 2018.

[49] M. D. Does, "Inferring brain tissue composition and microstructure via MR relaxometry," *NeuroImage*, Dec. 2017.

[50] J. Hennig, "Multiecho imaging sequences with low refocusing flip angles," *Journal of Magnetic Resonance (1969)*, vol. 78, no. 3, pp. 397–407, Jul. 1988.

[51] D. Kumar, H. Hariharan, T. D. Faizy, P. Borchert, S. Siemonsen, J. Fiehler, R. Reddy, and J. Sedlacik, "Using 3D spatial correlations to improve the noise robustness of multi component analysis of 3D multi echo quantitative T2 relaxometry data," *NeuroImage*, vol. 178, pp. 583–601, Sep. 2018.

[52] D. Kumar, T. D. Nguyen, S. A. Gauthier, and A. Raj, "Bayesian algorithm using spatial priors for multiexponential T2 relaxometry from multiecho spin echo MRI," *Magnetic Resonance in Medicine*, vol. 68, no. 5, pp. 1536–1543, 2012.

[53] R. Duarte, A. Repetti, P. A. Gómez, M. Davies, and Y. Wiaux, "Greedy Approximate Projection for Magnetic Resonance Fingerprinting with Partial Volumes," *arXiv:1807.06912 [eess]*, Jul. 2018.

[54] K. Steiglitz and L. McBride, "A technique for the identification of linear systems," *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 461–464, Oct. 1965.

[55] P. Stoica and T. Soderstrom, "The Steiglitz-McBride identification algorithm revisited–Convergence analysis and accuracy aspects," *IEEE Transactions on Automatic Control*, vol. 26, no. 3, pp. 712–717, Jun. 1981.

[56] P. Stoica and P. Babu, "Parameter estimation of exponential signals: A system identification approach," *Digital Signal Processing*, vol. 23, no. 5, pp. 1565–1577, Sep. 2013.

[57] M. Björk, "Contributions to Signal Processing for MRI," Ph.D. dissertation, Acta Universitatis Upsaliensis, Uppsala, 2015, oCLC: 942465443.

[58] S. C. L. Deoni, B. K. Rutt, T. Arun, C. Pierpaoli, and D. K. Jones, "Gleaning multicomponent T1 and T2 information from steady-state imaging data," *Magnetic Resonance in Medicine*, vol. 60, no. 6, pp. 1372–1387, Dec. 2008.

[59] J. Zhang, S. H. Kolind, C. Laule, and A. L. MacKay, "Comparison of myelin water fraction from multiecho T2 decay curve and steady-state methods," *Magnetic Resonance in Medicine*, vol. 73, no. 1, pp. 223–232, Jan. 2015.

[60] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. SIAM, 1974.

[61] A. M. Bruckstein, M. Elad, and M. Zibulevsky, "Sparse non-negative solution of a linear system of equations is unique," in *2008 3rd International Symposium on Communications, Control and Signal Processing*, Mar. 2008, pp. 762–767.

[62] T. T. Nguyen, C. Soussen, J. Idier, and E.-H. Djermoune, "An optimized version of non-negative OMP," in *XXVIe Colloque GRETSI Traitement Du Signal Des Images*, Juan-les-Pins, France, 2017, p. 5.

[63] D. Needell, J. Tropp, and R. Vershynin, "Greedy signal recovery review," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*, Oct. 2008, pp. 1048–1050.

[64] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 317–329, Apr. 2010.

[65] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics*, vol. 11, no. 5, pp. 393–401, Sep. 1997.

[66] R. Peharz and F. Pernkopf, "Sparse nonnegative matrix factorization with l0-constraints," *Neurocomputing*, vol. 80, pp. 38–46, Mar. 2012.

[67] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. London: Academic Press Inc, 1981.

[68] Y. Itoh, M. F. Duarte, and M. Parente, "Perfect Recovery Conditions For Non-Negative Sparse Modeling," *IEEE Transactions on Signal Processing*, vol. 65, no. 1, pp. 69–80, Jan. 2017.

[69] M. H. V. Benthem and M. R. Keenan, "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems," *Journal of Chemometrics*, vol. 18, no. 10, pp. 441–450, 2004.

[70] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, Jan. 2009.

[71] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[72] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, Dec. 2008.

[73] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, Mar. 1997.

[74] J. Nocedal and S. Wright, *Numerical Optimization*, ser. Numerical Optimization. Springer New York, 1999.

[75] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK ; New York: Cambridge University Press, 2004.

[76] E. Candès and J. Romberg, "ℓ_1-magic : Recovery of sparse signals via convex programming," *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*, p. 14, 2005.

[77] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An Interior-Point Method for Large-Scale -Regularized Least Squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[78] M. A. Woodbury, "Inverting Modified Matrices. Statistical Research Group, Memo. Rep. No. 42," *Statistical Research Group*, vol. Memo. Rep. no. 42, p. 4, 1950.

[79] K. Koh, S.-J. Kim, and S. Boyd, "An Interior-Point Method for Large-Scale 1-Regularized Logistic Regression," *Journal of Machine Learning Research*, vol. 8, pp. 1519–1555, 2007.

[80] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, Nov. 1993, pp. 40–44 vol.1.

[81] J. A. Tropp and A. C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 53, pp. 4655–4666, Dec. 2007.

[82] D. Needell and R. Vershynin, "Uniform Uncertainty Principle and Signal Recovery via Regularized Orthogonal Matching Pursuit," *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317–334, Jun. 2009.

[83] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, Feb. 2012.

[84] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, May 2009.

[85] A. Rakotomamonjy, "Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms," *Signal Processing*, vol. 91, no. 7, pp. 1505–1526, Jul. 2011.

[86] Y.-R. Fan, Y. Wang, and T.-Z. Huang, "Enhanced joint sparsity via iterative support detection," *Information Sciences*, vol. 415-416, pp. 298–318, Nov. 2017.

[87] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2008, pp. 3869–3872.

[88] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[89] K. Nehrke and P. Börnert, "DREAM—a novel approach for robust, ultrafast, multislice B1 mapping," *Magnetic Resonance in Medicine*, vol. 68, no. 5, pp. 1517–1526, Nov. 2012.

[90] L. Kjær, C. Thomsen, O. Henriksen, P. Ring, M. Stubgaard, and E. J. Pedersen, "Evaluation of Relaxation Time Measurements by Magnetic Resonance Imaging," *Acta Radiologica*, vol. 28, no. 3, pp. 345–351, Jan. 1987.

[91] W. M. Brink, P. Börnert, K. Nehrke, and A. G. Webb, "Ventricular B1+ perturbation at 7 T – real effect or measurement artifact?" *NMR in Biomedicine*, vol. 27, no. 6, pp. 617–620, Jun. 2014.

[92] "Krabbe Disease," https://www.orpha.net/consor/cgi-bin/OC_Exp.php?Expert=487.

[93] C. Laule, I. M. Vavasour, E. Shahinfard, B. Mädler, J. Zhang, D. K. B. Li, A. L. MacKay, and S. M. Sirrs, "Hematopoietic Stem Cell Transplantation in Late-Onset Krabbe Disease: No Evidence of Worsening Demyelination and Axonal Loss 4 Years Post-allograft: Hematopoietic Stem Cell Transplantation in Late-Onset Krabbe Disease," *Journal of Neuroimaging*, vol. 28, no. 3, pp. 252–255, May 2018.

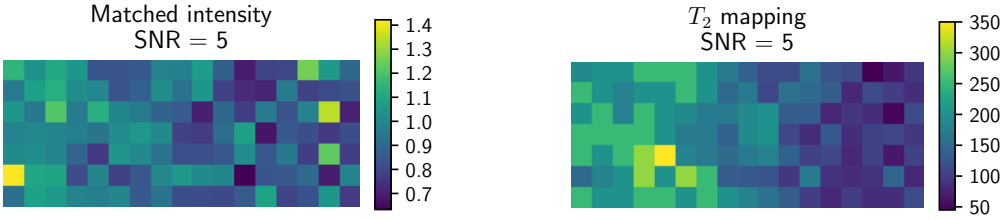
[94] G. L. Bretthorst, W. C. Hutton, J. R. Garbow, and J. J. Ackerman, "Exponential parameter estimation (in NMR) using Bayesian probability theory," *Concepts in Magnetic Resonance Part A*, vol. 27A, no. 2, pp. 55–63, Nov. 2005.

[95] M. Nagtegaal, B. Mädler, T. Amthor, P. Koken, and M. Doneva, "A fast, joint sparsity constrained algorithm for improved myelin water fraction mapping," submitted to Proc. Intl. Soc. Mag. Reson. Med. 27.

[96] M. Nagtegaal, T. Amthor, P. Koken, and M. Doneva, "MR Fingerprinting multi-component analysis using a fast joint sparsity algorithm," submitted to Proc. Intl. Soc. Mag. Reson. Med. 27.

A. Appendix

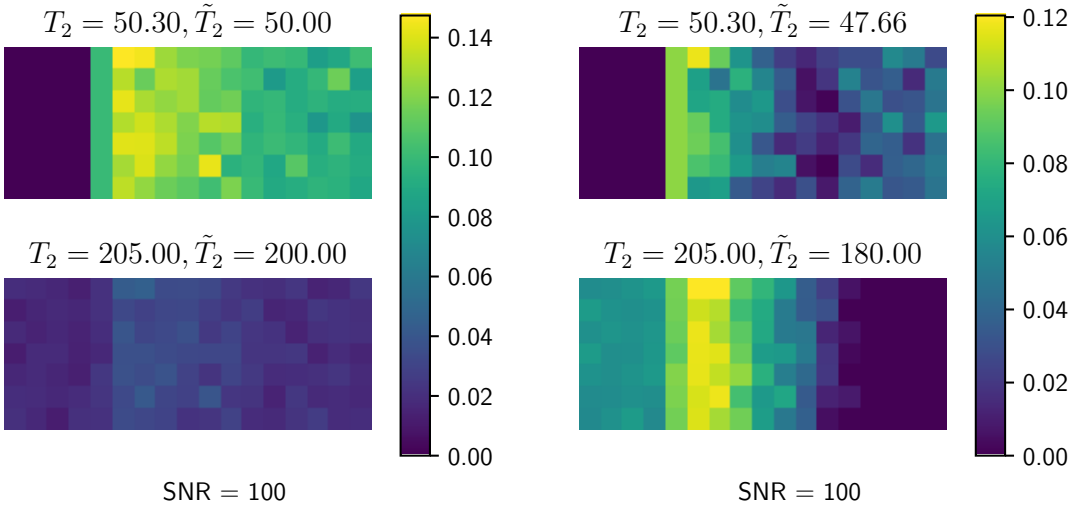
A.1. Simulated images results T_2



(a) The matched scales

(b) The matched T_2 values.

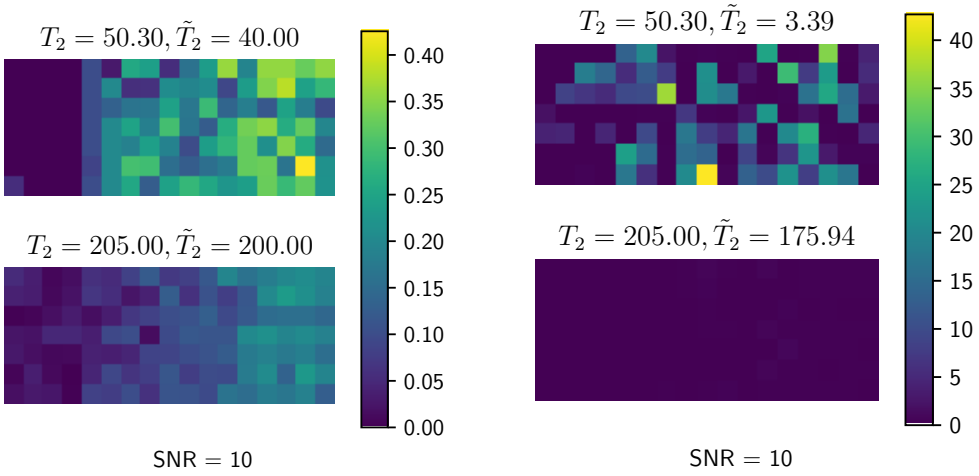
Figure A.1.: The result of a single component T_2 exponential matching with a signal to noise ratio of 5.



(a) Using the SPIJN algorithm the reweighting scheme from Equation 3.27 with $\lambda = 0.25$.

(b) Using the RNNOMP algorithm with two segments.

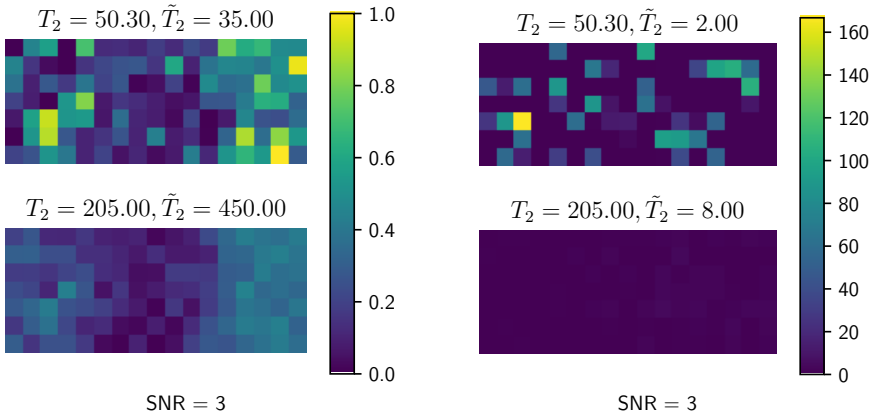
Figure A.2.: The multi-component matching for the simulated image using the T_2 exponential sequence with SNR=100. The true T_2 values and found T_2 values ($\tilde{T}_2$) are given.



(a) Using the SPIJN algorithm the reweighting scheme from Equation 3.27 with $\lambda = 0.25$.

(b) Using the RNNOMP algorithm with two segments.

Figure A.3.: The multi-component matching for the simulated image using the T_2 exponential sequence with SNR=10. The true T_2 values and found T_2 values ($\tilde{T}_2$) are given.



(a) Using the SPIJN algorithm the reweighting scheme from Equation 3.27 with $\lambda = 0.25$.

(b) Using the RNNOMP algorithm with two segments.

Figure A.4.: The multi-component matching for the simulated image using the T_2 exponential sequence with SNR=3. The true T_2 values and found T_2 values ($\tilde{T}_2$) are given.