Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

Christoffer Bro Sørensen

Applications of Deep Neural Networks in Pulse Design with Parallel Transmission for Ultra-High Field MRI

Master's thesis in Applied Physics
Supervisor: Dr. Desmond Ho Yan Tse
June 2020

NTNU
Norwegian University of
Science and Technology

Christoffer Bro Sørensen

Applications of Deep Neural Networks in Pulse Design with Parallel Transmission for Ultra-High Field MRI

Master's thesis in Applied Physics
Supervisor: Dr. Desmond Ho Yan Tse
June 2020

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

NTNU
Norwegian University of
Science and Technology

Acknowledgements

I would like to express my sincere gratitude to my main supervisor Desmond, and co-supervisor Pål Erik, for the guidance and support I have received during my thesis. I feel very fortunate to have had supervisors who always expressed interest and curiosity towards my ideas, and who always encouraged me to explore the questions I had no answer to. I wish to give an extra thanks to Desmond for the extensive help I received during the research phase of my thesis, and for providing the code which served as a backbone for this thesis.

I would also like to thank my family for all the love and emotional support I have received during the course of my degree, leading up to this point. I could not have done this without your help.

Lastly, I wish to thank my dear friend and dance partner Desirée, for always being there whenever I needed someone to talk to. The importance of your support and friendship can not be overstated.

Abstract

Two main objectives are investigated in this thesis, both of which consist of predicting the complex parallel transmission (PTx) weights for an 8-channel transmit (Tx) 32-channel receive (Rx) Nova head coil on a Siemenes Magneton 7T by deep neural networks (DNNs). The main results are based on anonymized data (B_1^+- and B_0-maps) from 17 different scan sessions, which are discerned on a volunteer-to-volunteer basis prior to being applied in pulse designs. The proposed matching consisted of matching reconstructed magnitude 3D images from the respective scans with the Pearson Correlation Coefficient (PCC). The method yields a clean volunteer separation, but is deemed sensitive to artifacts from pre-processing steps and the choice of masking- and PCC-thresholds. Fundamental MRI-, PTx- and Deep Learning theory is also thoroughly reviewed in this thesis.

For the first main objective, two separate multi-layer preceptron (MLP) neural networks (NNs) are trained, validated and tested for the prediction of 8 universal (i.e. subject-independent) PTx-weights for RF-shimming for general volunteer applications. The predictions' shim configurations are aimed at maximizing the concentration of RF-amplitude over a small (approximately) $2 \times 2 \times 2$ cm^3 cube shifted around to user-defined locations in the brain for use in e.g. Single-Voxel Spectroscopy (SVS), while minimizing the estimated maximum and head-averaged local $\mathrm{SAR}_{10\mathrm{g}}$. The results indicate that a network trained with data for which the network learns the desired universal pulse (UP) settings *during* network training performs better on average than for one which the universal pulse settings are already pre-calculated and contained within its training set. The networks' performance is compared to that of pre-calculated universal shims and volunteer-tailored shims, which the two networks respectively manage to sufficiently mimic. The PTx default-drive shim (CP-mode) and a phase-only shim (weighted CP-mode) are also computed for comparison. Weighted CP-mode is tailored to yield constructive phase-interference of each transmit channel's (complex) sensitivity at the cubes' center voxel. The network-predicted pulses, pre-calculated UPs and tailored pulses are all outperformed by the weighted CP-mode. For further work, a method is proposed to train a network in a similar fashion to those presented here, but with weighted CP-mode shims (phase-only) instead of full shims (amplitude and phase).

Secondly, a convolutional neural network (CNN) is trained, validated and tested with sparse amounts of examples (13, 2 and 2 examples in the training-, validation- and test sets, respectively) for the prediction of time-varying PTx-weights of an 8-k_{T}-point trajectory for whole-brain flip-angle (FA) homogenization for general volunteer applications, with the goal of maximizing the FA homogeneity (measured by the coefficient of variance (CoV) of the FAs) over the brain, while minimizing the estimated maximum and head-averaged local $\mathrm{SAR}_{10\mathrm{g}}$. The prediction from the CNN is based solely on the resulting RF-amplitude map from PTx default-drive (CP-mode). For performance comparison, a UP and volunteer-tailored pulses are computed. The CNN-predicted pulse settings share approximately equal SAR-levels (maximum and head-average $\mathrm{SAR}_{10\mathrm{g}}$) as its tailored counterparts, but with approximately equal FA-inhomogeneity as the UP. The CNN-approach presented here should be further investigated to include more MRI data (e.g. relative RF phase data and off-resonances) in its input to improve its predictions.

As all main results presented here rely on the discernment process yielding *true* volunteer discernments, they are all only indicative. The

two main objectives of this thesis should be applied to data which is guaranteed to originate from different volunteers. The discernment process itself should also be verified by application on a set of volunteer data for which the true discernment is already known. All results presented here should also be validated over larger sets of volunteer data.

Keywords: MRI, UHF, PTx, RF, B1+, Shimming, Flip-Angle, Homogenization, Brain, Pulse Design, Deep Learning, Neural Network, CNN, MLP

Contents

1 Introduction

MRI (Magnetic Resonance Imaging) systems, based on the principle of nuclear magnetic resonance (NMR), are often characterized by their strong, static magnetic field strength, denoted B_0. With MRI systems reaching ultrahigh magnetic field (UHF), i.e. $B_0 > 3$T, the advantages are numerous – most importantly, signal-to-noise ratio (SNR) is increased, hence allowing higher image resolution and shorter scan times[1]. There are, however, a set of technical hurdles which must be overcome before these systems can be widely adopted. For instance, achieving control of the flip-angle (FA) across a region-of-interest (ROI), is not a trivial matter at UHFs, where the wavelength of the transmitted radio-frequency (RF) electromagnetic waves in the human body can be of the same order as the spatial dimensions of the body-part being imaged (e.g. the head). The interference between travelling waves within the object from individual transmission coil elements gives rise to standing waves patterns[2] in the magnetic field associated with the waves, causing spatial fluctuations in the amplitude of the NMR-active component of the transmitted RF-field. This fluctuation can in addition be caused (or enhanced) by true dielectric resonance effects[3], i.e. spatial fluctuations in the amplitude depending on the overlap between the transmitted frequency and the dielectric resonance frequencies of the object. Regardless of its source, this problem manifest itself in UHF MR images as either regions of complete signal voids or contrast shading across the image.

Traditionally, at lower field strengths, single-channel transmission coils have been used to transmit RF pulses, regardless of the type of pulses (e.g. selective, non-selective, one-dimensional and multi-dimensional). At UHFs, multi-channel transmission coils are essential tools to achieve the desired control of the transmit field. The framework which describes the simultaneous, independent pulsing of more than one channel is called Parallel Transmission (PTx), and is the foundation which makes the desired control of the transmit-field achievable. In this regard, *the primary aim of this thesis is to*

- *revisit this author's project thesis work*[4] *of RF-shimming with fully-connected multi-layer perceptron (MLP) networks, to improve and verify results with networks more adapted to generalize beyond a single transmit sensitivity map by predicting universal PTx-settings to move the concentration of RF-amplitude to a desired location in the brain for general volunteer application, and compare the results to tailored pulse settings and universal pulse (UP) settings.*

- *investigate the feasibility of training a convolutional neural network (CNN) with a sparse amount of training data for the prediction of time-varying PTx-weights of an 8-k_T-point excitation trajectory for general volunteer whole-brain FA homogenization, and compare the results to tailored pulse settings and UP settings.*

The MRI data used in this thesis consists B_1^+-sensitivity maps gathered at 7T for an 8-channel transmit (Tx) 32-channel receive (Rx) Nova head coil for 17 scans of volunteers, along with the scans' respective B_0-map. The data is completely anonymized (and may contain several scan of the same volunteer), and consequently needs to be grouped within volunteers before it can be applied

to the creation of training, validation and test data. *Thus, a secondary aim of this thesis is to shortly present and apply a simple method for within-volunteer grouping of anonymized data based upon intra-modality pixel-by-pixel comparison of full-head magnitude images.*

All MATLAB code written for this thesis is available at `https://github.com/chrisbso/MastersThesis`.

The information regarding the Q-matrices used for estimation of SAR-levels from the Nova head coil in this thesis is found in section 7.5 of the appendix.

2 Background and Theory

2.1 Fundamentals

2.1.1 Transmission in MRI

Transmission in MRI is the action of applying another smaller magnetic field $\mathbf{B}_{\text{transmit}}(\mathbf{r}, t)$ in a direction[1] perpendicular to the main, static magnetic field, $\mathbf{B_0} = -B_0\hat{\mathbf{z}}$. This nutates ("tips") the macroscopic net magnetization vector, $\mathbf{M}$, out of equilibrium ($\mathbf{M} = M_0\hat{\mathbf{z}}$), such that it starts precessing and induces a time-varying voltage (i.e a signal) in receiving coils. This field is commonly applied by transmitting an electromagnetic wave through a set of RF-coils, whose resultant magnetic field's sole purpose is to disturb the aforementioned equilibrium. For a single transmission coil, the associated magnetic field in the laboratory frame, $\mathbf{B_1}(\mathbf{r}, t)$, is a linearly polarized field with carrier frequency ω_{RF} and vector-amplitude $\mathbf{B_{1\,\text{amp}}}(\mathbf{r}, t)$, s.t.

$$\mathbf{B_1} \equiv \mathbf{B_1}(\mathbf{r}, t) \equiv \mathbf{B_{1\,\text{amp}}}(\mathbf{r}, t)\cos\left(\omega_{\text{RF}}t + \phi_{\text{coil}}\right),$$

where ϕ_{coil} is a phase constant. We further decompose its vector-amplitude into components along the $\hat{\mathbf{x}}$- and $\hat{\mathbf{y}}$-directions, s.t.

$$\mathbf{B_{1\,\text{amp}}}(\mathbf{r}, t) \equiv B_{1x}\hat{\mathbf{x}} + B_{1y}\hat{\mathbf{y}} \equiv B_{1x}(\mathbf{r}, t)\hat{\mathbf{x}} + B_{1y}(\mathbf{r}, t)\hat{\mathbf{y}}.$$

In order to get rid of the time dependence due to the oscillation, we first introduce a rotating frame, rotating counter-clockwise with frequency $\omega > 0$ about the $\hat{\mathbf{z}}$-axis, having unit vectors

$$\hat{\mathbf{x}}' = \hat{\mathbf{x}}\cos(\omega t) + \hat{\mathbf{y}}\sin(\omega t), \quad \hat{\mathbf{y}}' = -\hat{\mathbf{x}}\sin(\omega t) + \hat{\mathbf{y}}\cos(\omega t), \quad \hat{\mathbf{z}}' = \hat{\mathbf{z}}. \tag{1}$$

Now, we assume that we "lock" our rotating frame to $\mathbf{B_1}$, such that in the given frame, $\mathbf{B_1}$ constitutes a non-rotating[2] field. This is equivalent to setting $\omega \equiv \omega_{\text{RF}}$. Furthermore, we match the frequency of our $\mathbf{B_1}$-field to the Lamour frequency $\omega_0 \equiv \gamma B_0$ to meet the *resonance condition*,

$$\omega = \omega_0 \quad \text{(on-resonance)}. \tag{2}$$

This is the most effective condition to nutate spins. A macroscopic interpretation for this is that in the rotating frame, the $\mathbf{B_1}$-field is synchronized perfectly with the precession[3], such that $\mathbf{B_0}$ vanishes from the effective magnetic field experienced by the spins. Furthermore, any effects which may arise from (2) not being satisfied are known as off-resonance effects, and will become important in the later discussions.

Using phasor-notation[4] we introduce the NMR-active (i.e. contributing to spin nutation) part of $\mathbf{B_1}$ as the *transmit* B_1^+*-field*, defined in the rotating frame in terms of its vector-amplitude components *in the laboratory frame*[5]:

$$B_1^+(\mathbf{r}, t) \equiv \frac{1}{2}\left[B_{1x}(\mathbf{r}, t) + iB_{1y}(\mathbf{r}, t)\right], \tag{3}$$

[1] Meaning that its non-zero components are perpendicular to $\hat{\mathbf{z}}$.

[2] Meaning only its amplitude may be timely dependent on the carrier frequency.

[3] An analogy to this is pushing someone on a swing - pushing out of sync with the swing breaks its speed, while pushing in sync maintains or increases its speed.

[4] By phasor-notation, we mean assigning the $\hat{\mathbf{x}}'$-component to the real part of a complex number, and the $\hat{\mathbf{y}}'$-component to the number's imaginary part.

The factor of one-half in the above equation arises due to a decomposition of the linearly polarized field as the superposition of two counter-rotating fields, where only the terms following the primed frame is considered pertinent to spin nutation. This can easily be derived[6] by inserting definitions of eq. (1) into the definition of the $\mathbf{B_1}$-field, and truncating terms which are sinusoidal in 2ω (far off resonance) after trigonometric simplifications.

Take special note that the spatial and timely dependence in eq. (3) is *not* due to the oscillatory nature of the $\mathbf{B_1}$-field in the laboratory frame, but rather to emphasise that it may vary spatiotemporally – *in the ideal case, $B_1^+(\mathbf{r},t)$ is spatially constant for any fixed point in time, i.e. $B_1^+(\mathbf{r},t)$ is a homogeneous field*. However, in the presence of wave interference effects (prominent at UHF), this is no longer generally true.

We have not made any restriction on the RF coil configuration for multiple coils for eq. (3). The only assumption needed for the above discussion to be generalized to multiple RF coils transmitting in parallel with their respective relative carrier phase (ϕ_{coil}) and vector-amplitude ($\mathbf{B_1}_{\text{amp}}(\mathbf{r},t)$), is that they all share the same carrier frequency ω_{RF}. Furthermore, in practice, one considers a *timely average* of eq. (3) when estimating each coil's contribution to the resulting B_1^+-field – this is due to the inhomogeneity problem mentioned in the previous paragraph, and that the total transmitted field may not constitute a field which is circularly polarized (CP), i.e. a field in which the modulus of (3) is constant in time. *The time averaged B_1^+-field is the field which yields the same spin nutation as a CP-field over the time averaged* (this will be introduced as a coil's *sensitivity* in section 2.2). CP driving schemes requires that all coils transmit with the same-sized amplitude, and with amplitude direction and timely phase offset coinciding with their relative spatial offset. As an example, for two coils transmitting in quadrature with a 90° spatial offset at resonance and equal amplitude-size $B_1(\mathbf{r},t)$ (i.e. a two-coil CP driving scheme), their respective fields and their total combined fields, with superscripts $1,2$ identifying coil 1 and coil 2, can be described as

$$\begin{aligned}
\text{Coil 1: } \mathbf{B_1^1} &= B_1(\mathbf{r},t)\cos(\omega t)\,\hat{\mathbf{x}},\\
\text{Coil 2: } \mathbf{B_1^2} &= B_1(\mathbf{r},t)\cos\left(\omega t-\frac{\pi}{2}\right)\hat{\mathbf{y}},\\
\text{Sum: } \mathbf{B_1} &= B_1(\mathbf{r},t)\cos(\omega t)\,\hat{\mathbf{x}}+B_1(\mathbf{r},t)\sin(\omega t)\,\hat{\mathbf{y}}\\
&= B_1(\mathbf{r},t)\hat{\mathbf{x}}'\\
\implies B_1^+ &= B_1(\mathbf{r},t).
\end{aligned}$$

As a last remark, if $\mathbf{B_0}$ was oriented along the positive $\hat{\mathbf{z}}$-direction, the transmit field would be the complex conjugate of eq. (3), and a natural first instinct would be to instead denote the field as B_1^-, since the corresponding field of transmission would be its complex conjugate. However, in relevant literature, B_1^- is reserved to mean the *receive field*, and B_1^+ reserved to mean the *transmit field*, regardless of the static field orientation.

2.1.2 Small Tip Angle (STA) Approximation

The timely evolution of the macroscopic magnetization $\mathbf{M}' \equiv \mathbf{M}'(\mathbf{r},t)$ in the rotating frame defined by eq. (1) is governed by the *Bloch Equations*. We will now concern ourselves with the magnetization during transmission – defining

$\mathbf{M'} \equiv M_{x\prime}\hat{\mathbf{x}}' + M_{y'}\hat{\mathbf{y}}' + M_z\hat{\mathbf{z}}$ (all three components spatiotemporally dependent) and suppressing any dependencies in the transmitted field, the Bloch equations take the matrix form[7][8]

$$\frac{\mathrm{d}\mathbf{M'}}{\mathrm{d}t} \equiv \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} M_{x\prime} \\ M_{y'} \\ M_z \end{bmatrix} = \gamma \begin{bmatrix} 0 & \mathbf{G}\cdot\mathbf{r} & -B_{1y} \\ -\mathbf{G}\cdot\mathbf{r} & 0 & B_{1x} \\ B_{1y} & -B_{1x} & 0 \end{bmatrix} \begin{bmatrix} M_{x\prime} \\ M_{y'} \\ M_z \end{bmatrix}, \tag{4}$$

where $\mathbf{G}\cdot\mathbf{r}$ is the additional field along in the static field direction produced by the gradient $\mathbf{G} \equiv \mathbf{G}(t)$ at position $\mathbf{r}$ relative to the iso-center in the laboratory frame, and $\gamma/2\pi \approx 42.58\mathrm{MHz\,T^{-1}}$ is the gyromagnetic ratio of ^{1}H[9, p. 26]. We have neglected relaxation effects in (4) as we assume the duration of the RF-pulse is much shorter than the relaxation times of the object subject to the pulse.

We now make the small-tip-angle (STA) approximation to (4), where we assume the longitudinal component M_z of the magnetization remains approximately constant and equal to its equilibrium value during RF-pulsing, as we assume the magnetization vector is tipped only a small angle $\theta \equiv \measuredangle(\hat{\mathbf{z}}, \mathbf{M'})$, i.e.

$$\begin{aligned} M_z &\equiv M_0\cos\theta \approx M_0, \\ M_{x'y'} &\equiv M_0\sin\theta \approx M_0\theta. \end{aligned} \tag{5}$$

Here we have introduced the transversal component $M_{x'y'}$ of the magnetization, defined in phasor-notation as

$$M_{x'y'} \equiv M_{x'} + iM_{y'}.$$

Under the STA approximation (5), eq. (4) decouples for the longitudinal and transversal component. We now assume we apply the RF-pulse for time $t \in [0, T_p]$. For initial condition $\mathbf{M'}(\mathbf{r}, t = 0) = M_0\hat{\mathbf{z}}$, the solution[5] for the transversal magnetization at time $t = T_p$ (i.e. at the end of the pulse) is

$$M_{x'y'}(\mathbf{r}, T_p) = i\gamma M_0 \int_0^{T_p} B_1^+(\mathbf{r}, t)e^{i\mathbf{r}\cdot\mathbf{k}(t)}\mathrm{d}t, \tag{6}$$

where we define the *transmit k-space trajectory* as

$$\mathbf{k}(t) \equiv -\gamma \int_t^{T_p} \mathbf{G}(\tau)\mathrm{d}\tau.$$

This trajectory exists in the same k-space which is commonly associated with image encoding, but the trajectory itself is expressed as time-inverted integral compared to its image encoding counterpart. Citing [10], an interpretation of this time-inversion is that "as the RF-pulse is being played out, new transverse magnetisation is being created, which is then subject to all future applied gradients".

When solving for $B_1^+(\mathbf{r}, t)$ in (6) through iterative methods, it is often feasible to include contributions to the phase-term in eq. (6) due to (static) inhomogeneties $\Delta B_0(\mathbf{r})$ in the static field, i.e. $\mathbf{B_0}(\mathbf{r}) = (B_0 + \Delta B_0(\mathbf{r}))\hat{\mathbf{z}}$. These inhomogeneties may arise from technical imperfections in the MRI system and the inability to achieve perfect shimming, susceptibility variations across the

[5] See appendix, section 7.2.1, for details.

imaged object or chemical shifts effects[8]. To incorporate these contributions, one augments the accrued phase in the integral of (6), specifically[8]

$$\exp(i\mathbf{r}\cdot\mathbf{k}(t)) \rightarrow \exp\left(i\mathbf{r}\cdot\mathbf{k}(t) + i\gamma\Delta B_0(\mathbf{r})(t-T_p)\right). \tag{7}$$

The validity of this augmented solution can be verified by setting

$$\mathbf{G}\cdot\mathbf{r} \rightarrow \mathbf{G}\cdot\mathbf{r} + \Delta B_0(\mathbf{r})$$

in eq. (4) and following the same derivation as outlined above.

2.1.3 B_0-mapping

B_0-mapping is the process of estimating the off-resonance contributions

$$\Delta B_0(\mathbf{r}_n)\,\forall\, n,$$

associating each of $n = 1,\dots N_s$, discretized spatial points $\mathbf{r}_n$ with a (non-overlapping) voxel. The mapping can be done by calculating the phase-difference between the two images obtained in a dual-echo (DE) gradient recalled echo (GRE) sequence, each with echo times TE_1 and TE_2, respectively. We here outline its theory[11]: let

$$Z_1 = \mu_1 e^{i\phi_1},$$
$$Z_2 = \mu_2 e^{i\phi_2},$$

be the complex pixel value of the two images associated with the voxel at r_n. The off-resonance map can be calculated as[9]

$$\Delta B_0(\mathbf{r}_n) = \frac{\phi_{\mathrm{diff}}}{\gamma(\mathrm{TE}_1 - \mathrm{TE}_2)}, \tag{8}$$

where ϕ_{diff} is the (unwrapped) phase difference between the two images for voxel at $\mathbf{r}_n$. See figure 1 for a simplified sequence diagram. The phase difference can be calculated by the four-quadrant arctan function ATAN2$[\cdot,\cdot]$,

$$\phi_{\mathrm{diff}} = \mathrm{ATAN2}[\mathrm{Im}(Z_1 Z_2^*), \mathrm{Re}(Z_1 Z_2^*)],$$

but needs to be unwrapped prior to be used in (8). For e.g 3-D dual-echo gradient recalled echo (3DEGRE) sequences, a phase unwrapping method is presented in [12].

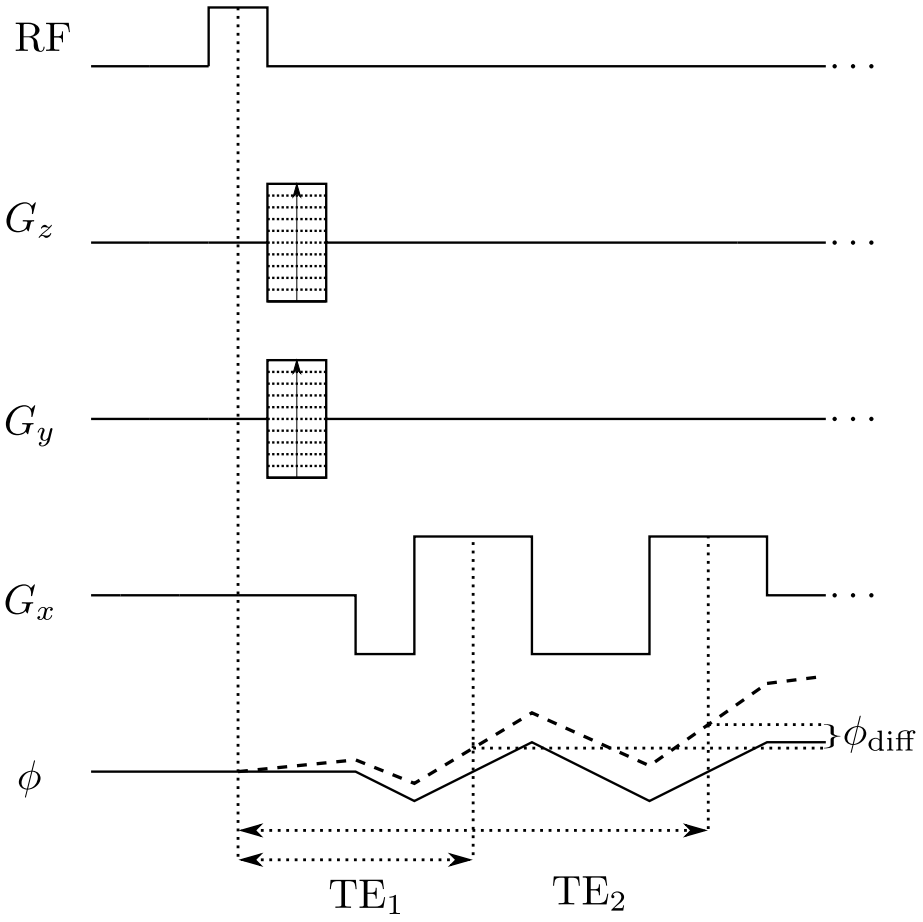


Figure 1: A simplified 3-D dual echo gradient recalled echo (3DEGRE) sequence used for B_0-mapping. The RF pulse is non-selective. ϕ is the phase accumulated from the **x**-gradient G_x and static field inhomogeneties only (we have left out the phase accrued due to the y- and z-gradients G_y and G_z). The dashed and solid lines along indicate the accrued phase with and without static field inhomogeneties, respectively. Ellipses indicate a sufficiently long repetition time before the sequence is repeated for the next Fourier line.

2.1.4 B_0-shimming

After off-resonance contributions have been mapped, the field can be corrected by the use of shimming coils. $\mathbf{B_0}(r)$ must satisfy Laplace's equation (and thus also its $\hat{\mathbf{z}}$-component), implying

$$\nabla^2(\Delta B_0(\mathbf{r})) = 0. \tag{9}$$

Denoting $X_l^m \equiv X_l^m(\mathbf{r})$ as the l^{th} order real solid spherical harmonic function of degree $|m| \leq l$ and C_l^m its corresponding (real) coefficient, the solution to (9) can be written as a linear combination of all real solid spherical harmonic functions[13][14],

$$\Delta B_0(\mathbf{r}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=l} C_l^m X_l^m,$$
$$\text{where } X_l^m \equiv \begin{cases} \cos(m\mu) P_l^m(\cos\nu), & m \geq 0 \\ \sin(-m\mu) P_l^{-m}(\cos\nu), & m < 0 \end{cases}. \tag{10}$$

Here, $P_l^m(\cdot)$ is the l^{th} order associated Legendre polynomial[13][15] of degree m, and (r, ν, μ) the spherical coordinates, related to Cartesian coordinates (x, y, z) by

$$r = \sqrt{x^2 + y^2 + z^2}, \quad \nu = \text{ATAN2}[y, x], \quad \mu = \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right). \tag{11}$$

Denote now $X_l^m(\mathbf{r}_n) \equiv X_{l,n}^m$, i.e X_l^m evaluated at voxel position $\mathbf{r}_n \equiv [x_n, y_n, z_n]$. Suppose we have shimming coils, each able to produce[6] magnetic fields $C_l^m X_l^m(\mathbf{r})\hat{\mathbf{z}}$ of orders $l = 0, \ldots, L$ for all $\mathbf{r} = \mathbf{r}_n$, whose coefficient C_l^m we can freely choose. This coefficient can be interpreted physically as a measure of the current we drive the coil of order and degree l, m with. Let $\mathbf{b}_0 \in \mathbb{R}^n$ be the off-resonance vector whose n^{th} entry is $\Delta B_0(\mathbf{r}_n)$. Define $\mathbf{c} \in \mathbb{R}^{L(L+2)}$ as[7]

$$\mathbf{c} \equiv \begin{bmatrix} C_0^0 & C_1^{-1} & C_1^0 & C_1^1 & \ldots & C_L^{-L} & C_L^{-L+1} & \ldots & C_L^L \end{bmatrix}^T,$$

and the shimming system matrix $X \in \mathbb{R}^{N_s \times L(L+2)}$ as

$$X \equiv \begin{bmatrix} X_{0,1}^0 & X_{1,1}^{-1} & X_{1,1}^0 & X_{1,1}^1 & \ldots & X_{L,1}^{-L} & X_{L,1}^{-L+1} & \ldots & X_{L,1}^L \\ X_{0,2}^0 & X_{1,2}^{-1} & X_{1,2}^0 & X_{1,2}^1 & \ldots & X_{L,2}^{-L} & X_{L,2}^{-L+1} & \ldots & X_{L,2}^L \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{0,N_s}^0 & X_{1,N_s}^{-1} & X_{1,N_s}^0 & X_{1,N_s}^1 & \ldots & X_{L,N_s}^{-L} & X_{L,2}^{-L+1} & \ldots & X_{L,N_s}^L \end{bmatrix}.$$

The entries of X can be calculated at any voxel position by the relations in (11). We calculate the shimming corrections to $\mathbf{b}_0^{\text{corr}}$ up to L^{th} order as

$$\mathbf{b}_0^{\text{corr}} = X\hat{\mathbf{c}}, \quad \text{where } \hat{\mathbf{c}} \equiv \arg\min_{\mathbf{c}} ||X\mathbf{c} - \mathbf{b_0}||_2. \tag{12}$$

Thus, the static field resonance offset after shimming can be replaced as

$$\mathbf{b}_0 \to \mathbf{b}_0 - \mathbf{b}_0^{\text{corr}}.$$

Eq. (12) can be solved by e.g. the conjugate gradient method for least-squares[16]. Of course, any shim coil can be removed from the optimization by removing its corresponding column in X and entry in $\mathbf{c}$.

[6] This is a simplification – due to the nature and shape of the coils, each coil cannot produce a field corresponding to a single spherical harmonic basis function, and a basis-to-coil conversion matrix can be included in the calculation. Here, the conversion matrix is simply the identity matrix.

[7] The dimension of $\mathbf{c}$ is easily calculated by counting all combinations of l, m for $l \leq L$, and making use of the sum of all natural numbers up to L.

2.1.5 B_1^+-mapping

Volumetric B_1^+-mapping can be done by means of the slice-by-slice Dual Refocusing Echo Acquisition Mode (DREAM) sequence[17], in which the B_1^+-field is estimated from a series of slice-stacked flip-angle maps, each calculated from the ratio between the intensities of two 2-D images, made from a free induction decay (FID) signal and a stimulated echo (STE) signal, respectively. Before the imaging sequence, the magnetization is first prepared through a stimulated echo acquisition mode (STEAM) preparation sequence, where to two equal slice-selective RF-pulses (nominal FA of α), separated in time by T_s, are used. A small amount of the prepared magnetization is then repeatedly turned into transverse magnetization by a slice-selective imaging pulse (nominal FA of β), which yields two echos through gradient recalling (with echo times $\mathrm{TE_{FID}}$ and $\mathrm{TE_{STE}}$, respectively). The slice-thickness of the α-pulses is chosen at least twice that of the β-pulse to avoid signal contamination due to slice profile imperfections[18]. See figure 2 for a simplified sequence diagram.

Let the signal intensity images be I_{FID} of the FID and I_{STE} of the STE, respectively. We now assume the imaging k-space is sampled center-to-out. This is to minimize longitudinal relaxation effects and the effect of exhaustion of the prepared magnetization due to the repeated β-pulsing[19]. Under this assumption, the intensity images can be written[20]

$$\begin{aligned} I_{\mathrm{STE}} &= \frac{1}{2}\sin(\beta)\sin^2(\alpha)M_0 \\ I_{\mathrm{FID}} &= \sin(\beta)\cos^2(\alpha)M_0, \end{aligned}$$

which gives the flip-angle map (assuming $0° < \alpha < 90°$)

$$\alpha = \arctan\left(\sqrt{\frac{2I_{\mathrm{STE}}}{I_{\mathrm{FID}}}}\right) \implies \hat{B}_1^+ = \frac{\alpha}{\gamma\, w \int_0^\tau p_\alpha(t)\mathrm{d}t}, \tag{13}$$

where τ is the length of the STEAM preparation pulse, $p_\alpha(t)$ its complex pulse shape in units of µT/V, and $\hat{B}_1^+$ is the estimated average B_1^+-field map. A simplified derivation of the intensity equations are given in section 7.2.4 of the appendix. We have assumed T_s is chosen according to

$$T_s = \mathrm{TE_{FID}} - \mathrm{TE_{STE}}$$

to compensate for transversal relaxation effects due to both spin-spin interactions and inhomogeneities in the B_0-field[21]. This choice gives the desired compensation due to the magnetization of both signals existing in a transversal state for an equal amount of time before read-out.

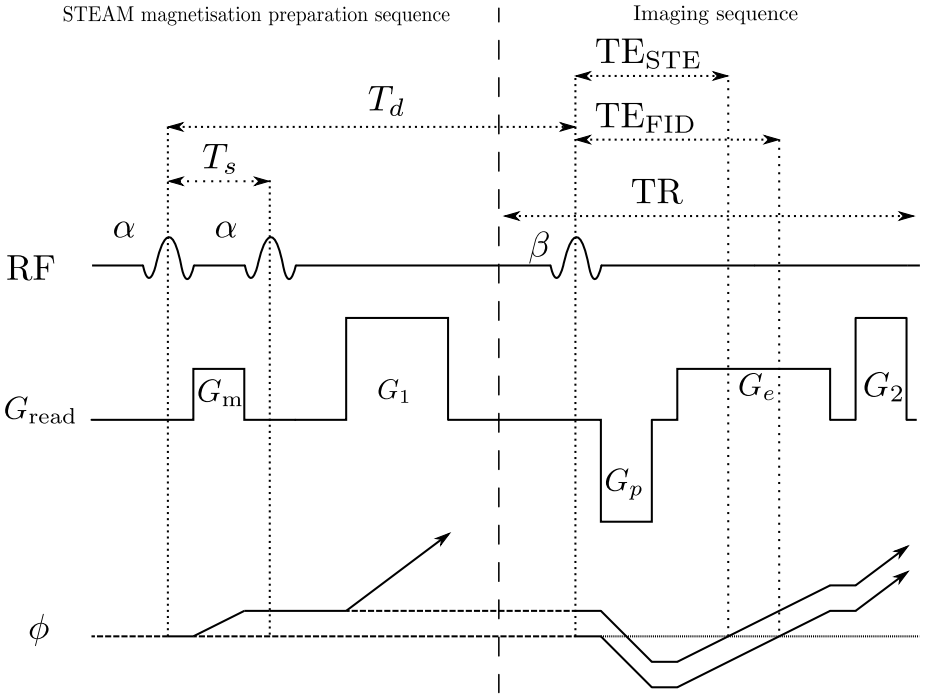


Figure 2: A simplified diagram of the DREAM sequence used for B_1^+-mapping. G_m is the gradient used to separate the FID and STE. The slice-selective preparation and imaging pulses are referred to by their (spatially variant) FAs α and β, respectively. TR is the repetition time for the imaging sequence, T_d is the effective time delay, T_s is the time between the two α-pulses, and $\mathrm{TE_{FID}}$ and $\mathrm{TE_{STE}}$ are the echo times of the FID and STE, respectively. G_p and G_e are used to center the echoes for read-out. G_1 and G_2 are spoiling gradients, whose function is to destroy any spurious signal. The arrows in the accrued phase ϕ indicates spoiling – the longitudinal and transversal magnetizations are indicated by dashed and soild lines, respectively. The echos are formed when the solid lines cross the tightly stapled line. We have left out both the slice-selective and phase encoding gradients in this figure for the sake of simplicity.

2.1.6 Non-Selective Transmit k-space Trajectory: k_{T}-points

We briefly introduce the concept of k_{T}-points[22] (k_{T} being shorthand for *transmission k-space*) – a k-space trajectory which visits low-frequency k-space locations, and remaining stationary at these locations (the k_{T}-points) while RF power is transmitted, see figure 3. The RF-pulse is divided into sub-pulses, with each sub-pulse being played out while stationary at a k_{T} point. The non-selectivity is evident (see eq. (5)) as there is no spatial encoding appearing from the gradients during RF-pulsing, and is therefore a common choice for e.g. whole-brain FA homogenization[23]. The location for $N_{k_{\mathrm{T}}}$ points can be chosen as the k-space locations corresponding to the $N_{k_{\mathrm{T}}}$ largest magnitude components of the 3D Fourier transform of the brain mask[22], with the trajectory traversing the shortest path between adjacent points. Another method is to sample $N_{k_{\mathrm{T}}}$ points corresponding to some of the highest frequency components of the

FOV, e.g. choosing some or all k_T-points with components $\pm 1/(2 \cdot \mathrm{FOV}_d)$ for the field-of-view (FOV) in the $d = \hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ directions in Cartesian coordinates. Lastly, the k_T-points can be chosen similar to the FOV-method, but choosing k_T-points with components in the $d = \hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ directions corresponding (roughly) to the inverse of the wavelength of RF in tissue at the given (or higher) field strength[23].

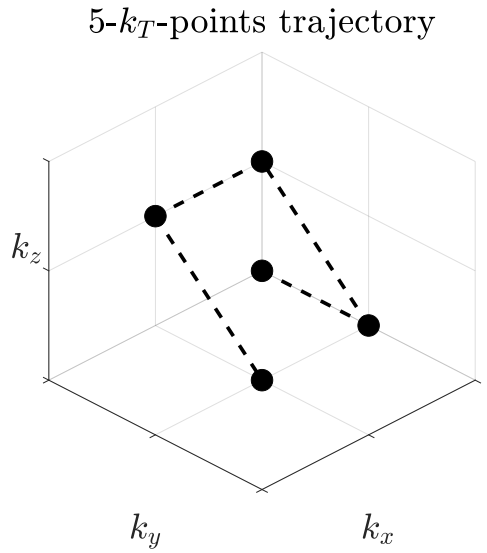


Figure 3: An example 5-k_T-points transmit k-space trajectory for non-selectivity. Here, $B_1^+(\mathbf{r}, t) \equiv 0$ while traversing the stapled lines.

This transmit trajectory is a counter-measure to B_1^+-inhomogeneities: increasing the number of k_T-points gives more control of the resulting magnetization, at the cost of longer pulse duration and sensitivity to off-resonance effects[24].

2.2 Parallel Transmission (PTx)

Parallel Transmission (PTx) is the framework which describes the transmission from multiple RF transmission coils in parallel, each driven by their own, independent RF front-end. This yields full control of each coil's amplitude and relative phase, in which each coil constitutes its own respective *channel*. During pulsing, the resulting B_1^+-field (3) is the superposition of the $B_{1,j}^+$-field for the j^{th} channel, from all N_{C} channels[10],

$$B_1^+(\mathbf{r},t) = \sum_{j=1}^{N_{\text{C}}} B_{1,j}^+(\mathbf{r},t).$$

Each $B_{1,j}^+$ can be further decomposed, as an approximation, into a spatial part, $S_j(\mathbf{r})$, and a temporal part, $p_j(t)$, yielding

$$B_1^+(\mathbf{r},t) \approx \sum_{j=1}^{N_{\text{C}}} S_j(\mathbf{r}) p_j(t). \tag{14}$$

Here, $S_j(\mathbf{r})$ and $p_j(t)$ (both complex) are the *transmit sensitivity* and the *pulsed waveform* of the j^{th} channel, respectively. There are many ways to assign units to eq. (14). Here, we explicitly assign the units

$$[S_j(\mathbf{r})] = \mu\text{T/V} \text{ and } [p_j(t)] = \text{V}.$$

2.2.1 Iterative STA Pulse Design with PTx

For practical reasons regarding computational efficiency, we will from here on assume all channels transmit the same waveform, modulated by a channel-specific constant complex weighting during the RF-pulse. Let

$$p_j(t) \equiv p(t) w_j(t), \quad \text{with } [p(t)] \equiv \text{V s.t. } [w_j(t)] = 1, \tag{15}$$

where $w_j(t)$ is the complex weighting of the j^{th} channel and $p(t)$ is the (unitless) common waveform across all channels. Under the STA approximation, the traverse magnetization created by the RF-pulse transmitted in parallel from N_{C} channels can by (14) thus be written as

$$M_{x'y'}(\mathbf{r},T_p) \approx i\gamma M_0 \sum_{j=1}^{N_{\text{C}}} S_j(\mathbf{r}) \int_0^{T_p} p(t) w_j(t) e^{i\mathbf{r}\cdot\mathbf{k}(t) + i\gamma \Delta B_0(\mathbf{r})(t-T_p)} \mathrm{d}t, \tag{16}$$

now have taking into account the accrued phase described by eq. (7). We will now discretize (16) timely and/or spatially, depending on the application, and present the details regarding iterative pulse designs for each application.

Before any pulse sequences can be used for applied for clinical purposes, the energy deposited in tissue by the associated electric field of the RF-pulses needs to be accounted for to ensure the *specific absorption rate* (SAR) is within regulatory limits. The SAR is a measure of the absorbed RF power averaged either *globally* over the whole body mass, or *locally*, often over 10 grams of tissue. To quantify a relative measure of the global RF power deposited in tissue during

the RF-pulse, $\mathrm{SAR}_{\mathrm{gbl}}$, we will regularize our pulse design on the L^2-norm of the channel weights:

$$\mathrm{SAR}_{\mathrm{gbl}}(t) \propto \sum_{j=1}^{N_\mathrm{C}} w_j^2(t).$$

2.2.2 Static PTx for RF-shimming

Assuming no gradients[8] and constant channel weighting (i.e. $w_j(t) = w_j$) during the RF-pulse, and neglecting inhomogeneity effects in the static field, eq. (16) simplifies to

$$\frac{M_{x'y'}(\mathbf{r}, T_p)}{i\gamma M_0 \int_0^{T_p} p(t)\mathrm{d}t} \approx \sum_{j=1}^{N_\mathrm{C}} S_j(\mathbf{r}) w_j. \tag{17}$$

By comparison with eq. (14), we note that *RF-shimming reduces the pulse design problem to deciding optimal weightings w_j of the superpositioned sensitivites $S_j(\mathbf{r})$*, where "optimal" depends on the target field pattern. The resulting flip-angles are found from rescaling the result after $p(t)$ is set by its integral $\int_0^{T_p} p(t)\mathrm{d}t$. Now, associating each of N_s discretized spatial points $\mathbf{r}_n$, $n = 1, \ldots N_s$, with a non-overlapping voxel within the ROI, and introducting

- $\mathbb{S}$ as the $N_s \times N_\mathrm{C}$ sensitivity matrix whose entry at (n, j) is $S_j(\mathbf{r}_n)$
- $\mathbf{w}$ as the $N_\mathrm{C} \times 1$ vector whose j^{th} entry is w_j
- $\mathbf{b}$ as the $N_s \times 1$ vector whose n^{th} entry is $\hat{B}_1^+(\mathbf{r}_n)$

then eq. (17) can be approximated as a matrix multiplication by

$$\mathbf{b} \approx \mathbb{S}\mathbf{w}.$$

If we restrict ourselves to optimizing the modulus of (17) across the ROI, then for a predefined, desired target field pattern $\mathbf{b_{tar}}$, the matrix inversion problem for the estimation $\hat{\mathbf{w}}$ of $\mathbf{w}$ can be cast as a *regularized magnitude least-squares* problem, i.e.

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left(|| \, |\mathbb{S}\mathbf{w}| - \mathbf{b_{tar}} \, ||_2^2 + \lambda ||\mathbf{w}||_2^2 \right). \tag{18}$$

where $\lambda||\mathbf{w}||_2^2$ is the regularization cost term penalizing on $\mathrm{SAR}_{\mathrm{gbl}}$ across the imaged volume, parameterized by the *Tikhonov regularization factor* $\lambda \geq 0$. Eq. (18) can be solved with a combination of a multishift conjugate gradient least-squares (mCGLS) method and a local variable exchange method (see [25] for details).

2.2.3 Dynamic PTx for Non-Selective $k_\mathbf{T}$-point Pulses

Dynamic PTx is the extension of static PTx, where the RF-pulse is divided into a set of constant-weighted sub-pulses. We will now focus on pulse design where N_{k_T}-spoke sub-pulses share a common waveform across all channels. Let $\rho(t)$ be the sub-pulse waveform, with timely equidistant samples $\rho_m \equiv \rho(t_m)$,

[8] A constant gradient can easily be included for slice-selectivity, but is omitted here for simplicity.

$m = 1, \ldots, N_t$, and keep the same spatial discretization $\mathbf{r}_n$ as before. Following [26], defining w_{jk} as the complex weight of the j^{th} channel for the k^{th} sub-pulse, $\Delta t \equiv t_2 - t_1$ as the sampling period, t'_k the remaining time from the end of the k^{th} sub-pulse to the end of the RF-pulse, and

$$a_{knj} \equiv i\gamma M_0 \Delta t S_j(\mathbf{r}_n) e^{i\mathbf{r}_n \cdot \mathbf{k}(t'_k)} \sum_{m=1}^{N_t} \rho_m e^{i\gamma \Delta B_0(\mathbf{r}_n)(t'_k + (N_t - m)\Delta t)},$$

with

- A_k as the $N_s \times N_{\text{C}}$ system matrix whose entry at (n, j) is a_{knj}
- $\mathbf{p_k}$ as the $N_{\text{C}} \times 1$ vector whose j^{th} entry is w_{jk}
- $\mathbf{m}$ as the $N_s \times 1$ vector whose n^{th} entry is $M_{x'y'}(\mathbf{r}_n, T_p)$

the discretized approximation to (16) for an N_{spk}-spokes RF-pulse can be written as

$$\mathbf{m} \approx A\mathbf{p}, \tag{19}$$

where we have used the horizontal concatenation $[\cdot]$ to define

$$A \equiv [A_1 \ A_2 \ \ldots \ A_{N_{k_{\text{T}}}}], \ \mathbf{p} \equiv [\mathbf{p}_1 \ \mathbf{p}_2 \ \ldots \ \mathbf{p}_{N_{k_{\text{T}}}}]^T.$$

This can, similar to (18), be cast as a regularized magnitude least-squares problem for the estimation $\hat{\mathbf{p}}$ of $\mathbf{p}$, if we only look to optimize the modulus of $\mathbf{m}$ towards a magnetization target $\mathbf{m}_{\text{tar}}$, i.e.

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \left(|| \, |A\mathbf{p}| - \mathbf{m_{tar}} \, ||_2^2 + \lambda ||\mathbf{p}||_2^2 \right). \tag{20}$$

of the previous section 2.2.3. The k-space trajectory $\mathbf{k}(t)$ is here of course an $N_{k_{\text{T}}}$-k_{T}-points trajectory.

2.2.4 Universal Pulse Designs

We will briefly introduce the concept of *universal pulse* (UP) design. UPs are designed by jointly optimizing the PTx-weights for a given $\mathbf{m}_{\text{tar}}$ for e.g. RF-shimming (eq. (18)) or k_{T}-points (eq. (20)), over different volunteers' data (i.e. their transmit sensitivities and/or B_0 data). This yields PTx-weights which gives an estimated optimal compromise between the volunteers for the target. Ideally, this obviates the necessity the time-costly procedures of gathering patient data and subsequently optimizing for a patient-tailored pulse designs. However, universal pulse relies on the assumption that the sensitivity and B_0-fields varies sufficiently little between patients such that the pulse gives a resulting excitation pattern sufficiently replicates the desired excitation pattern. What is deemed *sufficient* is based on the application – for e.g. whole-brain FA homogenization using k_{T} points, universal pulses are shown to give results on-par with tailored-pulses, however, not quite as inch-perfect as their patient-tailored counter-parts[27].

One way to cast the UP optimization problem is to construct the each volunteers' system matrix and target vector as presented in section 2.2.2 for RF-shimming or 2.2.3 for k_{T}-points. The system matrices and target vectors

are then vertically concatenated, respectively, to yield a new UP system matrix and UP target matrix. These can subsequently be plugged in to replace the system matrix and target vector, respectively, in eq. (18) for RF-shimming or eq. (20) for k_{T}-points, and solved in the same manner as the non-UP optimization problems. We will refer to the UP pulses designed for RF-shimming as *RF-UP*, and those designed for a k_{T}-trajectory as *k_{T}-UP*.

2.2.5 L-curve Approach for Regularization

The quadratic optimization problems posed in eq. (18) and (20) require the Tikhonov regularization parameter λ to be given before any optimization can be performed. The choice of this parameter value can be decided by means of an *L-curve approach*, where the optimization problem is solved for a set of parameter values $\lambda = \lambda_i \in [\lambda_{\min}, \lambda_{\max}]$, and choosing the solution corresponding to the λ_i for which the curvature of the graph traced by plotting the solution norm versus the normalized residual norm has the highest curvature. In the case of e.g. eq. (20), this can be stated as choosing the solution $\hat{\mathbf{p}}_{\lambda_i}$ for which the graph traced by the points

$$\left(|| \, |A\hat{\mathbf{p}}_{\lambda_i}| - \mathbf{m_{tar}} \, ||_2^2 \; / \; ||\mathbf{m_{tar}}||_2^2, \;\; ||\hat{\mathbf{p}}_{\lambda_i}||_2^2 \right),$$

has the highest curvature. An example is shown in figure for $\lambda_i \in [0.1, 1000]$.

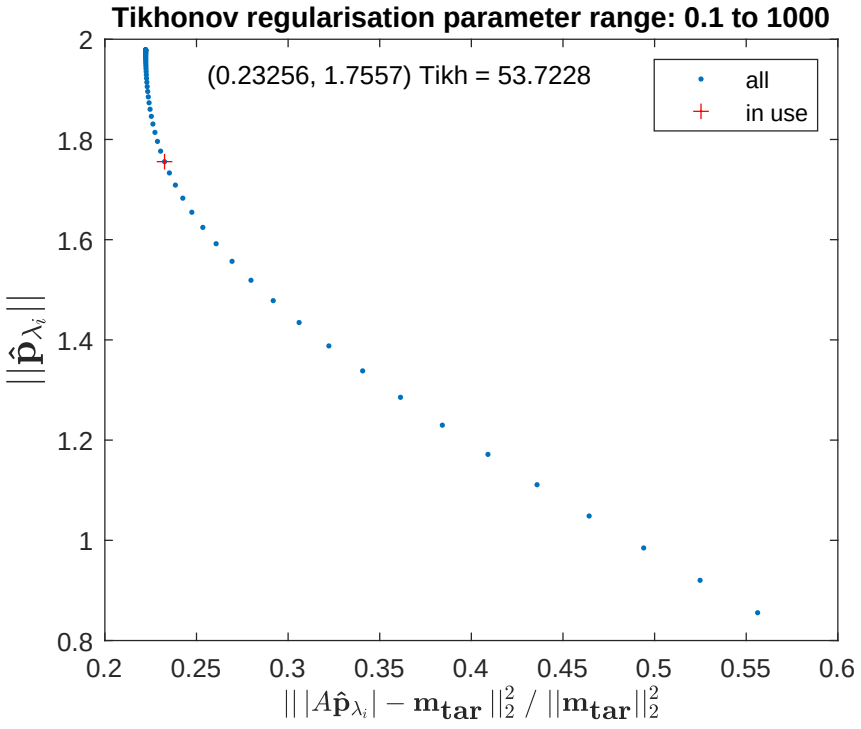


Figure 4: An example of the L-curve approach, here for eq. (20). The corner value chosen here corresponds to $\lambda_i = 53.7228$.

2.2.6 Local and Whole-Brain Specific Absorption Rate (SAR)

Associated with the RF-field from each PTx channel j is also its electric field $\mathbf{E}_j(\mathbf{r}, t)$. As the amplitudes and relative phases of each channel are changed, one needs to consider not only the superposition of each channel's magnetic field, but also their combined electric field,

$$\mathbf{E}(\mathbf{r}, t) = \sum_{j=1}^{N_\mathrm{C}} \mathbf{E}_j(\mathbf{r}, t),$$

as it gives rise to energy deposition in tissue and thus causes heating in the subject being scanned – focal heating is a prominent issue at UHFs[28]. Therefore, for any PTx configuration, the associated electric field needs to be accounted and ensured to give both local and global SAR levels which are within regulatory limits before the pulse can be applied for *in vivo* scanning.

We now assume our PTx weights are normalized to the relative amplitudes of a chosen maximum driving voltage, V_max, over all the channels, such that $|w_j| \leq 1$. This is the same as choosing (see eq. (15))

$$p(t) = \tilde{p}(t) V_\mathrm{max}, \tag{21}$$

where $\tilde{p}(t)$ it the normalized waveform s.t. $|\tilde{p}(t)| \leq 1 \; \forall \; t$. We now decompose each channel's electric field in the same manner as we did for their magnetic fields in eq. (14), into a *normalized spatial field* $\tilde{\mathbf{E}}_j(\mathbf{r})$, and its temporal part (same as for its channel's associated magnetic field's waveform) $p_j(t) \equiv p(t) w_j(t)$, i.e.

$$\mathbf{E}_j(\mathbf{r}, t) \approx \tilde{\mathbf{E}}_j(\mathbf{r}) p_j(t) = \tilde{\mathbf{E}}_j(\mathbf{r}) \tilde{p}(t) V_\mathrm{max} w_j(t)),$$

such that we now can write the combined electric field as the matrix multiplication

$$\mathbf{E}(\mathbf{r}, t) = V_\mathrm{max} \tilde{p}(t) \tilde{\mathbb{E}}(\mathbf{r}) \mathbf{w}(t),$$

where we have defined $\tilde{\mathbb{E}}(\mathbf{r})$ as the $3 \times N_\mathrm{C}$ matrix whose entry at (i, j) is the normalized electric field component $\tilde{E}_{i,j}$ in the i^th spatial direction from the j^th channel. The vector $\mathbf{w}(t)$ is as defined in section 2.2.2, except its entries are here time-dependent. By adapting the SAR-calculations presented in [29], the local SAR in a sample of volume V at a position $\mathbf{r}$ during a sample period t_i can calculated as

$$\mathrm{SAR}_\mathrm{sample}(\mathbf{r}, t_i) = V_\mathrm{max}^2 |\tilde{p}(t_i)|^2 \mathbf{w}(t)^H \mathbb{Q}(\mathbf{r}) \mathbf{w}(t), \tag{22}$$

where superscript H indicates taking the Hermitian transpose and $\mathbb{Q}(\mathbf{r})$ is the *Q-matrix* (of size $N_C \times N_C$), calculated as

$$\mathbb{Q}(\mathbf{r}) \equiv \frac{1}{V} \int_V \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})} \tilde{\mathbb{E}}(\mathbf{r})^H \tilde{\mathbb{E}}(\mathbf{r}) \mathrm{d}V.$$

with the electrical conductivity $\sigma(\mathbf{r})$ and mass density $\rho(\mathbf{r})$ of the tissue. (22) over all N_t time-samples of the the applied RF-pulse, the local SAR of the entire pulse can be calculated as

$$\mathrm{SAR}_{\mathrm{pulse}}(\mathbf{r}) = \frac{V_{\max}^2}{N_t} \sum_{m=1}^{N_t} |\tilde{\mathbf{p}}(t_i)|^2 \mathbf{w}(t_i)^H \mathbb{Q}(\mathbf{r}) \mathbf{w}(t_i) \tag{23}$$

Eq. (23) can further be multiplied by a *duty-cycle* factor, i.e. the ratio between the pulse duration and sequence repetition time to give the realistic SAR time-average of the pulse used in a sequence.

The normalized electric fields can be estimated by a numerical simulation[5] of Maxwell's equations for a set of voxels, assuming a model head as the subject. The model head can e.g. represent a healthy adult male, such as the model Duke[28]. For computational efficiency, the grid of voxels can be down-sampled by using the *virtual observation points* (VOPs) technique[30] for a conservative estimation of maximum local SAR.

2.2.7 Sensitivity Mapping

The sensitivities $S_j(\mathbf{r_n})$ can be inferred from repeating the DREAM sequence for $M \geq N_{\mathrm{C}}$ measurements by the unity-weighted encoding process[31]

$$S_j(\mathbf{r_n}) = \sum_{m=1}^{M} \epsilon_{j,m} \hat{B}_{1,m}^{+}(\mathbf{r}_n), \tag{24}$$

where we have defined

- $\epsilon_{j,m}$ as the entry at (j, m) of the matrix $((E^H E)^{-1} E^H)$
- $\hat{B}_{1,m}^{+}(\mathbf{r}_n) \in \mathbb{C}^M$ is the estimated average (complex) B_1^+-field map at voxel position $\mathbf{r}_n$ from the m^{th} measurement of the DREAM sequence.

Here, $E \in \mathbb{R}^{M \times N_{\mathrm{C}}}$ is the encoding matrix whose entry at (m, j) is the static PTx complex weight of the j^{th} channel for the m^{th} measurement.

For each measurement m, we take into account the relative channels phases by a phase-preserving sum-of-magnitude reconstruction method[18, eq. (23)]. Let each *receiving channel* be indexed by $k = 1, \ldots, K$, and let $I_{k,m}$ be the complex intensity at an arbitrary voxel in either the FID- or STE-image measured by receive channel k in measurement m. For each voxel, we choose the measurement m_{ref} as reference, which has the maximum sum-of-magnitude intensity,

$$m_{\mathrm{ref}} = \underset{m=1,\ldots,M}{\arg\max} \sum_{k=1}^{K} |I_{k,m}|$$

and perform the phase-preserving reconstruction such that the resulting (complex) intensity contribution $\hat{I}_m$ to the corresponding voxel in the m^{th} reconstructed FID- or STE-image image can be written

$$\hat{I}_m = \sum_{k=1}^{K} \frac{I_{k,m} I_{k,m_{\mathrm{ref}}}^*}{|I_{k,m_{\mathrm{ref}}}|}. \tag{25}$$

The magnitude of the map $\hat{B}_{1,m}^{+}$ is now found by eq. (13) for each measurement m, but replacing both the complex waveform-integral and intensity ratio by their respective magnitudes in the calculation. The phase of $\hat{B}_{1,m}^{+}$ is set equal to

the phase of $\hat{I}_m$ for each corresponding voxel, where the choice of the FID- or STE-image as reference is the same across all measurements.

We now assume we do M measurements with each channel transmitting with equal amplitude for all measurements, but for measurement m we set the relative phase of the j^{th} channel (and thus the entry at (m, j) of the encoding matrix E) to

$$\exp\left(\frac{2\pi i(m-1)j}{M}\right).$$

We can now show that (see section 7.2.5 of the appendix)

$$E^H E = M I_M,$$

where $I_M \in \mathbb{R}^{M\times M}$ is the identity matrix, giving

$$\left((E^H E)^{-1} E^H\right) = \frac{E^H}{M} \implies \epsilon_{j,m} = \frac{\exp\left(\frac{2\pi i(1-m)j}{M}\right)}{M}.$$

We can now use eq. (24) to calculate each channel's sensitivity map after all M measurements are gathered. We note that the sensitivities are just the Discrete Fourier Transform[32] over the sequence of measurements.

2.3 Deep Neural Regression

Deep Regression is the process of performing regression by means of a Deep Neural Network (DNN), here with a fully-connected, feed-forward, multi-layer perceptron (MLP) network. Here, an input $\mathbf{X} \in \mathbb{R}^{N_{\text{in}}}$ is propagated forward through the network $\text{NET}_\Theta : \mathbb{R}^{N_{\text{in}}} \to \mathbb{R}^{N_{\text{out}}}$, parameterized by its weights and biases, jointly referred to as Θ, to make a predicted output $\hat{\mathbf{Y}} \in \mathbb{R}^{N_{\text{out}}}$,

$$\hat{\mathbf{Y}} = \text{NET}_\Theta(\mathbf{X}).$$

2.3.1 The Forward Pass

The network consists of a structure of *layers* $l = 0, 1, \ldots, L$ of *nodes*, where each layer is associated with a *forward* function. Let $l = 0$ be the *input layer*, $l = 1, \ldots, L-1$ be the *hidden layers*, $l = L$ be the *output layer*. Denote the forward function of a hidden layer or the output layer l as $f_{\Theta_l} : \mathbb{R}^K \to \mathbb{R}^J$, which is parameterized by its set Θ_l of weights and biases for K nodes of the preceding layer $l-1$ and J nodes in layer l. The layers $l = 0, L$ have with N_{in} and N_{out} nodes, respectively. Introducing $\Theta \equiv \{\Theta_1, \ldots, \Theta_L\}$ as the set of all network parameters, we can write the network's forward pass as the composition of each preceding layer's forward function,

$$\text{NET}_\Theta(\mathbf{X}) = f_{\Theta_L} \circ f_{\Theta_{L-1}} \circ \ldots \circ f_{\Theta_1}(\mathbf{X}).$$

We will now shift focus to the forward pass on a layer-by-layer basis: let $k = 1, \ldots, K$ and $j = 1, \ldots, J$ count over the nodes of the preceding layer $l-1$ and current layer l, respectively. For layers $l = 1, \ldots, L$, introduce

- $W^{(l)} \in \mathbb{R}^{J \times K}$ as the weight matrix of layer l whose entry at (j, k) is the weight $w_{jk}^{(l)}$ connecting node k to j
- $\mathbf{b}^{(l)} \in \mathbb{R}^J$ as the bias vector of layer l whose j^{th} entry is the bias term $b_j^{(l)}$ of node j
- $\mathbf{a}^{(l-1)} \in \mathbb{R}^K$ as the activation vector of layer $l-1$ whose k^{th} entry is the activation $a_k^{(l-1)}$ of node k
- $\mathbf{z}^{(l)} \in \mathbb{R}^J$ as the *weighted sum* vector of layer l whose j^{th} entry is the weighted sum $z_j^{(l)} \equiv \sum_k w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)}$ of node j
- $\mathbf{a}^{(l)} \in \mathbb{R}^J$ as the activation vector of layer l whose j^{th} entry is the activation $a_j^{(l)} \equiv \sigma(z_j^{(l)})$ of node j

where $\sigma : \mathbb{R} \to \mathbb{R}$ is the *activation function* (or *non-linearity*), here chosen to be the same across all nodes of the hidden layers for simplicity. We can now write the forward pass to from layer $l-1$ to layer l in matrix form as

$$\mathbf{a}^{(l)} \equiv \sigma_{\text{vec}}\left(\mathbf{z}^{(l)}\right) \equiv \sigma_{\text{vec}}\left(W^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}\right). \tag{26}$$

Here, $\sigma_{\text{vec}} : \mathbb{R}^J \to \mathbb{R}^J$ is the (vector) activation function for which $\sigma(\cdot)$ is applied element-wise to its input. In short, computing a single forward pass is straight forward – one computes eq. (26) for layers $l = 1, \ldots, L$ (in that order).

For completeness, we here introduce the *rectified layer unit* (ReLU) activation function,

$$\sigma(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

for the hidden layers. For layer L, we assume its activation function is always the identity mapping $x \mapsto x$. We notice a node with a ReLU activation function will feed forward its node output in the network if and only if it is non-negative. The ReLU activation has been shown to have benefits over many other common activation functions[33][34].

As a final remark, we note that under the activation notation, we can explicitly write the network's input $\mathbf{X}$ and output $\hat{\mathbf{Y}}$ as

- $\mathbf{a}^{(L)} \equiv \hat{\mathbf{Y}}$
- $\mathbf{a}^{(0)} \equiv \mathbf{X}$

An illustration of an MLP network is given in figure 5.

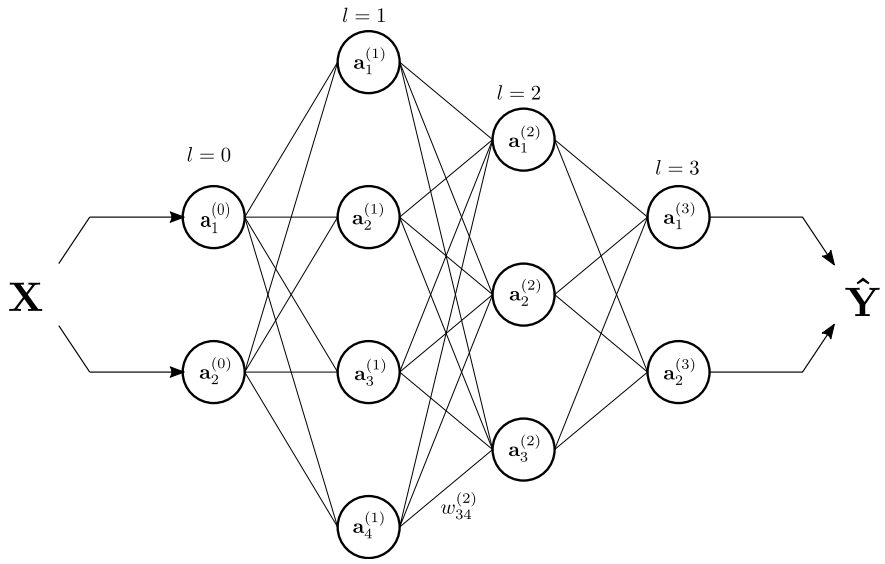


Figure 5: An illustration used to visualize an MLP network. Here, $N_{\text{in}} = N_{\text{out}} = 2$, and we have chosen $L = 3$ and hidden layers $l = 1, 2$ of sizes $J = 4, 3$, respectively. The edges connecting each node are the weights, indicated explicitly for the edge between node 4 of layer $l = 2$ to node 3 of layer $l - 1 = 1$.

2.3.2 The Cost Function

In order to measure the network's performance, we introduce the cost function $C : \mathbb{R}^{N_{\text{out}}+N_{\text{out}}} \to \mathbb{R}$, here as the half-mean-square-error (hMSE) metric:

$$C \equiv C(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{2N_{\text{out}}} ||\mathbf{Y} - \hat{\mathbf{Y}}||_2^2$$

Here, $\mathbf{Y} \in \mathbb{R}^{N_{\text{out}}}$ is the given true output which we desire from the network for a prediction $\hat{\mathbf{Y}}$.

2.3.3 Weight-Regularization of the Cost Function

Overfitting refers to the case where a network data trains well (i.e. returns a low cost) over the training examples, but fails to generalize to examples which are not contained in the training set. To reduce overfitting, one can add a Tikhonov regularization term R_{λ_w}, paramterized by the weight regularization factor $\lambda_w > 0$, to the cost function to punish having large network weights (i.e. weight decay). The new objective function J subject to minimization during training is now the *regularized* cost function, i.e. the original cost function C, but with the added regularization term:

$$J \equiv C + R_{\lambda_w}, \quad \text{where } R_{\lambda_w} \equiv \frac{\lambda_w}{2} \sum_{l,j,k} \left(w_{j,k}^{(l)} \right)^2 .$$

2.3.4 The Network Gradient

In order to improve the network's performance, we wish to minimize the regularized cost function with respect to the set Θ of all network parameters,

$$\Theta \equiv \left\{ w_{jk}^{(l)}, b_j^{(l)} \right\},$$

where l, j, k run over all valid indices. This can be done through e.g. gradient descent, which requires its gradient (or an approximation to it) with respect to Θ. Thus, for a single given true output $\mathbf{Y_n} \equiv [y_1, \ldots, y_{N_{\text{out}}}]^T$ and input $\mathbf{X}_n$ with predicted output $\hat{\mathbf{Y}}_\mathbf{n} \equiv [a_1^{(L)}, \ldots, a_{N_{\text{out}}}^{(L)}]^T$, we wish to calculate the network gradient $\nabla_\theta C_n$ as the set that contains all partial derivatives for the parameters in the network with respect to the non-regularized cost function,

$$\nabla_\theta C_n \equiv \left\{ \frac{\partial C_n}{\partial w_{jk}^{(l)}}, \frac{\partial C_n}{\partial b_j^{(l)}} \right\}$$

where $C_n \equiv C(\mathbf{Y}_n, \hat{\mathbf{Y}}_n)$ and l, j, k run over all valid indices – in addition, we also wish to calculate the set that contains all partial derivatives for the parameters in the network with respect to the regularization term,

$$\nabla_\theta R_{\lambda_w} \equiv \left\{ \frac{\partial R_{\lambda_w}}{\partial w_{jk}^{(l)}}, \frac{\partial R_{\lambda_w}}{\partial b_j^{(l)}} \right\} = \left\{ w_{jk}^{(l)}, 0 \right\},$$

where l, j, k run over all valid indices. We do not specify the form of $\nabla_\theta C_n$ or $\nabla_\theta R_{\lambda_w}$ further than saying that we will make use of the notation

$$\begin{aligned} &\Theta - \eta \left(\frac{1}{N} \sum_n \nabla_\theta C_n - \nabla_\theta R_{\lambda_w} \right) \\ &\qquad \equiv \left\{ w_{jk}^{(l)} - \eta \left(\frac{1}{N} \sum_n \frac{\partial C_n}{\partial w_{jk}^{(l)}} - \lambda_w w_{jk}^{(l)} \right), b_j^{(l)} - \frac{\eta}{N} \sum_n \frac{\partial C_n}{\partial b_j^{(l)}} \right\} \end{aligned} \tag{27}$$

for any $\eta, N > 0$, and l, j, k run over all valid indices.

2.3.5 The Backward Pass

We now wish to state an efficient way of calculating $\nabla_\theta C_n$. We will do this by *backpropagation* (BP), in which we calculate the *error* at each layer l, starting from L, and propagate the error backwards to the preceding layer $l-1$. Because of the implicit dependencies of the activations $\mathbf{a}^{(l)}$ on $\{\Theta_l, \Theta_{l-1}, \dots, \Theta_1\}$, it is natural to approach the problem with the chain rule. We now define[35]

- $\boldsymbol{\delta}_n^{(l)} \in \mathbb{R}^J$ as the error vector of layer l for example n whose j^{th} entry is the error[9]
$$\delta_{j,n}^{(l)} \equiv \frac{\partial C_n}{\partial a_j^{(l)}}$$
of node j. Equivalently defined for layer $l-1$ for data n, but with $l \to l-1,\ J \to K,\ j \to k$.

- $\nabla_a C_n \in \mathbb{R}^{N_{\text{out}}}$ as the cost gradient at the output layer L for example n, whose j^{th} entry is
$$\frac{\partial C_n}{\partial a_j^{(L)}} = \frac{\left(y_j - a_j^{(L)}\right)}{N_{\text{out}}}$$
.

- $\nabla_w^{(l)} C_n \in \mathbb{R}^{J \times K}$ as the weight gradient matrix of non-input layer l for example n, whose entry at (j,k) is
$$\frac{\partial C_n}{\partial w_{jk}^{(l)}}$$

- $\nabla_b^{(l)} C_n \in \mathbb{R}^J$ as the bias gradient vector of non-input layer l for example n whose j^{th} entry is
$$\frac{\partial C_n}{\partial b_j^{(l)}}$$

- ${\sigma_{\text{vec}}}'\left(\mathbf{z}^{(L)}\right) \in \mathbb{R}^J$ as the vector of layer l whose j^{th} entry is $\sigma'(z_j^{(L)})$

and express the essential equations in terms of matrices we can compute:

$$\boldsymbol{\delta}_n^{(L)} = \nabla_a C_n \tag{BP1}$$

$$\boldsymbol{\delta}_n^{(l)} = \left(W^{(l+1)}\right)^T \left(\boldsymbol{\delta}_n^{(l+1)} \odot {\sigma_{\text{vec}}}'\left(\mathbf{z}^{(l+1)}\right)\right) \tag{BP2}$$

$$\nabla_w^{(l)} C_n = \left(\boldsymbol{\delta}_n^{(l)} \odot {\sigma_{\text{vec}}}'\left(\mathbf{z}^{(l)}\right)\right)\left(\boldsymbol{a}^{(l-1)}\right)^T \tag{BP3}$$

$$\nabla_b^{(l)} C_n = \boldsymbol{\delta}_n^{(l)} \odot {\sigma_{\text{vec}}}'\left(\mathbf{z}^{(l)}\right) \tag{BP4}$$

Here, $\odot$ indicates element-wise multiplication. These four equations are known as the BP equations, and are computed after all activations are found from a single forward pass. In short, for a single backward pass, we compute (BP1) and (BP3)-(BP4) for $l = L$, and repeat (BP2)-(BP4) for $l = L-1, \dots, 1$. Element-wise derivation of the above equations are given in the appendix, see section 7.2.3.

[9]The error can be chosen to be defined as $\partial C_n / \partial z_j^{(l)}$, and makes some of the forumlas presented shorter. However, the choice of error in the main text is done to accommodate for the forward- and backward pass through convolutional layers, see sections 2.3.9 and 7.3.

2.3.6 Initializing Network Parameters

Before we can use the network, we need to initialize its weights and biases. A common way[36] to initialize the biases is to set them equal to zero. For the weights, the He initialization method can be used: at layer l, each weight $w_{jk}^{(l)}$ is sampled from a normal distribution with zero mean and variance $2/K$.

2.3.7 Deep Learning

With the forward and backward pass for a single training example $(\mathbf{X}_n, \mathbf{Y}_n)$ accounted for, we now introduce Deep Learning as the process of optimizing the regularized cost function J on the grounds of training data set $\{(\mathbf{X}_n, \mathbf{Y}_n)\}_{n=1,\dots,N_{\text{train}}}$. We will here focus on the common learning algorithm stochastic gradient descent (SGD). We will run through the entire training set for $e = 1, \dots, N_{\text{epoch}}$ epochs. For each epoch, we will shuffle the order of the training set randomly, and split the data set into N_{mini} mini-batches, each mini-batch of size $N_{\text{MBS}} \equiv \lfloor N_{\text{train}}/N_{\text{mini}} \rfloor$. After each mini-batch is run through, we update the network parameters by doing a step $\Delta\Theta$ in the direction of (approximated) steepest descent for J,

$$\Delta\Theta := -\eta \left(\frac{1}{N_{\text{MBS}}} \sum_{n=1}^{N_{\text{MBS}}} \nabla_\theta C_n + \nabla_\theta R_{\lambda_w} \right),$$

where $\eta > 0$ is the *network learning parameter*. As the training progresses past a certain amount of epochs, the learning rate from outset may become too large, and the training algorithm may begin to overshoot. To accommodate for this, certain parameters often replace the learning rate: the *initial learning rate* η_0 is the learn rate used from the first iteration, with a *learning rate schedule* determining when changes to the learning rate should occur over the training period (e.g. *piece-wise* implying a periodic change). In the case of a piece-wise learning rate schedule, the *learning rate drop period* $P \in \mathbb{N}$ determines the amount of epochs passed before the current learning rate is multiplied by the *learning rate drop factor* $0 \leq D \leq 1$.

A small adjustment can be made to the SGD-algorithm. Before a new step is made in the approximated steepest descent direction, the direction is perturbed to include a fraction of the direction from the previous step. The new direction keeps some of its previous "momentum", thus giving rise to the adjusted algorithm stochastic gradient descent with momentum (SGDM). The fraction included is decided by the *momentum coefficient* $\alpha_m \in [0, 1]$. This method is often more robust than its standard counter-part as it reduces oscillations in the search trajectory.

During training, the network may exhibit signs of either overfitting or *underfitting*, the latter referring to a network which is unable to capture the degree of non-linearity in the training data[37]. To detect if these effect are present in the network during training, the cost function is calculated over a validation data set, with a *validation frequency* determining the amount of epochs passed before the validation cost is re-calculated. The validation set is ideally generated separately from the training library, and should capture the data which we would like the network to generalize to after training is complete. In the case of overfitting, *early stopping* of the training can be implemented to terminate the training by introducing a *validation patience*, i.e. the number of times that the validation

cost can be equal to or larger than its previously smallest calculated value before network training is terminated. In the case of underfitting, the mean validation and mean training cost are equal or nearly equal, and neither cease to decrease in spite of further training. In this case, the network is not complex enough and/or the quality of the training data is insufficient.

A simplified example of Deep Learning with SGDM is demonstrated by the pseudo-code presented in Algorithm 1. Note that we have left out all the validation procedures in the example to make the code short and simple to read.

Algorithm 1 Simplified Deep Learning with Stochastic Gradient Descent with Momentum (SGDM)

Require: network architecture and its layers' activation function $\sigma_{\text{vec}}(\cdot)$
Require: training set $\{(\mathbf{X}_n, \mathbf{Y}_n)\}_{n=1,\ldots,N_{\text{train}}}$
Require: weight regularization factor $\lambda_w > 0$
Require: momentum coefficient $0 \leq \alpha_m \leq 1$
Require: initial learning rate $\eta > 0$
Require: learning rate drop period P
Require: learning rate drop factor D
Require: mini-batch size N_{mini}
Require: number of epochs N_{epoch}
  initialize biases (e.g. to zero) and weights (e.g. He initializer)
  initialize search direction $\Delta\Theta := 0$
  set mini-batch size $N_{\text{MBS}} := \lfloor N_{\text{train}}/N_{\text{mini}} \rfloor$
  for epoch $e = 1, \ldots, N_{\text{epoch}}$ **do**
    if e mod $P == 0$ **then**
      $\eta := D\eta$
    end if
    shuffe training and split shuffled set into N_{mini} mini-batches of sizes N_{MBS}
    for mini-batch $m = 1, \ldots, N_{\text{mini}}$ **do**
      for example $n = 1, \ldots, N_{\text{MBS}}$ in mini-batch **do**
        for layer $l = 1, \ldots, L$ (forward pass) **do**
          $\mathbf{a}^{(l)} := \sigma_{\text{vec}}\left(W^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}\right).$
        end for
        for layer $l = L, \ldots, 1$ (backward pass) **do**
          if $l == L$ **then**
            $\boldsymbol{\delta}_n^{(L)} := \nabla_a C_n$
          else
            $\boldsymbol{\delta}_n^{(l)} := \left(W^{(l+1)}\right)^T \left(\boldsymbol{\delta}_n^{(l+1)} \odot {\sigma_{\text{vec}}}'\left(\mathbf{z}^{(l+1)}\right)\right)$
          end if
          $\nabla_w^{(l)} C_n := \left(\boldsymbol{\delta}_n^{(l)} \odot {\sigma_{\text{vec}}}'\left(\mathbf{z}^{(l)}\right)\right)\left(\boldsymbol{a}^{(l-1)}\right)^T$
          $\nabla_b^{(l)} C_n := \left(\boldsymbol{\delta}_n^{(l)} \odot {\sigma_{\text{vec}}}'\left(\mathbf{z}^{(l)}\right)\right)$
        end for
      end for
      $\Delta\Theta := -\eta\left(\frac{1}{N_{\text{MBS}}}\sum_n \nabla_\theta C_n + \nabla_\theta R_{\lambda_w}\right) + \alpha_m \Delta\Theta$
      $\Theta := \Theta + \Delta\Theta$
    end for
  end for

2.3.8 The Adaptive Moment Estimation (Adam) Solver

By treating the objective function itself as a stochastic function, adjustments can be made to adapt the learning rate by a bias-corrected estimation of the function's first- and second order moments – which is the idea behind the *Adaptive Moment Estimation* (Adam) solver[38]. In short, each iteration t made while attempting to minimize the objective function comes down to computing a moving average of the gradient, m_t, and a moving average of the gradient's squared values, v_t, defined recursively as

$$\begin{aligned} m_t =& \beta_1 m_{t-1} + (1-\beta_1)\left(\frac{1}{N_{\text{MBS}}}\sum_n \nabla_\theta C_n + \nabla_\theta R_{\lambda_w}\right), \\ v_t =& \beta_2 v_{t-1} + (1-\beta_2)\left(\frac{1}{N_{\text{MBS}}}\sum_n \nabla_\theta C_n + \nabla_\theta R_{\lambda_w}\right)^2, \end{aligned}$$

and updating the learning rate η_t and descent-step at iteration t as

$$\begin{aligned} \eta_t :=& \eta\frac{\sqrt{(1-\beta_2^t)}}{(1-\beta_2^t)}, \\ \Delta\Theta :=& -\eta_t\frac{m_t}{\sqrt{v_t}+\epsilon}. \end{aligned}$$

Here, $\beta_1, \beta_2 \in [0,1]$ are known as the *gradient decay factor* and *squared gradient decay factor*, respectively, and ϵ is the *offset factor* which can help the algorithm avoid division by (values close to) zero when the second-order moving average is small. Both averages are initialized to zero for $t = 0$. The square- and square-root operations in the equations above refer to element-wise operations.

2.3.9 Convolutional and Pooling Layers

We will here briefly introduce the concept of convolutional and pooling layers, and how a one can extend the definitions of the previous discussion on regressional MLP networks to a neural network containing these layers, i.e. a convolutional neural network (CNN). For the sake of simplicity, the introduction will here focus on 2-D convolutional layers (i.e. convolution layers which accept and pass forward 2-D *feature maps*, e.g. grayscale images). All the presented discussion can easily be extended to N-D. Furthermore,the mathematical details of the forward pass and backward pass through such layers are covered extensively in section 7.3 of the appendix. They are omitted here due to their cumbersome and lengthy derivations.

A 2-D convolutional layer is a layer in a neural network which takes a set of C_1 channeled *input feature maps* of size $H_1 \times W_2$, convolves each input with a unique kernel/filter of size $k_1 \times k_2$, and sums up the input to produce an *output feature map* of size $H_2 \times W_2$ (the latter's size is decided by the *padding* and *stride* of the convolution process) after passing the sum through an activation function. This process can be repeated C_2 number of times with different sets of kernels to produce C_2 channels. Thus, the resulting total set of kernels is a 4-D tensor of size $C_2 \times C_1 \times k_1 \times k_2$. The process is illustrated in 6.

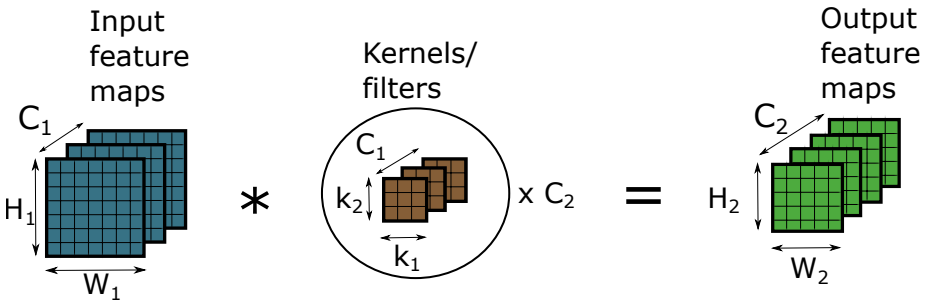


Figure 6: The forward pass of a 2-D convolutional layer. Each of the C_1 *input feature maps* of size $H \times W$ are convolved with a unique $k_1 \times k_2$ kernel. For each input feature map, the resulting convolution is summed up to produce an output feature map, which size depends on the details (padding, stride) of the convolution process. In the figure, *valid* convolution (i.e. no padding with unity stride) is shown for $C_1 = 3$ input channels of size $H_1 \times W_1 = 7 \times 7$, each kernel of size $k_1 \times k_2 = 3 \times 3$, which produces an $H_2 \times W_2 = 5 \times 5$ output, where we have chosen $C_2 = 4$ output channels.

Convolutional layers introduce the idea of shared parameters – instead of a full connection table between the input and output pixels (e.g. as in a FCL), the number of trainable parameters is reduced to the choice of kernel size. This is beneficial in terms of overfitting and combating the unstable gradient problem which is very common in fully connected feed-forward networks.

As we are training a *kernel* rather than a fully connected table of parameters, the kernel is trained to recognize *features* (e.g. shapes) which ultimately reduces the cost function. Thus, this layer introduces *translational invariance* of our input feature maps, rather than training on a pixel-by-pixel basis of the input layer.

Note that an FCL is equivalent to a non-padded convolution layer with kernel dimensions matching the input feature map dimensions. Therefore, any fully-connected MLP network can be regarded as a CNN with each convolutional layers having said structure, with the number of "fully connected nodes" chosen by setting the layers' number of output channels.

Usually present in a convolutional network architectures are pooling layers, e.g. max pooling layers or average pooling layers, whose function is to down-sample the feature map sizes and to reduce overfitting of the networks. A max pooling layer is similar to the convolutional layer, except its kernel has no optimizeable parameters, and its kernel only passes forward the *maximum* value for a given input-with-kernel "overlap", where the input channel size matches the output channel size (each output channel is the max-sampled version of its input channel). An average pooling is similar to the max pooling layer, as its kernel also has no optimizeable parameters, but passes forward the *average* value for a given input-with-kernel "overlap", where the input channel size matches the output channel size (each output channel is the averaged-sampled version of its input channel).

2.3.10 Previous applications of Machine Learning in Pulse Designs

Several examples where Machine Learning (not necessarily restricted to *Deep* Learning) has been applied to pulse designs can be found in recent MRI literature. Ianni et al.[39] successfully predicted shimming weights using a machine learning approach, from a large set of simulated sensitivity maps and corresponding MLS shim solutions. This was done using a method of Iteratively Projected Ridge Regression, which had the advantage of not requiring full information of the sensitivity maps, while being SAR-efficient and slice-specific.

Vinding et al.[40] trained an MLP network to predict RF-settings for localized excitation on training sets constructed from an image library. The images were processed to create target flip-angle maps across an 64×64-grid, with corresponding best RF-settings calculated by means of optimisation. This served as training data for the neural networks. The predicted RF-settings from the network were deemed feasible even with a relatively small ($N \approx 2000$) training set.

3 Material and Methods

3.1 Volunteer Scans

The MRI data used for this thesis were all provided and collected prior to this thesis from an 8-channel transmit (Tx), 32-channel receive (Rx) Nova head coil on a Siemenes Magneton 7T in Maastricht, The Netherlands, and consisted of 17 scans. Associated with each head scan is the data needed to construct the complex B_1^+-sensitivity map per transmission channel and the B_0-map.

Data for the complex B_1^+-maps were collected with a DREAM sequence (see figure 2) to obtain stacked slices in the head-feet (HF) orientation ($\alpha = 49°$, $\beta = 7°$, TR = 6.20ms, TE_{STE} = 1.98ms, TE_{FID} = 3.96ms, T_s = 1.98ms, T_d = 7.10ms, FOV = 256×224 mm^2 (anterior-posterior (AP) $\times$ right-left (RL)), voxel size = 4 mm isotropic, phase-encoding direction right-left, preparation pulse slice thickness = 8 mm, imaging pulse slice thickness = 4 mm, imaging inter-slice distance (center-to-center) = 8 or 10 mm (varied between scans), even-odd slice ordering, center-to-out phase encoding order, repetition time per slice = 8.00s). The number of slices obtained per scan varied between 15 to 21 slices. The B_1^+-mapping process was repeated for $M = 16$ measurements using the Fourier phase encoding described in 2.2.7, yielding the $N_{\text{C}} = 8$ complex sensitivity maps.

The data for the off-resonance maps were collected with a 3DEGRE sequence (see figure 1, nominal FA = 8°, TR = 30.0ms, TE_1 = 1.00ms, TE_2 = 2.98ms, FOV = $200 \times 200 \times 176$ mm^3 (anterior-posterior $\times$ right-left $\times$ head-feet), voxel size = 4 mm isotropic, first and second phase-encoding directions AP and LR, respectively, total scan time = 1min 49s). Phase unwrapping was performed by an implementation of the algorithm presented in [12]. A shimming routine corrected the maps with all spherical harmonics up to and including order $L = 3$, for a ROI defined by a brain mask, see the next paragraph. During pulse designs, the off-resonance maps were interpolated to the space of their associated complex B_1^+-maps using a linear interpolation method from MATLABs standard library[41].

Brain masks were constructed using the Statistical Parametric Mapping (SPM) software[42], where a reconstructed magnitude image from the DREAM data and 3DEGRE data, respectively, was used for template matching. The magnitude image from the DREAM data was reconstructed by the sum-of-squares over all read-channels and measurements of the FID, while the magnitude image from the 3DEGRE data was reconstructed as the square-rooted element-wise product of the magnitude images of the two echos, each of which were reconstructed by the sum-of-squares over all read-channels. Both magnitude images were subsequently filtered through a minimum-5%-of-maximum-intensity threshold filter before being fed into SPM for brain mask construction.

3.2 Within-Volunteer Grouping of Anonymized Data

All the 17 head scans provided for this thesis were completely anonymized, and needed to be grouped on a volunteer-by-volunteer basis before they could be used to ensure proper separation during creation of data sets for Deep Learning. The separation was based on the calculated PCC (see section 7.4) between all DREAM and 3DEGRE intensity images, respectively, the same of which were constructed and threshold-filtered for the brain mask construction process. Before the PCC was calculated, the DREAM and 3DEGRE intensity images were all processed with SPM – the images were re-aligned by a rigid-body transformation to the same orientation as a reference scan and subsequently re-sliced to it. The reference scan was chosen as one of the 17 scans which was deemed (by eye) to have little-to-no artifacts in its two intensity images, and where the head looked to be nicely centered. After the reorientation, all images were threshold-filtered once more at a minimum-0.01%-of-maximum-intensity threshold to reduce artifacts arising from the re-slicing process. The use of the more sophisticated method of image matching by estimating the relative mutual information[43] between the images was attempted, but the method was deemed unable to discern the relatively small differences in the images needed to separate them properly.

Two images were classified as being of the same volunteer (i.e. a *match*) if their PCC was $r > r_0$. We here set the correlation threshold $r_0 = 95\%$. The matches were tracked in *matching matrices*. To validate the classification, within the DREAM and 3DEGRE images, respectively, the PCC was first cross-checked to further validate the classification – for each image, the images being compared which yielded a match were checked to *only* yield a match between the other images matching with said image. *This way, each group of matched images uniquely identified a volunteer.* As an example, say we have three images: image A, image B and image C. If, for instance, image A is matched with image B but not with image C, then to properly pass this first cross-check, image B should only match with image A and not image C, while image C should match with neither image A nor image B.

The matching result after this first cross-check was further validated by a second cross-check between the DREAM and 3DEGRE matching matrices, to see if they gave the same volunteer discernment. To pass the second cross-check, both matching matrices had to be equal. Any failure to pass the second cross-check was resolved with inspection by eye. The threshold $r_0 = 95\%$ and the threshold of the post-re-slicing filter were adjusted to properly pass the first cross-check, and give reasonable results in the second cross-check – too low thresholds gave non-unique groupings, too high thresholds were deemed too strict in the grouping process. See figure 7 for a flow-chart for the discernment process.

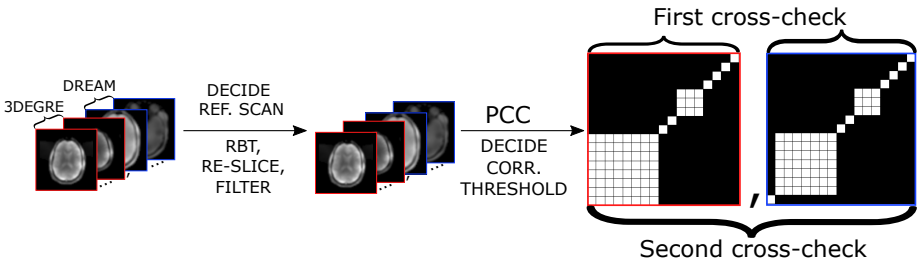
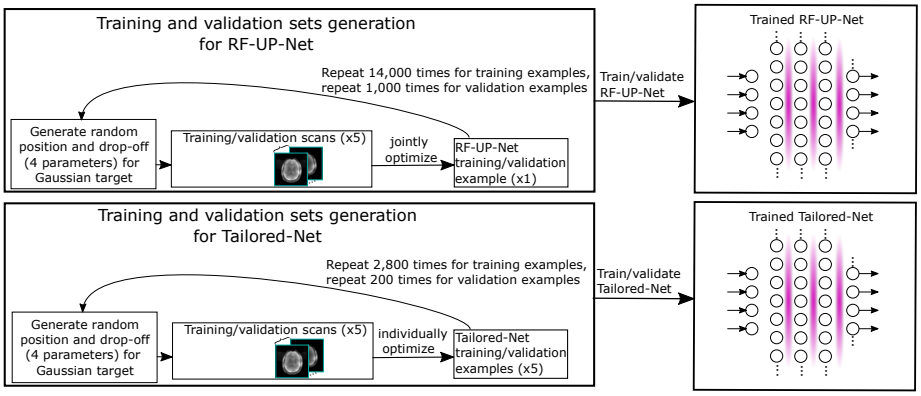


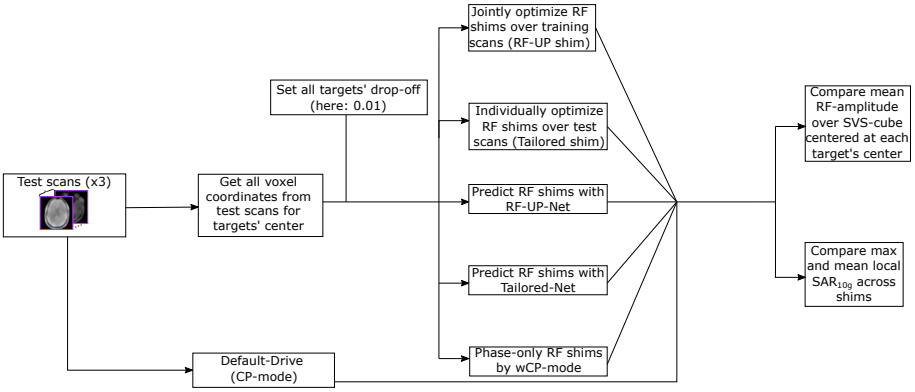
Figure 7: A flow-chart of the within-volunteer grouping of anonymized data based on intensity image from the DREAM and 3DEGRE data, illustrated respectively by one of their transversal slices. Starting from the intensity images, one decides a reference scan to reorient the other images. One performs a rigid-body rotation (RBT) and re-slicing to the reference images for all images, and subsequently threshold filter after re-slicing. The re-sliced images are then compared by calculating their Pearson Correlation Coefficient (PCC), i.e. r, to the other images within the same data group (i.e. DREAM or 3DEGRE), and setting a matching threshold r_0. The two cross-checks (see main text) are done on the two matching matrices, corresponding to $r > r_0$ for the DREAM or 3DEGRE data, respectively. The matching matrices shown pass the first cross-check, but fail the second, as both matching matrices give unique discernments, but fail to be identical.

3.3 Regressional MLP Networks for RF-shimming

Two MLP networks were trained for prediction of universal PTx-weights for RF-shimming, with targets consisting of spherically symmetric 3D Gaussian shapes of unit intensity centered at an arbitrary position and spatial drop-off. The workflow of the entire process is summarized in figure 8.



(a) Training/validation of the networks



(b) Testing and comparing the different RF-shimming methods

Figure 8: Workflow for the RF-shimming by fully-connected MLP neural networks. Figure (a) shows how the training- and valdiation sets were generated, and figure (b) shows how the trained networks were tested and compared. Weighted CP-mode is denoted wCP-mode.

The intent was to investigate the feasibility of training neural networks to essentially operate as an "interpolated look-up table" for universal RF-shimming settings to move the concentration of B_1^+-amplitude to an arbitrary location relative to the lab-frame, and to compare their performance to that of their corresponding pre-calculated RF-UPs and volunteer-tailored RF-pulses. This kind of concentrated pulses could be beneficial for e.g. Single-Voxel Spectroscopy. The reasoning behind the choice of Gaussian targets lies in the nature of the system and its limited degrees of freedom, as the targets qualitatively replicate the field from CP driving mode, shifted around the ROI, and were able to be sufficiently mimicked by the B_1^+-amplitude from MLS RF-shimming. For a

reference comparison, the B_1^+-map from default driving mode (CP-mode, all PTx weights set to have equal amplitude and no phase shift), and *weighted* CP-mode (all PTx weights set to have equal amplitude and phase shift weighted to give constructive phase-interference at the desired location for the amplitude concentration) was simulated. Thus, weighted CP-mode corresponds to a *phase-only shim*. Note that as with the tailored RF-pulse, weighted CP-mode required full sensitivity information of each channel, which will become important in later discussion.

Both networks shared the same architecture and were trained with the same training parameters – the network architecture is shown in figure 9 and the applied training parameters were as is detailed in table 1 (both networks were trained with SGDM). The training parameters were tuned s.t. further extending the number of epochs yielded no further decrease in the cost function while indicating little-to-no overfitting relative to the validation set in both networks. The training set and validation set were generated by randomly selecting the targets' center and drop-off, calculating the PTx-weights by solving eq. (18) with mCGLS and the local variable exchange method presented in section 2.2.2 for said targets, and choosing the weights' L_2-norm trade-off by the L-curve approach.

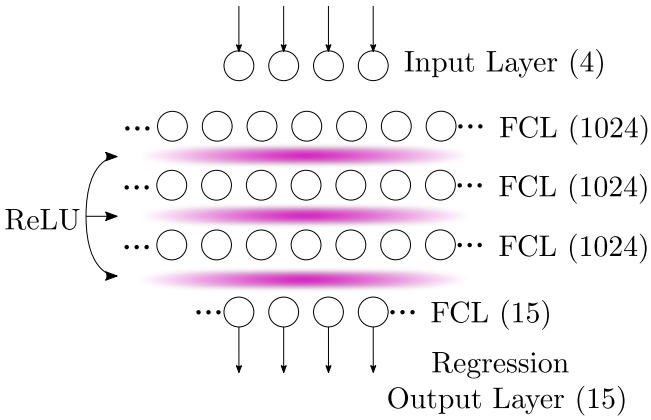


Figure 9: A visual representation of the fully-connected network architecture used for RF-shimming. Horizontal ellipsis indicate more nodes than indicated by the figure. Weights have been left out in the illustration for the sake of clarity, however, every single node in the hidden layers is fully connected to all nodes in its adjacent layers, similar to that in figure 5. The network consisted of an input layer with 4 input nodes (i.e. the 3 spatial coordinates and the drop-off of the desired $|B_1^+|$-hotspot), and a repeated structure of 3 FCLs and ReLU activation layers was used for the deep part of the networks, with each FCL consisting of 1024 nodes. As the size of the FCL preceding the output layer needed to be the same size as the output layer itself, a FCL of size 15 was added ensuing the 1024-layers. The output layer's 15 responses constituted the RF settings, i.e. the predicted weights.

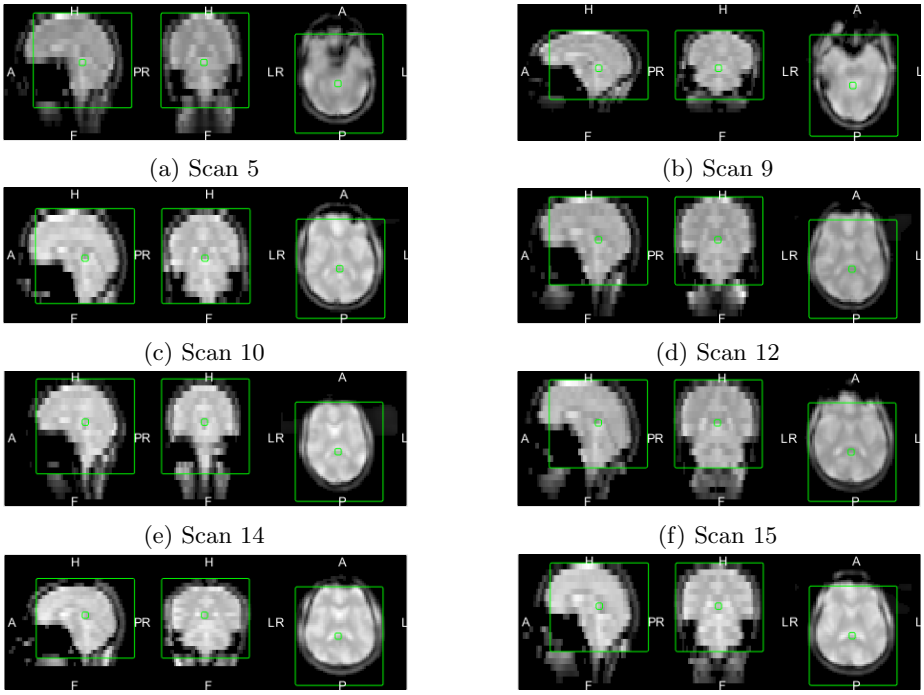
Table 1: Training parameters for the SGDM-algorithm, shared by both networks trained for RF-shimming.

Parameter	**Value**
Momentum coefficient (α_m)	0.90
L_2-regularization factor (Λ)	10^{-4}
Maximum number of epochs	100
Shuffle mini-batch	Every epoch
Mini-batch size	500
Initial learning rate (η_0)	0.3
Learning rate schedule	Piece-wise
Learning rate drop period	25 epochs
Learning rate drop factor	0.5
Validation frequency	50 epochs

The main difference between the two networks was their training and validation data sets. The first network, denoted as the *RF-UP-Net*, was trained and validated with the targets' position and drop-off as input and corresponding RF-UP PTx weights, with each RF-UP jointly optimized over 5 volunteers which discerned by the volunteer discernment process as presented in 3.2, i.e. with the data from scan numbers $4, 9, 10, 12, 14$ in figure 15. Thus, the RF-UP-Net was trained on a training set with a guaranteed 1-to-1 correspondence between input and output and was taught universal pulse settings directly from the training set. The latter network, denoted as the *Tailored-Net*, was trained and validated with the targets' position and drop-off as input and corresponding tailored PTx weights as output, individually optimized to each of the same 5 volunteer as for the RF-UP-net. In other words, there were at least 5 training examples in the training and validation sets, respectively, sharing the same input value (i.e. target hot-spot center and drop-off), but with their own unique output (i.e. tailored PTx-weight coefficients). Thus, Tailored-Net was trained on a training set with an (at least) 1-to-5 correspondence between input and output, and found universal pulse settings by finding the best compromise (i.e. the network parameters which minimized the network' objective function) over the training set during training. This is the reason behind the choice of the relatively large mini-batch size of 500 used for training the networks, as many training examples was deemed necessary to properly find a decent compromise at each gradient calculation.

The networks' input consisted of 4 parameters, i.e. 3 spatial coordinates for the center of the hot-spot and 1 for its spatial drop-off. The inputs' scalar values for the spatial coordinates used for training (but before normalization) were in the range of $\pm 0.72 \cdot \mathrm{FOV}_d/2$, where FOV_d is the FOV in the $d = \mathrm{HF}, \mathrm{AP}, \mathrm{RL}$ directions of the DREAM sequence (see section 3.1) in units of meter. The factor of 0.72 was chosen as the coordinates made all targets sufficiently cover the

heads of the volunteers used for generation of the training and validations sets, while sufficiently minimizing the amount of Gaussian targets which were centered outside any of said heads (moving the center of the target outside a head is not desired). See figure 10 for an illustration. The spatial drop-off values were in the range of $[0.01, 0.04]$, lower values giving a more rapid drop-off. The range for the drop-off was chosen as the targets with these drop-offs were deemed large enough to be sufficiently replicable by RF-shimming, without getting targets which were homogeneous throughout the ROI (which would correspond to whole-brain homogenization of the B_1^+-field instead of concentrating it). See figure 11.



(a) Scan 5 (b) Scan 9

(c) Scan 10 (d) Scan 12

(e) Scan 14 (f) Scan 15

(g) Scan 16 (h) Scan 17

Figure 10: Magnitude images from the 8 scans (from different volunteers as decided by the discernment process) used for generating tailored pulses and UPs for training, validation and testing of Tailored-Net and RF-UP-Net, respectively, shown for three perpendicular slices which intersect at the origin of the lab-coordinates. Scans $4, 9, 10, 12, 14$ were used for generation for the training and validation sets, scans $15, 16, 17$ were used for generation of the test sets. These scan numbers are as indicated in figure 15. The lab-origin is indicated by the inner, smaller green circles. The green boxes indicate the volume within all the target hot-spots were generated for the training and validation sets, i.e. $\pm 0.72 \cdot \mathrm{FOV}_d/2$ relative to the lab-origin, where FOV_d is the FOV in the $d = \mathrm{HF}, \mathrm{AP}, \mathrm{RL}$ directions of the DREAM sequence (see section 3.1) in units of meter. Note that all positions for the Gaussian targets' center for the test set were chosen to be within the brain according to the SPM brain masks, instead of random positions within the green box – the boxes are included for the scans used for generation of the test sets to show what volume was considered during training and validation.

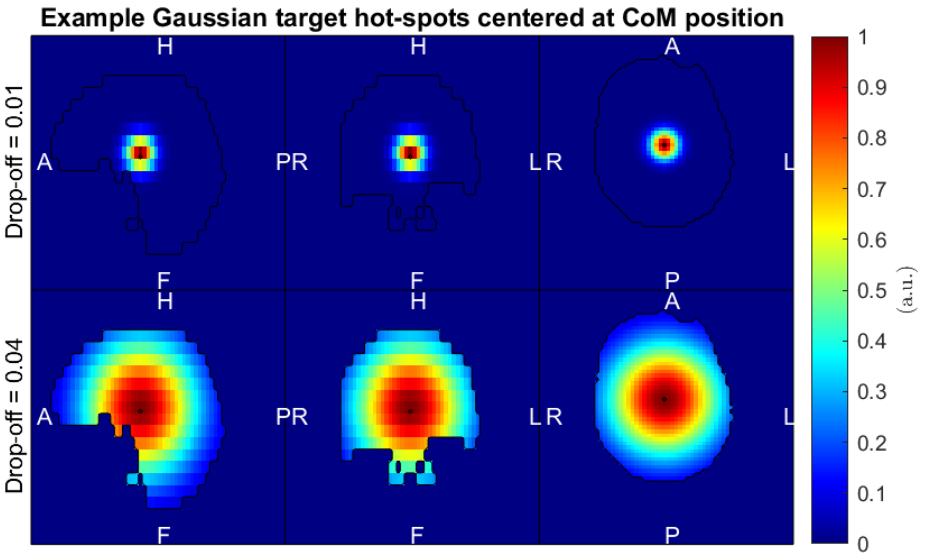


Figure 11: Two Gaussian hot-spot targets shown with their fastest and slowest drop-offs, 0.01 and 0.04, respectively, for targets centered at a calculated center-of-mass (CoM) position. The drop-off values corresponds to the distance (in meters) from mask center for which the target has an intensity of $\exp(-1^2) \approx 37\%$ of its maximum. Here, the brain mask of scan 1 (see figure 15) was used for illustration.

The networks' output consisted of 15 responses, corresponding to the real and imaginary part of all the 8 individual channels' weight[10], respectively, neglecting the first channel's imaginary part (which was always subsequently set to zero) due to its outset zero-phasing. Under the STA approximation, the channels' amplitude can be scaled arbitrarily by the RF-pulses' waveform, and therefore all network outputs were all in the range $[-1, 1]$ during training, such that the weights' amplitude were the relative scaling of their associated channel's amplitude.

The training set for both networks consisted for 14k training examples. The training performance was evaluated during training by the root of the mean square error (RMSE) between the true and network-estimated responses in each iteration, and further validated by a separately, randomly generated validation set during training of 1k examples, optimized over the same volunteers and with target center positions and drop-off generated in the same manner as the training set. A similar procedure was performed for generating the test set, but the positions were chosen to cover all of the voxel's center coordinates within the SPM brain mask, and the drop-off was set to 0.01 for all targets. In this way, the different shim methods were tested on all voxel positions within the brain mask, without attempting to shim to a target whose center lied outside the brain and with a very concentrated target. The main evaluation of the shim performance of the different methods consisted of comparing three different metrics across the shim methods for a given target (position and drop-off):

[10]Training on the standard form of the weights instead of polar form was done to avoid the problem of discontinuities due to phase-wrapping.

1) The mean RF-amplitude (i.e. $\left|B_1^+\right|$) over a collection of voxels forming a cube with volume of (approximately) 2x2x2cm^3, with the cube's center voxel coinciding with the Gaussian target's center voxel. As the desire for e.g. SVS is to achieve the highest RF-amplitude within a small volume such as that cube, it will be referred to as an *SVS-cube* for brevity.

2) The estimated maximum local $\mathrm{SAR}_{10\mathrm{g}}$ over all voxels for which the Q-matrices are calculated.

3) The estimated mean SAR of the head, i.e. the $\mathrm{SAR}_{10\mathrm{g}}$ averaged over all voxels for which the Q-matrices are calculated.

We assumed a square (block) RF-pulse is used[11], i.e. setting the waveform as in eq. (21), with the normalized waveform set to unity during RF-pulsing, zero otherwise. The pulse length can be inferred after setting the repetition time for a given sequence and thus deciding the desired RF duty-cycle. The reasoning behind this choice lies in the fact we can now compare the estimated efficiency of each shim method *prior* to setting $V_{\max}$ – from eqs. (14) and (23), respectively, we see with the given choice of waveform,

$$\left|B_1^+\right| \propto V_{\max} \text{ and } \mathrm{SAR}_{\mathrm{pulse}}(\mathbf{r}) \propto V_{\max}^2. \tag{28}$$

Thus, the results for the RF-amplitudes and SAR-levels can be adapted to be investigated prior to choosing $V_{\max}$.

[11]The argument presented here can be adapted to calculate the SAR for any pulse shape (e.g. sinc) by scaling the results by the square of the sampled waveform, see [29].

3.4 Regressional CNN for k_T-point FA homogenization

A regressional CNN was trained for the prediction of PTx-weights for an 8-k_T-point trajectory, with the goal of investigating how the predictions from a network trained and verified with a very small number of volunteer data (13 and 2 scans, respectively) compared with volunteer-tailored pulses (i.e. time-varying PTX-weights found by eq. (20)) and a k_T-UP optimized over 5 discerned volunteers (data from the same 13 examples used for network training, but only one scan for each discerned volunteer), with the goal of achieving the highest FA homogeneity across the brain for. All pulse methods were tested on data from 2 discerned volunteers previously neither seen by the network during training nor during the k_T-UP optimizations. The workflow is summarized in 12.

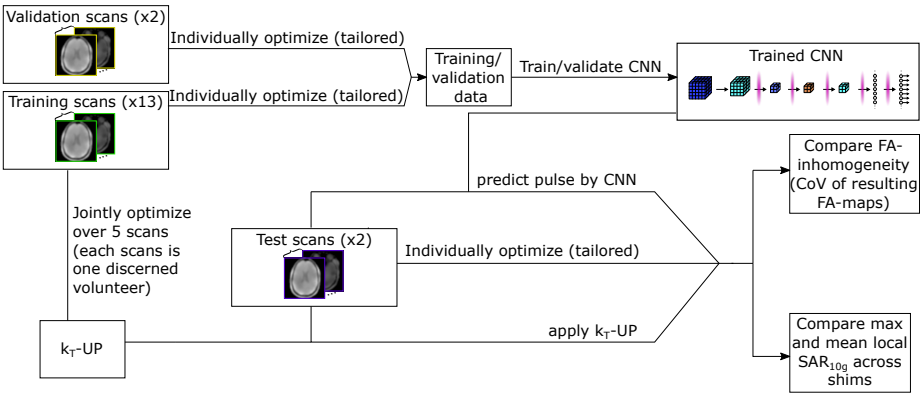


Figure 12: The workflow for the prediction of the PTx-weights for the 8-k_T-point trajectory using a convolution neural network (CNN). The data from the training-, validation- and test scans consisted of their associated (shim-corrected) off-resonance B_0-maps and complex B_1-maps during optimization. The network training and predictions only used the $|B_1^+|$-map from default drive (CP-mode) with $V_{\max}$ set to unity. The performance of the pulse settings predicted by the CNN on the test scans (whose data was not seen during training) was compared to the performance of the pulse tailored to the test scans, as well the performance of the universal pulse (k_T-UP) which was jointly optimized using the data from the training scans. The comparison was made based on the FA-inhomogeneity, measured as the coefficient of variance (CoV) of the FA-map resulting from all the pulse settings, respectively, as well as their assoicated SAR-efficiency, measured by each pulse settings estimated maximum local SAR_{10g} and local SAR_{10g}, meaned over all voxel for which the Q-matrices were calculated.

The 8-k_T-point trajectory used is shown in figure 13. The trajectory was designed to visit k-space locations at $\pm 6.33\mathrm{m}^{-1}$ in the $d = \hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ directions in Cartesian coordinates, with its first and final point at $\mathbf{k} = 0$. $\pm 6.33\mathrm{m}^{-1}$ is a rough underestimate of the wavelength of RF in tissue at 7T[12].The total pulse duration was $T_p = 1.12$ms, with each of the 8 rectangular sub-pulses lasting 80µs, interleaved by 60µs gradient blips (i.e. trajectory traversals).

[12]The choice of k-space distances was chosen according to a rough estimate of the wavelength of RF in tissue at 9.4T, but was kept as changing it had little effect on the results.

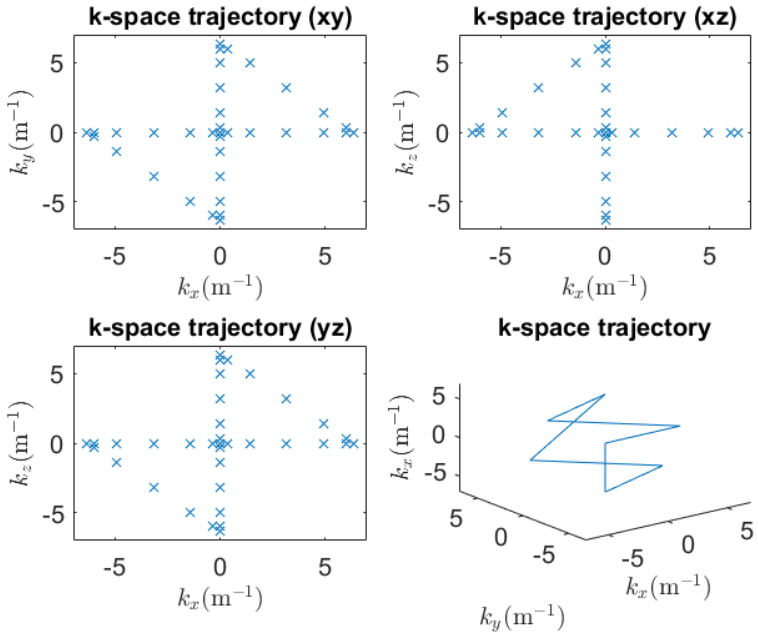


Figure 13: The 8-k_{T}-point used in the whole-brain FA homogenization for all designed pulses, and its projections onto the xy-, xz- and yz-planes. The trajectory was designed to visit k-space locations at $\pm 6.33\mathrm{m}^{-1}$ in the $d = \hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ directions in Cartesian coordinates, with its first and final point at $\mathbf{k} = 0$. The k-space velocity is not indicated here, however the trajectory stops while transmitting RF at its corners or at the origin (similar to that presented in figure 3.)

The network's input data was the 3-D RF-amplitude map (i.e. $|B_1^+(\mathbf{r}_n)|$) of $56 \times 64 \times 21$ voxels resulting from driving the PTx-system in default drive and $V_{\max}$ set to unity, with the network's output being the (time-varying) PTx-weights settings for each of the 8 sub-pulses across all 8 transmit channels, constituted by the network's 127 outputs (real and imaginary part of all weights, with the first weight's imaginary part being set to zero from outset). The network architecture is shown in figure 14. The network was trained with the training algorithm *Adam*, see the solver-specific parameters applied as listed in table 2. Adam was chosen as it showed to be more robust to overfitting while the network hyper-parameters were adjusted. As the 3-D RF-amplitude maps were constructed from a number of stacked transversal slices, varying between 17 and 21 slices for each scan, the 3-D maps were augmented by stacking additional slices at the top and bottom slices until 21 slices were reached. The main assumption behind the choice of network input was that the information contained in said RF-amplitude maps was sufficient for the network to adequately predict time-varying weights. The justification for this assumption was that information of the transmit channels' sensitivity and their interference patterns was implicitly contained in the RF-amplitude maps, inspired by the method of which each channel's sensitivity map were measured (i.e. the method presented in section 2.2.7). The targets used for all optimizations of the pulse settings contained in the training-, validation- and test set for the network, as well as for the

k_T-UP and tailored pulses, was a binary brain mask calculated by SPM, i.e. a target which was zero outside the brain and homogeneous inside. The L_2-norm trade-off on the weights was decided using the L-curve approach. All scans of the same (discerned) volunteer was included in the training- and validation sets to exhaust the amount of data available, i.e. including as many examples in the sets as possible by including the data from all scans 1-15 (scans 16-17 were reserved for the test set). However, the sets were attempted to remain separated within volunteers during training, validation and testing by ensuring that a discerned volunteer was not included across the sets. The scans included in the optimization of the training- and validation sets were scans 1-13 and 14-15, respectively, see figure 15.

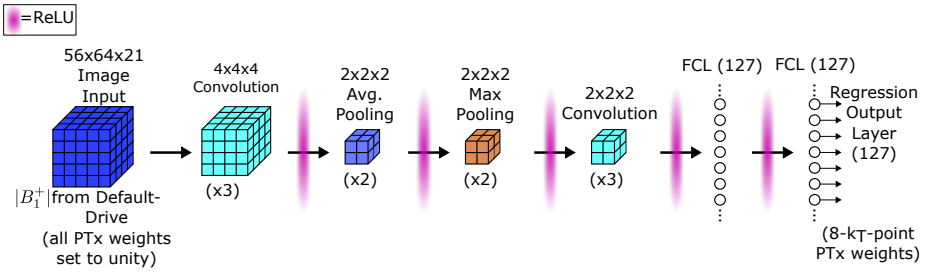


Figure 14: The architecture of the CNN used for the prediction of the time-varying weights used for whole-brain FA homogenization in the 8-k_T-point trajectory. The choice of architecture was based on the idea to have a small fully-connected structure learning on a down-sampled version of the RF-amplitude map.

Table 2: Training parameters used in the training of the regressional CNN by the Adam-solver[38].

Parameter	**Value**
Gradient decay factor (β_1)	0.90
Squared gradient decay factor (β_2)	0.99
Offset factor (ϵ)	10^{-8}
L_2-regularization factor (Λ)	10^{-4}
Maximum number of epochs	50
Shuffle mini-batch	N/A
Mini-batch size	13 (all)
Learning rate (η)	0.3
Learning rate schedule	None
Validation frequency	1 epoch

The main evaluation of the three different pulse designs methods, i.e. k_{T}-UPs, tailored pulses and network-predicted pulses, consisted of comparing three different metrics across the methods:

1) The coefficient of variance (CoV) of the FA-map for a given pulse setting – this is a scale-invariant measure of the inhomogeneity of the FAs. The CoV was calculated by dividing the standard deviation of the FAs over the voxels contained in the SPM brain mask by their mean.

2) The estimated maximum (local) $\mathrm{SAR}_{10\mathrm{g}}$ over all voxels for which the Q-matrices were calculated.

3) The estimated mean SAR of the head, i.e. the $\mathrm{SAR}_{10\mathrm{g}}$ averaged over all voxels for which the Q-matrices were calculated.

Here, we made the same assumption on the applied RF-pulse as for RF-shimming, i.e. we set its waveform as in eq. (21). We also note that the normalized waveform was a train of 8 block sub-pulses with unity amplitude – the sub-pulse of each channel was therefore modulated by its respective complex weight, each of which were normalized to an amplitude between zero and unity. The weights' amplitudes were thus the relative scaling of $V_{\max}$ across all sub-pulses and channels.

4 Results

4.1 Volunteer Discernment

The volunteer discernment process presented in 3.2 yielded $N = 8$ unique volunteers. The matching matrices for the DREAM and 3DEGRE data are shown in figure 15 for the correlation threshold $r_0 = 95\%$, where scan number 10 was used as reference for the rigid-body transformation (RBT) and re-slicing.

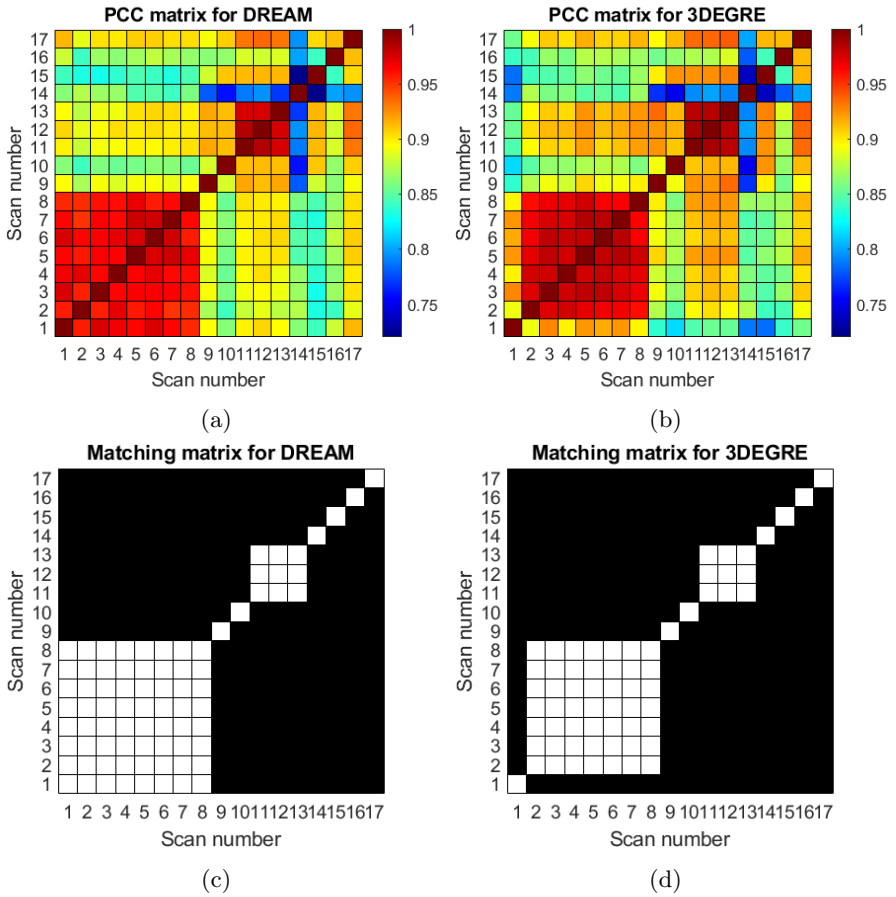
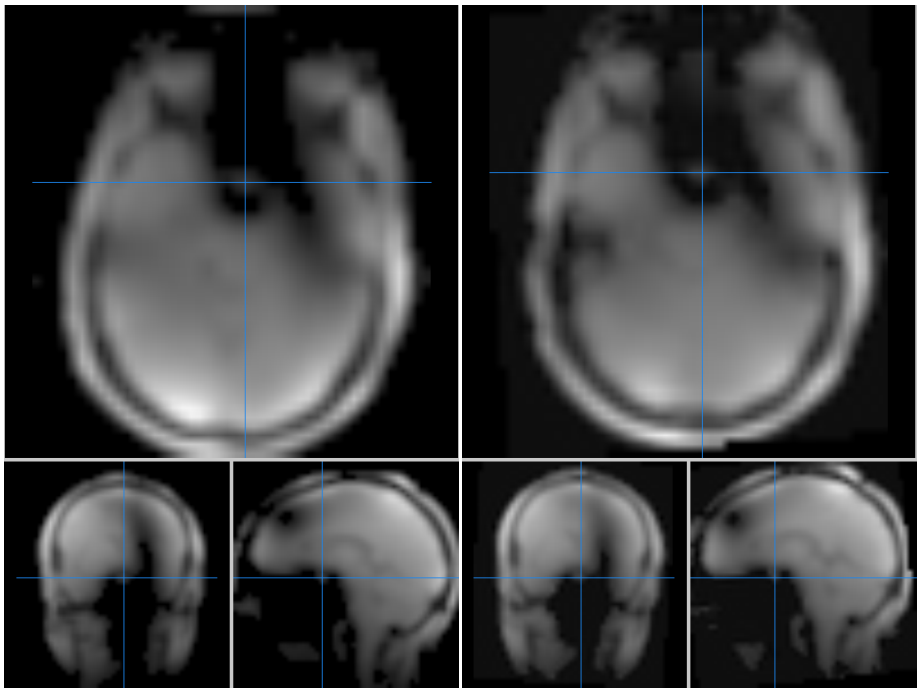


Figure 15: Pearson Correlation Coefficient (PCC) matrices in (a) and (b), and matching matrices in (c) and (d) at threshold $r_0 = 95\%$ with white tiles indicating matches. Scan number 10 was chosen as reference for the rigid-body transformation (RBT) and re-slicing. Note that both matching matrices pass the first cross-check, but fail the second due to their discrepancy in scan number 1. The discrepancy was solved by inspection by eye, and (c) shows the final discernment used for this thesis, i.e. $N = 8$ unique volunteers.

The discrepancy in scan number 1 between the two data sets is attributed to artifacts from the re-slicing process after the RBTs. The 3DEGRE intensity image did not contain the entire head within its FOV, which lead to cut-offs in the re-sliced image after filtering. See figure 16. The discrepancy was solved with

inspection by eye, and scan number 1 was deemed to be of the same volunteer as in scan number 2, i.e. figure 15c shows the final discernment used for this thesis.



(a) (b)

Figure 16: Scan number 1 (see figure 15) before and after the re-slicing and filtering in (a) and (b), respectively, for (approximately) the same slices in the transversal, coronal and sagittal plane. The blue cross-hairs indicate where the slices intersect. Note how the far posterior position is not within the FOV in (a), which yields a zig-zagged cut-off at the corresponding area in (b).

4.2 RF-shimming and MLP Performance

An example of the SVS-cube-means for an arbitrary location of the target is shown in figure 17. The same procedure repeated for all voxel locations over scans 15, 16 and 17 (scan numbers relative to figure 15) is shown in figure 18. The calculated distributions are shown for slices in the HF-, AP- and RL-planes which intersect at a calculated center-of-mass position. For the SAR-calculations, a 1% RF duty-cycle was assumed (changing the RF duty-cycle does not change the relative scaling of the achievable SVS-cube-means). The training/validation progress of RF-UP-Net and Tailored-Net is shown in figure 19.

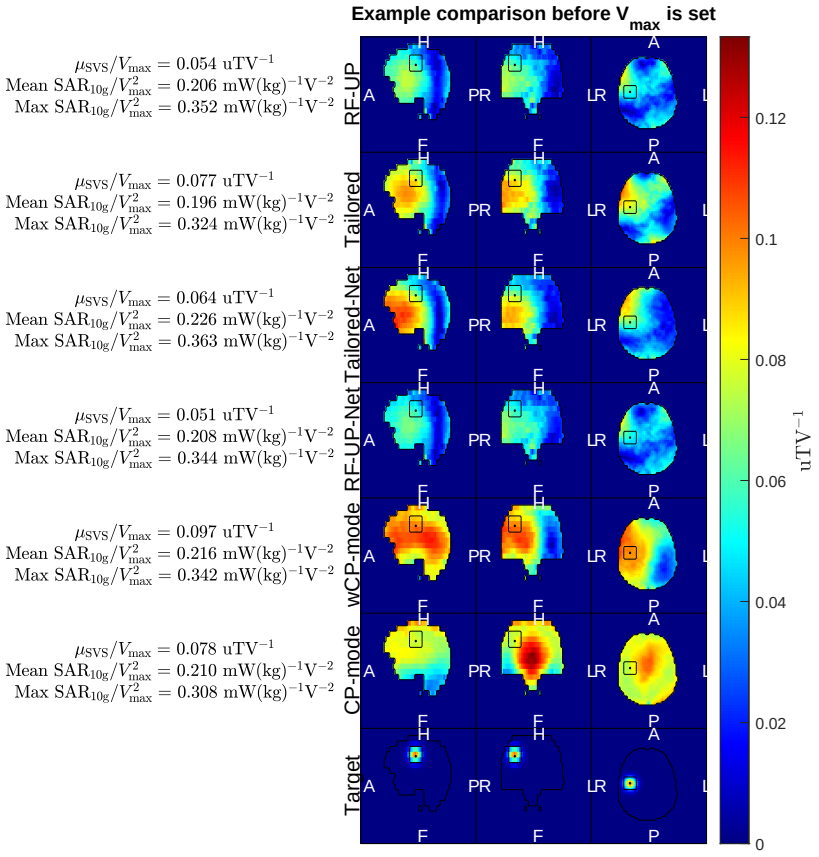
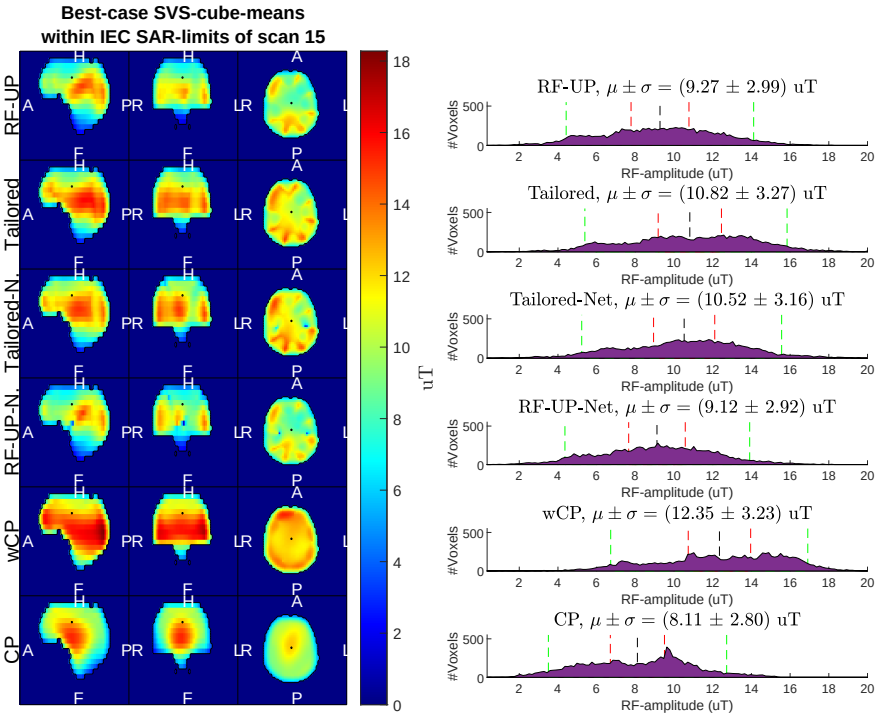
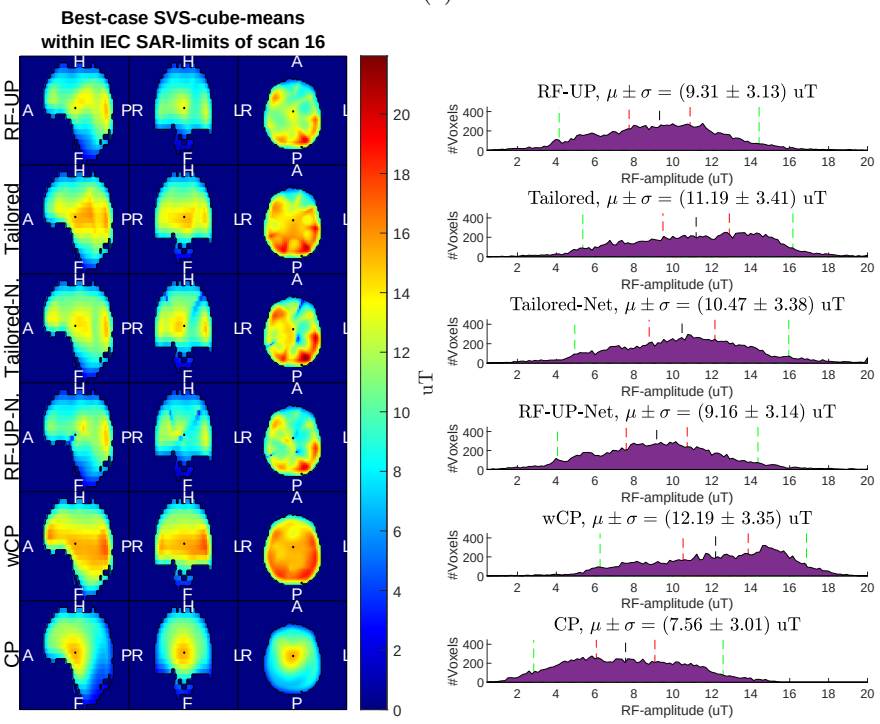


Figure 17: An example comparison of the different method for a given Gaussian target's location and drop-off (drop-off set to 0.01, target shown in the bottom-most row), with the resulting SVS-cube-mean (μ_{SVS}), and mean and maximum SAR_{10g} for the different configurations *prior* to setting V_{max} (see eq. 28). Here, data from scan 15 (see figure 15) was used. After V_{max} has been set, these maps correspond become physical. The maximum (physically achievable within regulatory SAR-limits) SVS-cube-means for each shim configuration is restricted by the estimated associated low maximum- and mean SAR_{10g} of each shim configuration. The results from RF-UP, tailored, Tailored-Net, RF-UP-Net, weighted CP-mode (wCP-mode) and CP-mode are shown. The black cube within the brain-mask and its central dot represents the SVS-cube and its center, respectively.

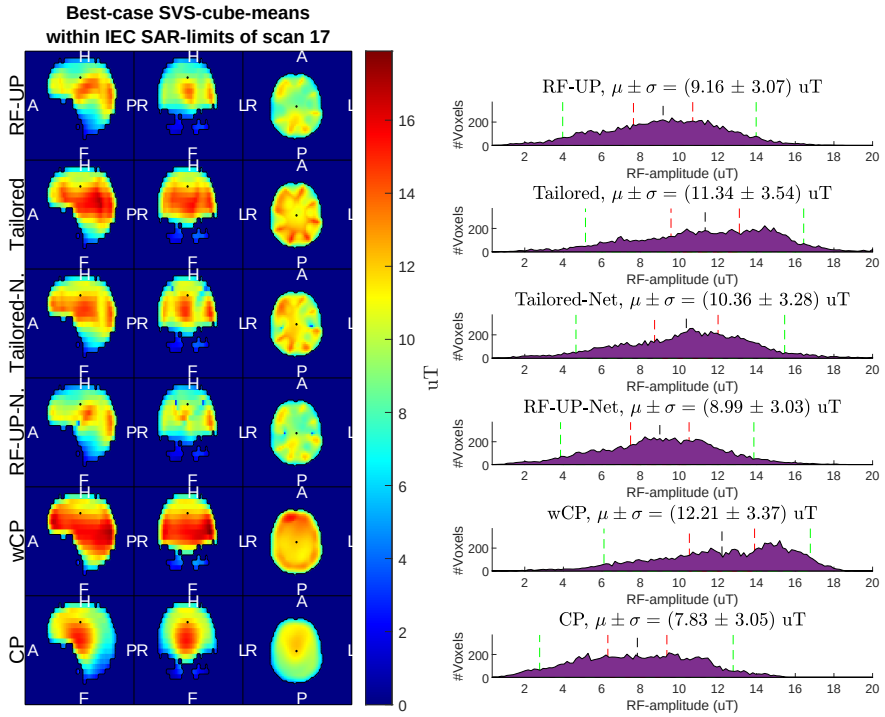


(a)



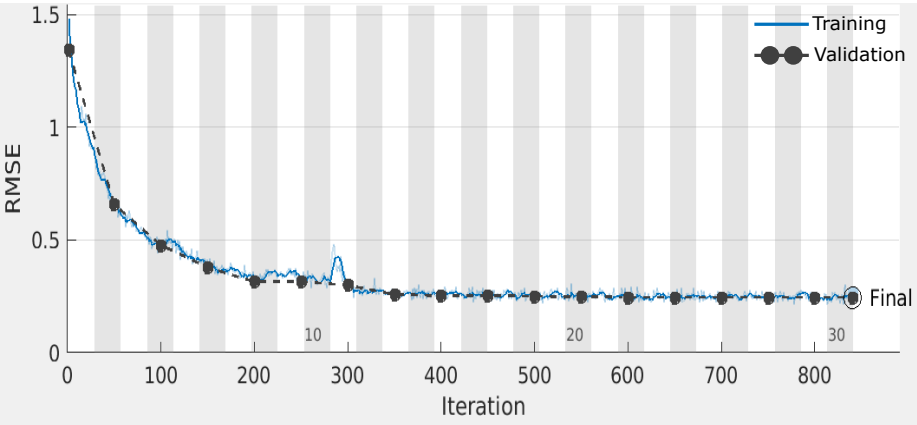
(b)

Figure 18: *Continued, see the next page for figure details.*

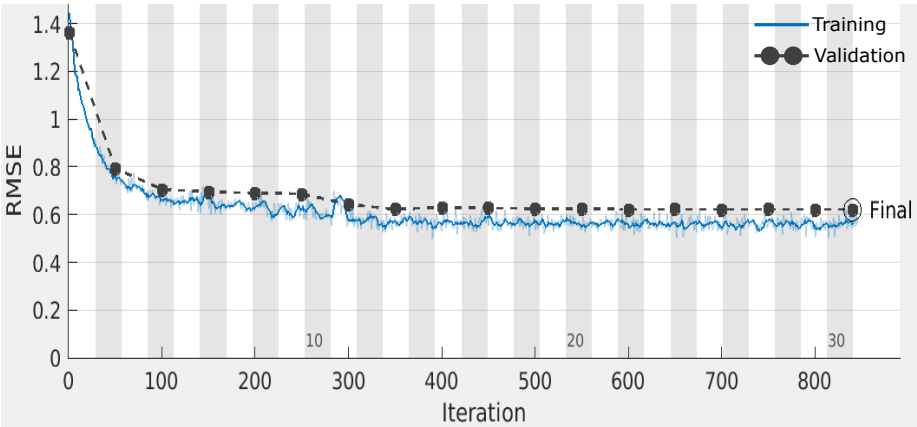


(c)

Figure 18: The distribution of the best-case SVS-cube-means from the RF-shimming methods applied to the data from scan 15, 16 and 17 in (a), (b) and (c) respectively – the scan numbers are as indicated in figure 15. An important remark is that the maps and histograms shown are *not* physical. *Each voxel value indicates the mean RF-amplitude over the SVS-cube centered at said voxel, with* $V_{\max}$ *set to reach either the max or head-averaged local* SAR_{10g} *limits* (10W/(kg) *and* 3.2W/(kg)*, respectively), whichever* $V_{\max}$ *is lowest (the SAR-limits used here are those recommended by the International Electrotechnical Commission (IEC)*[44]). *That is, each voxel represents a unique shim configuration targeted at maximizing the RF-amplitude over the SVS-cube centered at that voxel.* The results from RF-UP, tailored, Tailored-Net (Tailored-N.), RF-UP-Net (RF-UP-N.), weighted CP-mode (wCP) and CP-mode (CP) are shown here. The histograms indicate the calculated mean RF-amplitude (μ) *of the distribution (i.e. a mean of SVS-cube-means)* by the black, stapled line, and its associated standard deviation (σ) from the mean is indicated by the red, stapled lines. The 90^{th} percentile ranges are also shown to further indicate the spread, shown by the green stapled lines.



(a) RF-UP-Net

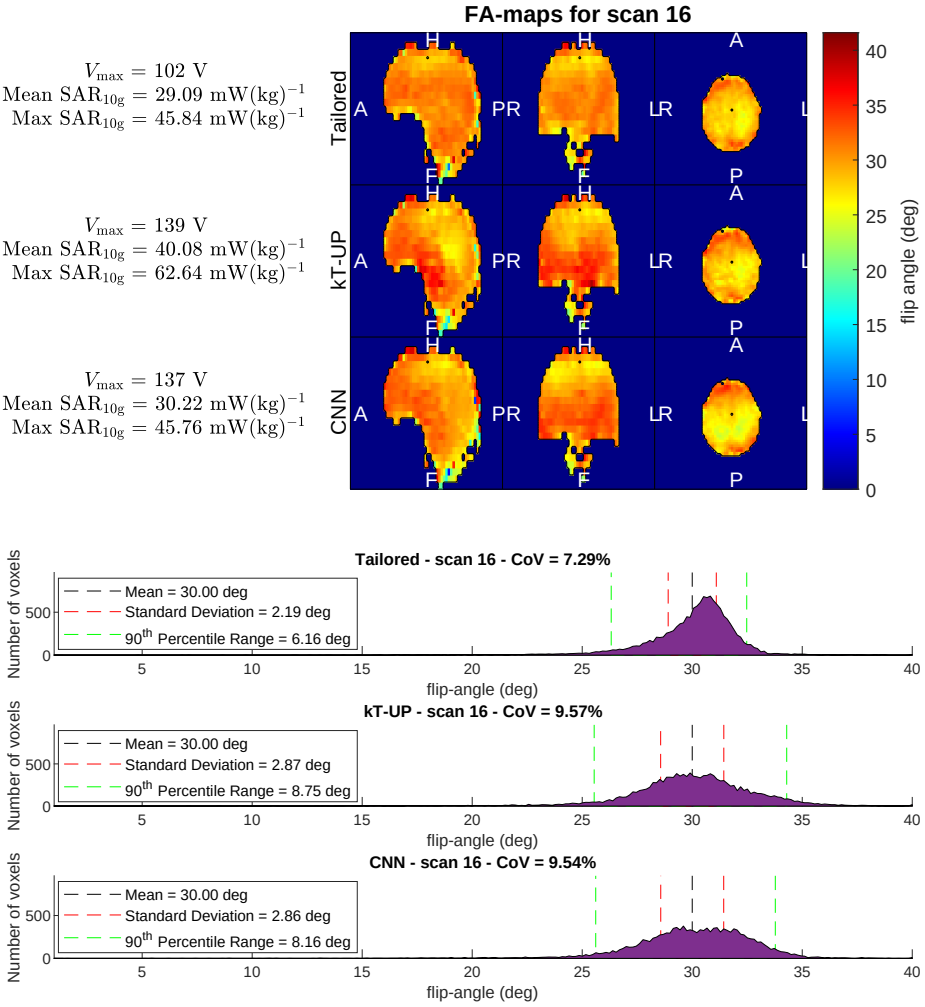


(b) Tailored-Net

Figure 19: The training/validation progress of the MLP networks used for RF-shimming.

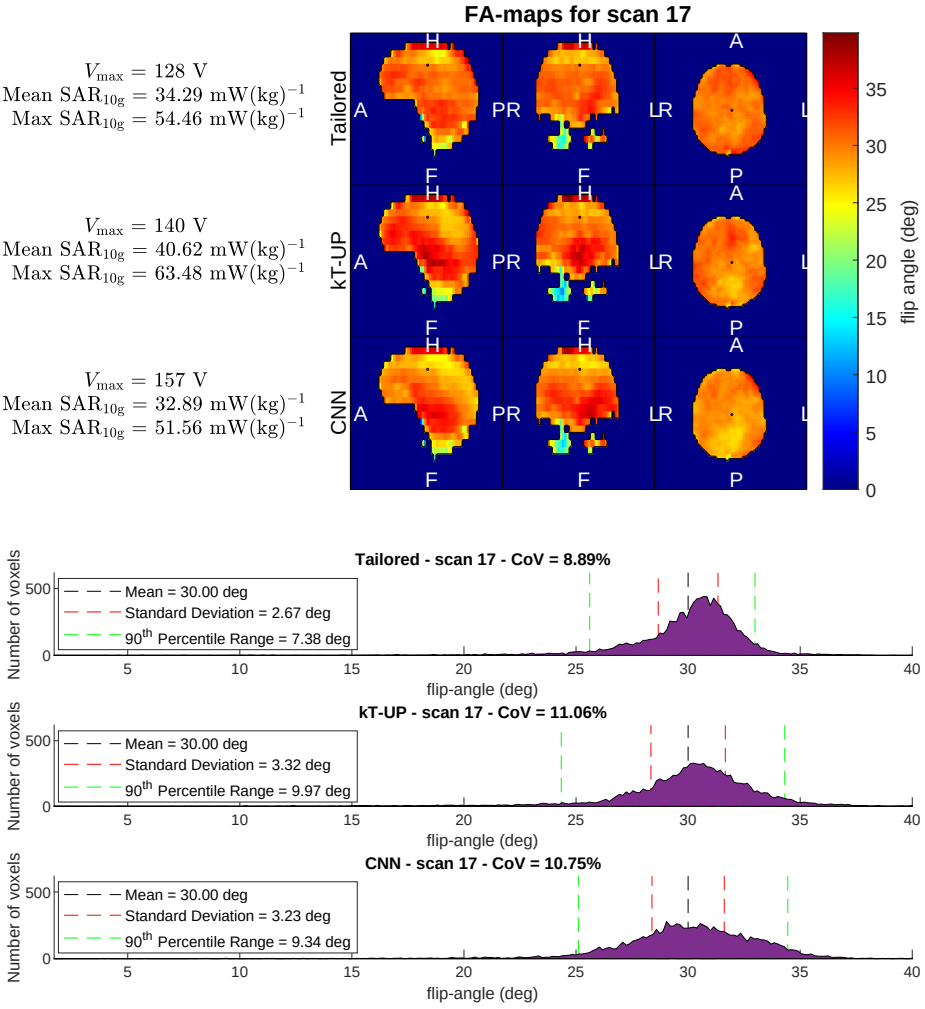
4.3 8-k_{T}-point Weight Predictions and CNN Performance

The results from the 8-k_{T}-point weight predictions are shown in figure 20, where the results from tailored, k_{T}-UP and the CNN-predicted time-varying weights are shown for comparison for scans 16 and 17 (with scan numbers as indicated in figure 15). The training progress of the CNN used for prediction is shown in figure 21, along with the raw output from the networks compared to its tailored counterpart for both said scans. The raw output is compared to give a qualitative impression of the network's performance in its predictions.



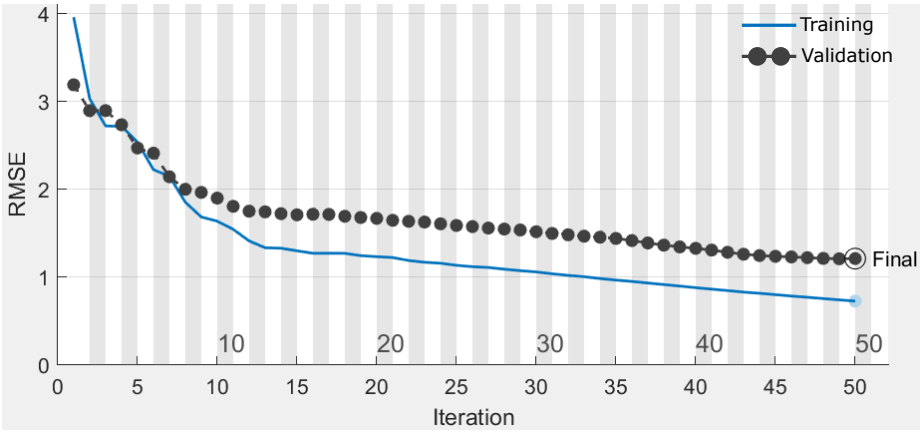
(a)

Figure 20: *Continued, see the next page for figure details.*

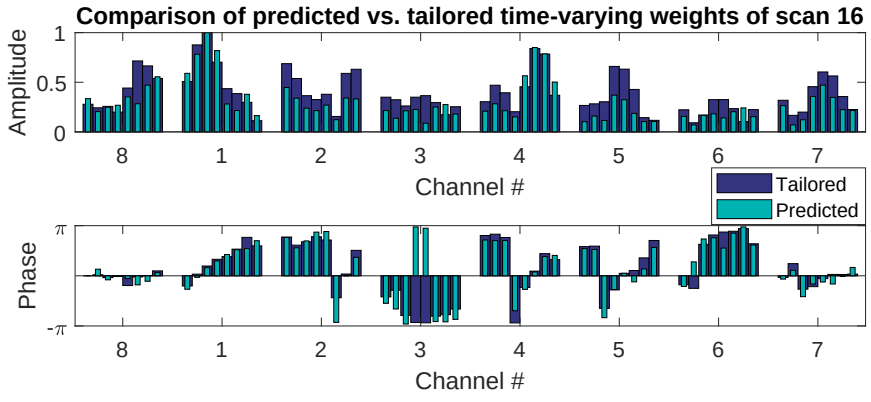


(b)

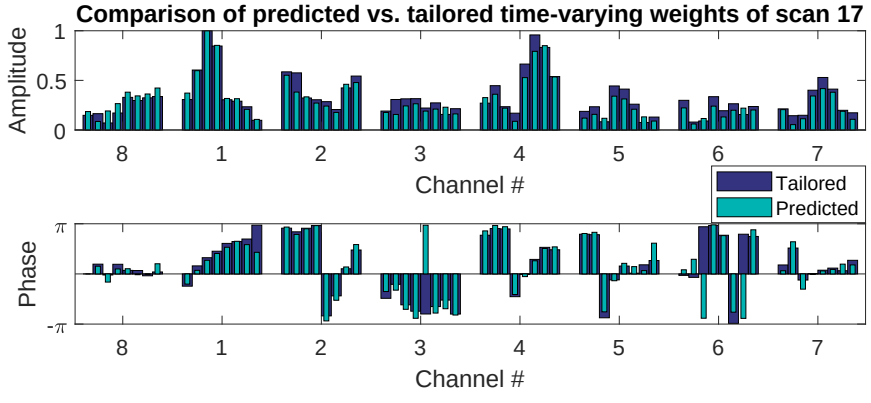
Figure 20: The FA distribution from the 8-k_T-point trajectory applied to the data from scan 16 and 17 in (a) and (b), respectively – the scan numbers are as indicated in figure 15. The results from tailored, k_T-UP and CNN-predicted time-varying weights are shown here. The calculated distribution is shown for a slices in the HF-, AP- and RL-planes which intersect at a calculated center-of-mass position, with the applied V_{max} which yields a mean FA of 30° under the STA approximation. The maximum and mean SAR_{10g} are also reported in units of mW(kg)$^{-1}$. For the SAR-calculation, we have assumed a repetition time of 1s. As the total pulse duration is $T_p = 1.12$ms, this corresponds to a 0.11% RF duty-cycle. The histograms indicate the mean flip angle, the calculated standard deviation, and the coefficient of variance (CoV). The CoV was calculated by dividing the mean by the standard deviation. The 90th percentile range is also shown to further indicate the spread.



(a)



(b)



(c)

Figure 21: In (a), the training/validation progress of the CNN used for the 8-k_{T}-point weight predictions. In (b) and (c), the network output (i.e. its predictions) and its tailored counterpart (i.e. its test set) of the test scans, shown to qualitatively assess the network's performance, i.e. the accuracy of the predicted weights. Each group of bars is the *relative* amplitude/phase of the indicated channel during the 8 sub-pulses, with the first-to-last sub-pulse ordered left-to-right in each group. Note the visualization is affected by phase-wrapping.

5 Discussion

5.1 Volunteer Discernment

5.1.1 Evaluation and Sensitivity to Head Shape and Size

The volunteer discernment as shown in figure 15 seems to indicate a clean separation in the discernment of volunteers, in spite of the artefact in scan 1 which was attributed to the rigid-body transformation (RBT) process of the 3DEGRE-data. As the PCC is purely a (scale-invariant) pixel-by-pixel-based metric, it can be very sensitive to differences in head shapes and head sizes – after the RBT and re-slicing process, any differences in either shape or size will cause big difference in image intensity along the boundary of the heads, as the "overlap" of two intensity images of heads from separate volunteers will have big discrepancies in regions where only one image has high signal, e.g. at the extremities of the largest[13] head. Another way to understand this is to realize that the PCC can be regarded as a metric which measures the amount of "jointness" in two images on a pixel-by-pixel-basis. This could be the underlying reason why the cut-off artifact of scan 1, which is shown in figure 16, caused the PCC to differ so significantly between the DREAM-data and 3DEGRE-data.

5.1.2 Effects of PCC- and Masking Thresholds

The effect of changing the masking thresholds (minimum-of-maximum-intensity) should be discussed in light of the previous paragraph. The masking threshold reduces noise in the images, but can remove regions of the head with little-to-no signal (i.e. dark areas). A consequence of choosing masking threshold which is too high is removing signal along the head boundary. It is important to ensure that the choice does not remove the regions which constitute a head's shape – if these regions' signal is removed/diminished, the true head shape can be lost/weakened in the images and can cause the accuracy of the discernment process to become lower, as the images' "overlap" contains less information of the differences in the head shape/size.

An important feature (and issue) with this process is the freedom in choosing the PCC-threshold (r_0) and the masking thresholds (minimum-of-maximum-intensity). As stated earlier, these thresholds were chosen to properly pass the first cross-check, and give reasonable results in the second cross-check – too low thresholds gave non-unique groupings, too high threshold was too strict in the grouping process. This means that the choice of thresholds was made on the grounds of the data itself, which in turn can lead to inaccurate volunteer discernment if one is not careful. During the process of choosing the thresholds, the PCC matrices themselves need to be taken into account, and not only the matching matrices (after setting the thresholds) to pass the first and second cross-checks. An example of this is to consider values in a PCC matrix which are very close to r_0 – a small adjustment to r_0 can give big differences in the discernment for these values, and extra consideration needs to be made on whether the chosen r_0 is a good choice or not. For the case in figure 15, the choice $r_0 = 95\%$ was chosen partly to compensate for this issue (i.e not being too close the the calculated PCC values).

[13] In volume.

5.1.3 Checking for Validity

A validating factor of the process is found in the similarities between figure 15a and 15b. The PCC matrices are nearly identical, in spite of the matching associated with scan 1 (i.e. the left-most columns and bottom-most rows), which implies that the process can give similar results for discernment made on the basis of data collected from two different sequences. See the supporting figure 23, where the discrepancy is shown explicitly.

5.1.4 Reliability and True Discernment

All results presented in this thesis rely on the discerned volunteers *actually* being separate volunteers, such that the sample group in fact consists different volunteers, and that the testing-, validation- and test scans are separated in a well-controlled manner. As the ground truth was not known for the data used in this thesis, all results presented in this thesis should be further verified with data which is guaranteed to be from different volunteers. The discernment method applied here should also be verified by testing the accuracy of the process on a set of scans for which the true volunteer discernment is already known.

5.2 RF-shimming by MLP Networks

5.2.1 Feasibility of the MLP Networks for Prediction of Full RF-Shims

The shim configurations found by RF-UP-Net and Tailored-Net mimic those calculated by the RF-UPs and the tailored pulses, respectively, as evident from figures 17 and 18. For RF-UP-Net, this is as expected, as the network was trained to remember and interpolate between between the RF-UP configurations shown during training. For Tailored-Net, this is a verification of the assumption that a network can find its own universal pulse setting when forced to do so solely through its (tailored) training data. However, all shimming by either the networks or means of optimization, is outperformed by the weighted CP-mode. The weighted CP-mode will, by definition, construct phase-only shimming configurations which give a constructive interference of the sensitivity maps at the targets' center voxel. Its SAR-efficiency (on average) is higher than for even the tailored configuration, which corresponds to a full shim. Full shims, in contrast to phase-only shims, include modifications to the transmit channels' amplitude to e.g. reduce the global RF-power by repressing channels whose sensitivites have low amplitudes in the desired target location. However, one needs to consider the impact of setting the $V_{\max}$ for a given shim configuration. The full shims required, on average, higher $V_{\max}$ to reach the IEC SAR-limits indicated in the caption of figure 18, compared to the phase-only shim. As a consequence of the relations in eq. (28), their SAR-efficiency will also necessarily be lower compared to the phase-only shim. Another effect from the shim configurations to be considered is interference patterns in the electric fields, not only *constructive*, but also *destructive*. Reducing the relative amplitude of a transmit channel can potentially *increase* the estimated maximum local SAR for a given shimming configuration if e.g. said channel's electric field interferes destructively with the combined electric field of the other transmit channels near a potential point of focal heating. For an insight into the estimated maximum and local SAR-levels for each configuration, as well as the SVS-cube-means, prior to setting $V_{\max}$, see the supporting figure 22 in the appendix.

5.2.2 Comparing Data Requirements and Time-Efficacy

A substantial difference in the different shimming methods applied here lies in their data requirements, which could be used to argue *for* the feasibility of use of a network similar to e.g. Tailored-Net for RF-shimming:

- The tailored pulse requires full B_1^+-data (amplitude and relative phase), and at least $N_C = 8$ individual DREAM-measurements is needed. Gathering this data *in situ* can be very time-consuming, as well as the pulse optimization itself ($\sim$ 15s).
- The weighted CP-mode requires the relative phase-data of the channels' sensitivites, and thus the data gathering is identical to that of the tailored pulse. However, shimming by weighted CP-mode requires no optimization procedure.
- Both RF-UP and Tailored-Net operate similar to UPs, and thus require no volunteer-specific data. That is, any pulse can be predicted within

milliseconds after the center location of the target is given to the network.

- The RF-UP requires no volunteer-specific data, as it is by definition *universal*.

As evident from figure 18, the Tailored-Net outperforms RF-UP-Net. Tailored-Net also yields a similar performance to the tailored shims, in spite of having no data requirements – this is a big benefit of networks, as the user can decide the desired trade-off between the achievable RF-amplitude at a given location and the time required to produce the shimming configuration.

5.2.3 MLP Network Training Efficacy

As for the two MLP networks' training efficacy shown in figure 8a, it is apparent that both networks show signs of underfitting – After ~ 400 epochs, both the training and validation curves flatten out, and no more learning occurs. It is important, however, to keep in mind the goal of the networks – as mentioned in section 3.3, the intention was to train the networks to remember and interpolate between the solutions on which the networks were trained. Increasing the network sizes (and increasing the number of training examples) would more than likely not counteract this tendency to underfit, as it is apparent that the networks successfully learned the intended behavior.

The cause of the difference in the final RMSE of figure reached by the networks, as indicated in figure 8a, lies in their respective training sets. RF-UP-Net always had a 1-to-1 correspondence between its input and output, while Tailored-Net always had at least a 1-to-5 correspondence (see section 3.3). Therefore, after both networks had trained past the $\sim 400^{\text{th}}$ epoch, the RMSE would naturally be larger for Tailored-Net than RF-UP-Net, as there were no unique input-to-output correspondences within Tailored-Net's training set. However, the RMSE calculated during training is just a metric of how well each network manages to predict the PTx-weights constituted by their respective training- and validation sets, and is not a measure of their performance during RF-shimming, as indicated by the results in 17 and 18.

5.2.4 Proposing a universal weighted CP-mode method

The results shown in figure 18 indicate that weighted CP-mode is the most desirable method to use when performing the RF-shimming for the purpose presented here (i.e. shifting the concentration of RF-amplitude around the head). Further work should be done to investigate and expand upon two obvious areas for improvement:

1) Calculating and comparing RF-UPs with the configuration of weighted CP-mode, i.e. calculating universal phase-only RF-shims which yield constructive interference at a desired location, over a set of volunteers. This could be achieved by averaging the phases of all volunteers' B_1^+-maps, and calculating the weighted CP-mode for the given voxel's coordinate in the resulting phase-averaged B_1^+-map.

2) Calculating the weighted CP-mode individually in a similar manner to that discussed in 1), and instead of calculating an average, use the individual

shims to create the training set of a network. This would be identical to the training procedure of *Tailored-Net*, except the training set would here consist of phase-only shims.

5.3 8-$k_{\mathbf{T}}$-Point Whole-Brain FA Homogenization by CNN

5.3.1 Feasibility of the CNN for Weight Prediction

Applying CNN-predicted time-varying weights of an 8-k_{T}-point excitation trajectory for whole-brain FA homogenization may or may not be a feasible approach, as indicated by the results in figure 20 for scans 16 and 17, depending on what criteria is set for the desired maximum SAR-levels. The performance of the CNN for FA-shimming is almost equal to that of the k_{T}-UP, while both methods are outperformed by the tailored pulse, when only taking the resulting CoV from the two different methods into account – however, there is an increase in SAR-efficiency of about 25% in both the estimated maximum and average local SAR in the CNN-predictions and tailored pulses compared to their UP counterpart. If there is a desire to trade FA-homogeneity for SAR-efficiency, the CNN-method could be a feasible approach if there are time-constraints in the scanning procedures, see the next subsection.

5.3.2 Comparing Data Requirements and Time-Efficacy

A very important difference between the three applied methods lies in the amount of volunteer-specific data they respectively require:

- A fully[14] tailored pulse requires full B_1^+-data (amplitude and relative phase) and B_0-data for the volunteer. Gathering this data *in situ* can be very time-consuming, as at least $N_{\mathrm{C}} = 8$ individual DREAM-measurements and a 3DEGRE-measurement are needed. Also, the pulse optimization itself can be quite time-consuming ($\sim$ 30s).
- The CNN requires only the RF-amplitude data (i.e. $|B_1^+|(\mathbf{r}_n)$) from a single DREAM-measurement with the PTx-system in default-drive (all PTx-weight set to unity). The pulse prediction time after the data has been gathered is negligible ($\sim$ 10ms).
- The k_{T}-UP requires no volunteer-specific data, as it is by definition *universal*.

In light of the previous discussion in this section, each method has its own advantage, and the most feasible method is decided by the user-decided trade-off between FA-homogeneity, SAR-levels and time-constraints. Note that all methods assume that B_0-shimming is perform prior to pulse *application* (but after pulse *design*). Therefore, the gathering of an off-resonance map using e.g the 3DEGRE-sequence is inevitable regardless of the choice of pulse design method, unless e.g. a universal B_0-shim configuration is applied.

5.3.3 CNN Training Efficacy and Prediction Performance

The results in figure 21 seems to indicate that the network properly learned the necessary features of the $|B_1^+|$-maps during training to properly predict the pulse settings to perform efficient FA-shimming with the given excitation trajectory. From 21a, the network does not show any sign of underfitting, but a slight

[14] "Fully" is here to indicate that all possible data is included, as in eq. 20. A tailored pulse *could* be tailored assuming e.g. no B_0-inhomogeneities.

overfit on the training set seemed to have occurred from around the 10^{th} epoch. However, a important remark is that the number of examples present in the validation set of was extremely low, and a bigger validation set might have been able capture the statistical characteristics relative to the training set, bringing the two lines closer together. Also note that although the RMSE was lower over the training set than over the validation set after the 10^{th} epoch, the validation RMSE was still *decreasing* at about the same rate as the training RMSE, which indicates efficient learning.

A remark needs to be made on the sparsity of examples available for training, validation and testing of the network. The data from all 17 available scans were used to exhaust the amount of data for setting up and testing the network – the results presented here seem to indicate that only a sparse amount of training examples is required to capture the variability of the desired time-varying weights. The networks should, however, be trained, validated and tested with sets bigger than those presented here to further validate the results – 2 test examples is an insufficient amount of examples to draw any firm conclusions on the results, and increasing the number of training examples past the 13 examples applied here might prove an increase in the prediction performance of the network.

The comparisons shown in figure 21b and 21c seem to indicate that the network has picked up on the most essential traits needed to predict the components of the time-varying weights, compared to their tailored counterparts. The predicted amplitudes and phases manage to trace their tailored counterpart remarkably well, which indicates that the assumption discussed in the previous paragraph regarding the sufficiency of information contained in the $|B_1^+|$-maps might be a reasonable assumption. The bar-overlaps look to be greater for scan 17 than 16, especially for the amplitudes, which could be due to the data associated with scan 17 being more resemblant of the data associated with the scans used for training (scans $1-13$).

Training a network similar to the CNN presented here, but for a $k_{\text{T}} - \textit{point}$ trajectory of fewer than 8-k_{T}-points should be explored – with a sparse amount of training data, there is a drive to minimize the amount of trainable parameters in a network to increase its performance over its test set. Naturally, decreasing the amount of k_{T}-points will decrease the amount of spatial-modulation made to the magnetization, yielding a lower FA homogeneity across the brain. However, a CNN trained to predict the time-varying weights for e.g. a 4-k_{T}-point trajectory could show an increase in prediction performance due to the decrease in number of trainable networks parameters (as the network could potentially become less prone to overfitting).

5.3.4 Proposing Including More Input Data for the CNN

The assumption made when the network was constructed was that using only volunteer-specific RF-amplitude data was sufficient for the network to learn the variability of the output across different volunteers – the $|B_1^+|$-map from PTx default-drive constituting the network's input implicitly contains information about all channel's sensitivity amplitude, as it is simply the amplitude of their unity-weighted superposition. The network should be attempted to be retrained to include more data in the network input, e.g. the phase-data of the B_1^+-map or the off-resonance map, which can be achieved by increasing the number of input channels to the network. However, this comes at the cost of increasing

the network size (i.e. the number of trainable parameters), and will more than likely require even more training data to properly cover the variability of data for the network to be properly able to generalize for use in general volunteer applications. This will also cause the requirement of *gathering* more data – including phase-data would bring the required amount of data gathering to the same level as the tailored pulses and weighted CP-mode for RF-shimming, and including an off-resonance map would require a 3DEGRE-sequence to be performed.

6 Conclusion and Further Work

6.1 Regressional MLP Networks for RF-shimming

The results indicate that predicting universal PTx-weights for RF-amplitude shimming using MLP networks is a feasible approach for time-saving pulse design, although the approach naturally lacks the finesse of its volunteer-tailored counterpart. Both the RF-UP-Net and Tailored-Net successfully performed the shimming for which it was trained (i.e. properly mimicking the RF-UPs and tailored pulses, respectively). The network finding its own compromise for universality was found to be the better option (i.e Tailored-Net) when constructing the training data of a network for RF-amplitude shimming. However, the phase-only shimming found by weighting each channel's phase (i.e. phase-only shimming) by the phase necessary to create constructive phase-interference at the desired shimming location outperformed all other full shims (i.e. amplitude- and phase-shimming) without requiring on-line optimization (although requiring full B_1^+-data). Therefore, a network similiar to Tailored-Net should be further investigated with the same goal in mind, but with its training data consisting of training data constructed from the weighted CP-mode shims (i.e. constructing a network for predicting phase-only shims).

6.2 Regressional CNN network for $k_{\mathbf{T}}$-point FA homogenization

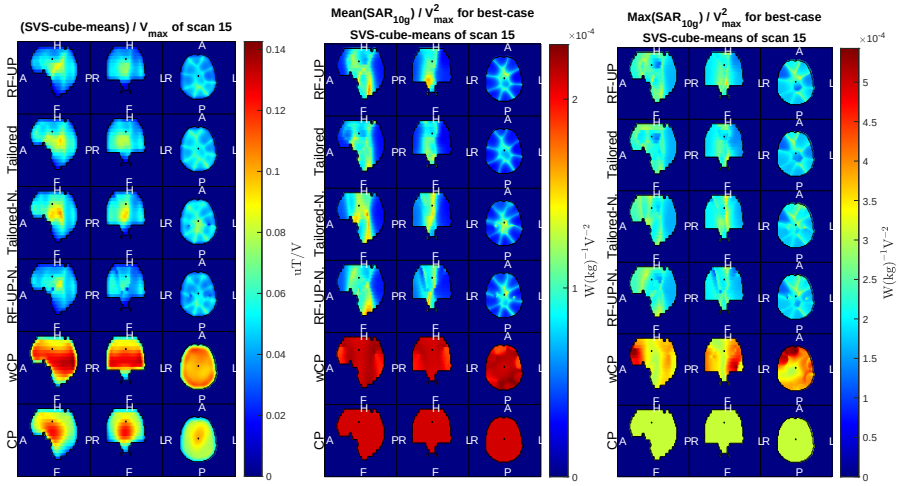
The results from training a CNN for the prediction of time-varying PTx-weights of an 8-k_{T}-point trajectory for whole-brain FA-homogenization are indicatively positive. Using only the RF-amplitude data from PTx default-drive (i.e. CP-mode) as the network input, the resulting pulses share traits from both the kT-UP and tailored pulses – the CNN-predicted pulse settings share approximately equal SAR-levels (maximum and head-average $\mathrm{SAR}_{10\mathrm{g}}$) as its tailored counterpart, but with approximately equal FA-inhomogeneity as the k_{T}-UPs. The CNN-approach presented here should be further investigated to include more MRI data (e.g. relative RF-phase data and off-resonances) in its input to improve its predictions.

6.3 Validity of Results and the Volunteer Discernment Process

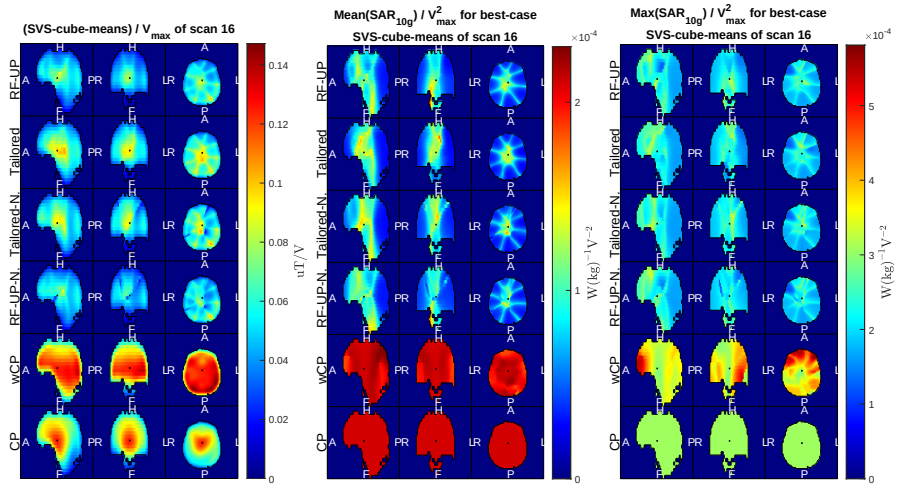
The results outlined above should be further verified for larger test sets, and for volunteer data which is guaranteed to have derived from separate volunteers – the results RF-shimming and whole-brain FA-homogenization were only verified over $N = 3$ and $N = 2$ discerned volunteers, respectively, as discerned by the PCC-method proposed as the secondary objective of this thesis. Although the results from the discernment process indicates a clean volunteer separation (despite some complications due to image artifacts), the discernment process itself should be further validated over a set of volunteer data for which the ground truth of separation is known.

7 Appendix

7.1 Supporting Figures

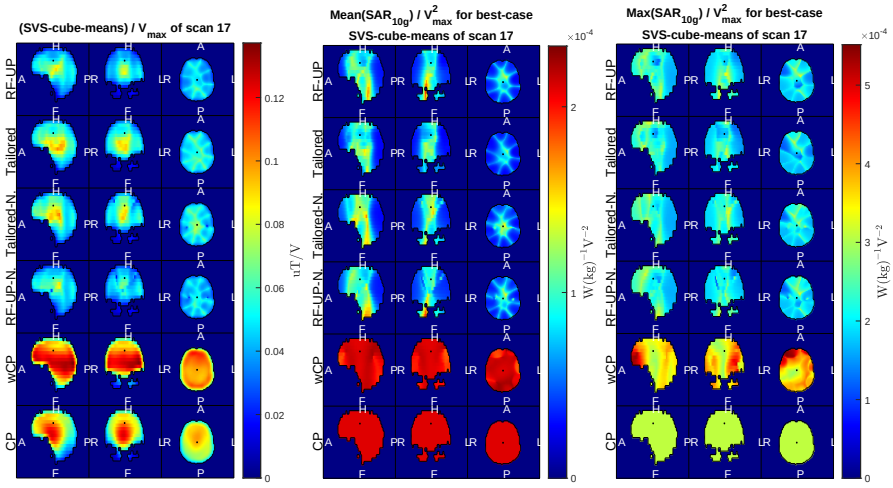


(a)



(b)

Figure 22: *Continued, see the next page for figure details.*



(c)

Figure 22: The distribution of the best-case SVS-cube-means and mean and maximum estimated SAR-levels from the RF-shimming methods applied to the data from scan 15, 16 and 17 in (a), (b) and (c), respectively – the scan numbers are as indicated in figure 15. An important remark is that the maps shown are *not* physical. *Each voxel position's value is derived from a configuration for which the SVS-cube is centered at said voxel, prior to setting V_{max}. That is, each voxel position represents a unique shim configuration targeted at maximizing the RF-amplitude over the SVS-cube centered at that voxel.* The results from RF-UP, tailored, Tailored-Net (Tailored-N.), RF-UP-Net (RF-UP-N.), weighted CP-mode (wCP) and CP-mode (CP) are shown here. The calculated distribution is shown for a slices in the HF-, AP- and RL-planes which intersect at a calculated center-of-mass position. For the SAR-calculations, a 1% RF duty-cycle was assumed (changing the RF duty-cycle does not change the relative scaling of the achievable SVS-cube-means).

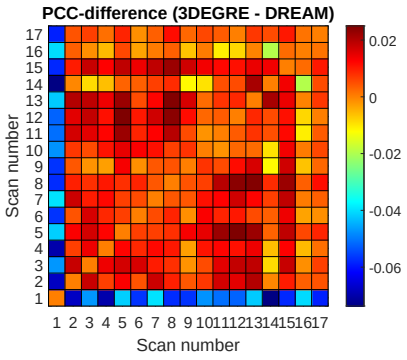


Figure 23: The difference in the calculated PCCs from the 3DEGRE- and DREAM-data. This map corresponds to element-wise subtraction of the values in figure 15b from those in figure 15a.

7.2 Derivations

7.2.1 Details of the Small-Tip-Angle (STA) Approximation

In this section, we will show the details omitted in the main text of section 2.1.2 when solving the decoupled Bloch equations.

The differential equation for $M_{x'y'}$ reads, after inserting (5) into (4) and calculating out its matrix multiplication,

$$\frac{\mathrm{d}}{\mathrm{d}t} M_{x'y'}(\mathbf{r},t) = -i\gamma \mathbf{G}\cdot\mathbf{r} M_{x'y'}(\mathbf{r},t) + i\gamma M_0 B_1^+(\mathbf{r},t) \tag{29}$$

.

Multiplying by the integrating factor $\exp\left(\int_0^t i\gamma \mathbf{G}(\tau)\cdot\mathbf{r}\,\mathrm{d}\tau\right)$ and applying the initial condition of zero transverse magnetization yields, after integration over the RF duration $t\in[0,T]$,

$$\begin{aligned}
&M_{x'y'}(\mathbf{r},T)\exp\left(\int_0^T i\gamma\mathbf{G}(\tau)\cdot\mathbf{r}\,\mathrm{d}\tau\right)\\
&= i\gamma M_0\int_0^T B_1^+(\mathbf{r},t)\exp\left(\int_0^t i\gamma\mathbf{G}(\tau)\cdot\mathbf{r}\,\mathrm{d}\tau\right)\mathrm{d}t\\
\Longleftrightarrow&\\
&M_{x'y'}(\mathbf{r},t)\\
&= i\gamma M_0\int_0^T B_1^+(\mathbf{r},t)\exp\left(\int_0^t i\gamma\mathbf{G}(\tau)\cdot\mathbf{r}\,\mathrm{d}\tau - \int_0^T i\gamma\mathbf{G}(\tau)\cdot\mathbf{r}\,\mathrm{d}\tau\right)\mathrm{d}t\\
&= i\gamma M_0\int_0^T B_1^+(\mathbf{r},t)\exp\left(i\mathbf{r}\cdot\left[-\gamma\int_t^T\mathbf{G}(\tau)\mathrm{d}\tau\right]\right)\mathrm{d}t\\
&= i\gamma M_0\int_0^T B_1^+(\mathbf{r},t)e^{i\mathbf{r}\cdot\mathbf{k}(t)}\mathrm{d}t,\ \text{where } \mathbf{k}(t)\equiv-\gamma\int_t^T\mathbf{G}(\tau)\mathrm{d}\tau.
\end{aligned}$$

7.2.2 Details of the Spatial Domain Pulse Design of Spokes Pulses

We refer to the nomenclature introduced in 2.2.3, and apply the spatial and timely discretization given there. Furthermore, the remaining time $(t-T_p)$ in the inhomogeneity contribution $\Delta B_0(\mathbf{r})$ of the integrand in (16) can be written for an arbitrary discretized time step $m\in[1,\ldots,N_t]$ into a sub-pulse at a given k_T-point $k\in[1,2,\ldots,N_{k_\mathrm{T}}]$ as (see figure 24)

$$(t-T_p)\rightarrow t'_k+(N_t-m)\Delta t. \tag{30}$$

The k-space trajectory $\mathbf{k}(t)$ is constant for a given k_T-point, and can be evaluated at t'_k for the entire duration of the k^{th} k_T-point . This approximation, along with eq. (30), yields for eq. (16) its discretiztion

$$
\begin{aligned}
&M_{x'y'}(\mathbf{r}_n, T_p) \\
&\approx i\gamma M_0 \Delta t \sum_{j=1}^{N_\mathrm{C}} S_j(\mathbf{r}_n) \sum_{k=1}^{N_{k_\mathrm{T}}} w_{jk} \sum_{m=1}^{N_t} \rho_m e^{i\mathbf{r}_n \cdot \mathbf{k}(t_m) + i\gamma \Delta B_0(\mathbf{r})(t'_k + (N_t - m)\Delta t} \\
&\approx \sum_{j=1}^{N_\mathrm{C}} \sum_{k=1}^{N_{k_\mathrm{T}}} w_{jk} \left[i\gamma M_0 \Delta t S_j(\mathbf{r}_n) e^{i\mathbf{r}_n \cdot \tilde{\mathbf{k}}(t'_k)} \sum_{m=1}^{N_t} \rho_m e^{i\gamma \Delta B_0(\mathbf{r})(t'_k + (N_t - m)\Delta t} \right] \\
&= \sum_{j=1}^{N_\mathrm{C}} \sum_{k=1}^{N_{k_\mathrm{T}}} w_{jk} a_{knj}.
\end{aligned}
$$

Defining the system matrix A and vectors $\mathbf{p}, \mathbf{m}$ as in section 2.2.3, the above equation is equivalent to the approximation of the transverse magnetization as given by eq. (19).

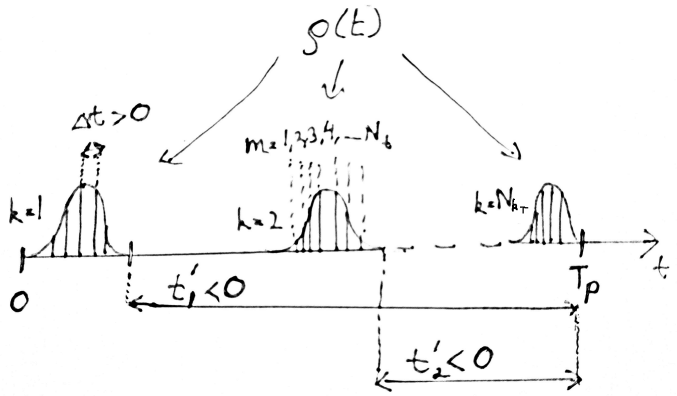


Figure 24: An illustration used to visualize the timely discretization of $(t - T_p)$ in eq. (16).

7.2.3 Derivation of the Backpropagation Equations in MLP Nets

We refer to the nomenclature presented in section 2.3. We compute each element through the chain rule:

$$
\begin{aligned}
\delta_{j,n}^{(L)} &= \frac{\partial C_n}{\partial a_j^{(L)}} && = \frac{y_j - a_j^{(L)}}{N_\mathrm{out}}, \\
\delta_{k,n}^{(l-1)} &= \sum_{j',j''} \frac{\partial C_n}{\partial a_{j'}^{(l)}} \frac{\partial a_{j'}^{(l)}}{\partial z_{j''}^{(l)}} \frac{\partial z_{j''}^{(l)}}{\partial a_k^{(l-1)}} && = \sum_{j'} \delta_{j',n}^{(l)} \sigma' \left(z_{j'}^{(l)} \right) w_{j'k}^{(l)}, \\
\frac{\partial C_n}{\partial w_{jk}^{(l)}} &= \sum_{j',j''} \frac{\partial C_n}{\partial a_{j'}^{(l)}} \frac{\partial a_{j'}^{(l)}}{\partial z_{j''}^{(l)}} \frac{\partial z_{j''}^{(l)}}{\partial w_{jk}^{(l)}} && = \delta_{j,n}^{(l)} \sigma' \left(z_{j'}^{(l)} \right) a_k^{(l-1)}, \\
\frac{\partial C_n}{\partial b_j^{(l)}} &= \sum_{j',j''} \frac{\partial C_n}{\partial a_{j'}^{(l)}} \frac{\partial a_{j'}^{(l)}}{\partial z_{j''}^{(l)}} \frac{\partial z_{j''}^{(l)}}{\partial b_j^{(l)}} && = \delta_{j,n}^{(l)} \sigma' \left(z_{j'}^{(l)} \right).
\end{aligned}
$$

7.2.4 Derivation of the Intensity Equations for B_1^+-mapping

We refer to figure 2. We will track the longitudinal magnetizations M_{FID} and M_{STE} which will eventually be the source of the signal of the FID and STE, respectively. For M_{FID}, the prepared magnetization is comprised of the spins which are not flipped into the transverse plane by either α-pulse, i.e. the twice repeated projection of the tipped magnetization onto the $\hat{\mathbf{z}}$-direction,

$$M_{\mathrm{FID}} = \cos(\alpha)[\cos(\alpha)M_0] = \cos^2(\alpha)M_0.$$

For M_{STE}, we flip the transverse magnetization, i.e. the projection of the tipped magnetization onto the $x'y'$-plane, created by the first α-pulse back into the longitudinal direction after a time T_s by a second α-pulse. During the duration T_s, we assume we have completely de-phased all transverse magnetization created by the first pulse with gradient G_m. Application of the second α-pulse returns, on average, half of the transverse magnetization back along the longitudinal direction, i.e.

$$M_{\mathrm{STE}} = \sin(\alpha)\left[\frac{1}{2}\sin(\alpha)M_0\right] = \frac{1}{2}\sin^2(\alpha)M_0$$

(a superb illustration of the evolution of the STEAM prepared magnetization is shown in [45, Fig. 2]). Both the signal of the FID and STE are then the transversal component of the M_{FID} and M_{STE} magnetizations, respectively, tipped by the β-pulse, which gives the intensities as provided in the main text.

7.2.5 Details of the Sensitivity Encoding Calculations

Denote the entry of E at (m, j) as

$$E_{m,j} \equiv \exp\left(\frac{2\pi i(m-1)j}{M}\right).$$

The entry at (j, j') of $E^H E$ can then be written as the dot product (by writing out the matrix multiplication explicitly)

$$\sum_{l=1}^{M} E_{l,j}^* E_{l,j'} = \sum_{l=1}^{M} \exp\left(\frac{2\pi i(l-1)(j'-j)}{M}\right) = \frac{1-\exp\left(2\pi i(j'-j)\right)}{1-\exp\left(\frac{2\pi i(j'-j)}{M}\right)}$$

In the last step we recognize the sum as a geometric series if $j \neq j'$, and note that it is zero. If $j = j'$, then the sum is simply equal to M.

7.3 Deep Learning in Convolutional Neural Networks

We here refer to the nomenclature introduced in section 2.3.9.

7.3.1 The Forward Pass in Convolutional Layer

For the convolutional layer l, we denote each activation (i.e. output feature map pixel value) as

$$a^{(l)}_{jx'y'}, \text{ at pixel } (x', y') \in [0, \dots, H_2 - 1] \times [0, \dots, W_2 - 1], \\ \text{for output channel } j \in [0, \dots, C_2].$$

For the preceding layer $l-1$, each activation is denoted as

$$a^{(l-1)}_{jxy}, \text{ at pixel } (x, y) \in [0, \dots, H_1 - 1] \times [0, \dots, W_1 - 1], \\ \text{for output channel } k \in [0, \dots, C_1].$$

The instances where the pixel indices are omitted we are referring to a specific map, i.e. $\mathbf{a}^{(l)}_j$ or $\mathbf{a}^{(l-1)}_k$. The kernel used in the convolution of map k contributing to $\mathbf{a}^{(l)}_j$ is denoted $\boldsymbol{w}^{(l)}_{jk}$, which is further indexed pixel-wise as $w^{(l)}_{jkab}$, where $(a, b) \in [0, \dots, k_1 - 1] \times [0, \dots, k_2 - 1]$. With this notation, we can now define the output maps $\mathbf{a}^{(l)}_j$ in terms of the input maps $\mathbf{a}^{(l-1)}_k$:

$$\begin{aligned} a^{(l)}_{jxy} &\equiv \sigma\left(z^{(l)}_{jxy}\right) \\ &\equiv \sigma\left(b^{(l)}_{j'} + \left[\text{rot180}^\circ\left(\boldsymbol{w}^{(l)}_{jk}\right) * \mathbf{a}^{(l-1)}_k\right]_{xy}\right) \\ &\equiv \sigma\left(b^{(l)}_{j'} + \sum_{k'=0}^{c-1}\sum_{a'=0}^{k_1-1}\sum_{b'=0}^{k_2-1} w^{(l)}_{j'k'a'b'} a^{(l-1)}_{k',s_1x+a'-p_1,s_2y+b'-p_2}\right). \end{aligned} \tag{31}$$

Here, $\sigma(\cdot)$ is the activation function of layer l (applied to each element of its argument separately), $\mathbf{z}^{(l)}_j$ is the convolved input of channel j (with same dimensions as $\mathbf{a}^{(l)}_j$), and $b^{(l)}_j$ is the bias of layer l, shared between all input channels for a given output channel j. The function $\text{rot180}^\circ(\cdot)$ takes a 2-D tensor a flips it horizontally and vertically. Furthermore, the $*$-operation denotes convolution, parameterized by the padding p_1, p_2 and stride s_1, s_2 in the x, y-directions, respectively. We only allow indices which are not out of bounds in (31), which may be adjusted by padding. For an arbitrary choice of padding and stride, the resulting output map is of size

$$\left(\left\lfloor \frac{H_1 - k_1 + 2p_1}{s_1} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{W_1 - k_2 + 2p_2}{s_2} \right\rfloor + 1\right).$$

To get nice-looking formulas, we're only going to do *valid* convolution (i.e. no padding, $p_1, p_2 = 0$) and use unity strides ($s_1, s_2 = 1$). The matrix results which will be presented here can be extended to general padding and strides by stretching of the kernels included in the computations[46].

7.3.2 Backpropagation in Convolutional Layers

In order to perform backpropagation, we need to compute gradients

$$\frac{\partial C}{\partial w_{jkab}^{(l)}}, \frac{\partial C}{\partial b_j^{(l)}},$$

to update layer l, and gradient

$$\frac{\partial C}{\partial a_{kxy}^{(l-1)}}$$

to pass down to layer $l-1$, *all in terms of variables we can compute explicitly.* Starting with the chain rule for $\frac{\partial C}{\partial w_{jkab}^{(l)}}$, we get

$$\frac{\partial C}{\partial w_{jkab}^{(l)}} = \sum_{j',x',y'} \frac{\partial C}{\partial a_{j'x'y'}^{(l)}} \frac{\partial a_{j'x'y'}^{(l)}}{\partial z_{j'x'y'}^{(l)}} \frac{\partial z_{j'x'y'}^{(l)}}{\partial w_{jkab}^{(l)}}. \tag{32}$$

Here, j' runs over all output maps, $j' = 0, \ldots, C_2 - 1$, and x', y' runs over all pixels (neurons) in a given output map (same dimensions for all maps in same layer). To compute (32) we note that:

1) The first factor on the RHS of (32) is given (assumed this was passed down when we computed backwards function in previous layer).

2) The second factor on the RHS of (32) is just $\sigma'(z_{j'x'y'}^{(l)})$.

3) The third factor filters out all terms except those which $k' = k, a' = a, b' = b$. That is,

$$\frac{\partial z_{j'x'y'}^{(l)}}{\partial w_{jkab}^{(l)}} = \begin{cases} a_{k,x'+a,y'+b}^{(l-1)} & \text{if } j' = j, \\ 0 & \text{otherwise.} \end{cases}$$

Note that we also filter on $j' = j$, so only the j-terms survive and the j'-summation in (32) vanishes.

Now, plugging in what we found above into (32), gives

$$\frac{\partial C}{\partial w_{jkab}^{(l)}} = \sum_{x',y'} \left[\frac{\partial C}{\partial a_{jx'y'}^{(l)}} \sigma'(z_{jx'y'}^{(l)}) \right] a_{k,x'+a,y'+b}^{(l-1)}. \tag{33}$$

This looks very much like convolution – in fact, this is just a valid, unity-strided convolution with a rotated kernel (the bracketed factor). That is, we can write the gradient of our weights for the kernel from input map k to output map j as a 2-D matrix, whose entries (a, b) indicate the weight gradient for kernel-weight (a, b):

$$\frac{\partial C}{\partial w_{jk}^{(l)}} = \text{rot}180^\circ \left(\frac{\partial C}{\partial \mathbf{a}_j^{(l)}} \odot \sigma'_{\text{vec}}(z_j^{(l)}) \right) * \mathbf{a}_k^{(l-1)}.$$

The latter factor of the convolution in the equation is the element-wise product of the two matrices with entries (x', y') as indicated by the two products, respectively, in the bracket of equation (33). *Point is, we can now update the weights*

in this layer. We can do the same for the bias using the same angle of attack as before. For each output bias $b_j^{(l)}$, we compute

$$\frac{\partial C}{\partial b_j^{(l)}} = \sum_{j',x',y'} \frac{\partial C}{\partial a_{j'x'y'}^{(l)}} \frac{\partial a_{j'x'y'}^{(l)}}{\partial z_{j'x'y'}^{(l)}} \frac{\partial z_{j'x'y'}^{(l)}}{\partial b_j^{(l)}}.$$

We note from equation (31) that we filter on $j' = j$, and that the derivative is just 1, i.e.

$$\frac{\partial z_{j'x'y'}^{(l)}}{\partial b_j^{(l)}} = \frac{\partial (b_j^{(l)} + \text{convolution stuff})}{\partial b_j^{(l)}} = 1.$$

Using the same notation as before, we quickly get the resulting bias gradient

$$\begin{aligned} \frac{\partial C}{\partial b_j^{(l)}} &= \sum_{x',y'} \frac{\partial C}{\partial a_{j'x'y'}^{(l)}} \frac{\partial a_{jx'y'}^{(l)}}{\partial z_j^{(l)} x'y'} \\ &\equiv \text{sum} \left(\frac{\partial C}{\partial \mathbf{a}_j^{(l)}} \odot \sigma'_{\text{vec}}(z_j^{(l)}) \right), \end{aligned}$$

where $\text{sum}(\cdot)$ indicates the sum over all columns and rows. Note how the RHS of this equation is independent of k. This makes sense if one convinces oneself that the bias is just a constant term added to the convoluted input, and its contribution is the same regardless of the input signal. Note that the bias gradient is still implicitly dependent on the input signals due to their contribution to $z_j^{(l)}$ and therefore $a_j^{(l)}$.

We also need a gradient to pass down to the next layer in the next backward pass, i.e. how the cost function changes w.r.t. the activations in the previous layer. We start with the chain rule, but this time, we need to express our weighted input $z_j'^{(l)}$, in terms of the activations $a_k^{(l-1)}$. Here's the key argument: *the activation $a_{j'x'y'}^{(l)}$ is only dependent on $a_{kxy}^{(l-1)}$ iff it was included in the weighted sum $z_{j'x'y'}^{(l)}$.* More specifically, $a_{j'x'y'}^{(l)}$ is dependent on $a_{kxy}^{(l-1)}$ iff[15] $(x', y') = (x - a, y - b)$ for at least one $(a, b) \in [0, \ldots, k_1 - 1] \times [0, \ldots, k_2 - 1]$. For the chain rule, these are the only terms which are not guaranteed to zero out, so we must include them:

$$\frac{\partial C}{\partial a_{kxy}^{(l-1)}} = \sum_{j'} \sum_{a=0}^{k_1-1} \sum_{a=0}^{k_2-1} \frac{\partial C}{\partial a_{j',x-a,y-b}^{(l)}} \frac{\partial a_{j',x-a,y-b}^{(l)}}{\partial z_{j',x-a,y-b}^{(l)}} \frac{\partial z_{j',x-a,y-b}^{(l)}}{\partial a_{kxy}^{(l-1)}}. \tag{34}$$

We here assume any term in the sum above which have out-of-bounds indices are set to zero to keep the equation well-defined for all x, y. Now, by inspection of equation (31), we see that only one term survives for the last factor in (31), filtering $a = a', b = b'$. This yields

[15] If this doesn't seem apparent, draw the convolution process for an arbitrarily sized input map and kernel, and convince yourself that pixel (x', y') in the input map contributes only to the pixels in the output map which are covered by the overlap of the kernel placed with its bottom-right pixel on output pixel (x', y').

$$\frac{\partial C}{\partial a_{kxy}^{(l-1)}} = \sum_{j'} \sum_{a=0}^{k_1-1} \sum_{a=0}^{k_2-1} \frac{\partial C}{\partial a_{j',x-a,y-b}^{(l)}} \frac{\partial a_{j',x-a,y-b}^{(l)}}{\partial z_{j',x-a,y-b}^{(l)}} w_{j'kab}^{(l)}.$$

Note that the two latter sums yield a convolution of the product of the two first factors with the last[16]. More precisely,

$$\frac{\partial C}{\partial a_{kxy}^{(l-1)}} = \sum_{j'} \left(\frac{\partial C}{\partial \mathbf{a}_{j'}^{(l)}} \odot \sigma'_{\text{vec}}(z_{j'}^{(l)}) \right) * \boldsymbol{w}_{j'k}^{(l)}. \tag{35}$$

7.3.3 Max Pooling Layers

A max-pooling layer is similar to the convolutional layer, except it has no optimizeable parameters, and its kernel only passes forward the maximum value for a given input-with-kernel "overlap", where the input channel size matches the output channel size (each output channel is the max-sampled version of its input channel). Thus, the activation $a_{jx'y'}^{(l)}$ in (31) is here replaced by

$$a_{jxy}^{(l)} \equiv \max_{\substack{0 \le a \le k_1 - 1 \\ 0 \le b \le k_2 - 1}} a_{j,s_1 x + a - p_1, s_2 y + b - p_2}^{(l-1)}. \tag{36}$$

As for backpropagation through a max pooling layer, the layer has no parameters to update, as its only purpose is to map the max value from an input map with respect to a maxing filter/kernel to an output feature map. Thus, we only need to calculate how we propagate the error backwards to the next layer down the line:

$$\frac{\partial C}{\partial a_{kxy}^{(l-1)}} = \sum_{x',y'} \frac{\partial C}{\partial a_{kx'y'}^{(l)}} \frac{\partial a_{kx'y'}^{(l)}}{\partial a_{kxy}^{(l-1)}}. \tag{37}$$

Note that x, y run over the *input map*, x', y' run over the *output map*, and the maps are generally of different sizes. However, each output map takes only one input map, and they can therefore be indexed equally in the first index (as with k show in (37)).

The last factor in (37) can be written

$$\frac{\partial a_{kx'y'}^{(l)}}{\partial a_{kxy}^{(l-1)}} = \begin{cases} 1 & \text{if } a_{kx'y'}^{(l)} \equiv a_{kxy}^{(l-1)}, \\ 0 & \text{otherwise,} \end{cases}$$

that is, *if* $a_{kxy}^{(l-1)}$ was passed as max to pixel (x', y') in feature map j' during the forward pass, *then* its activation is $a_{kxy}^{(l-1)}$. As a result, only instances where $a_{kxy}^{(l-1)}$ was passed contributes to the sum in (37). Naïvely, this can be

[16] By the commutation property of convolution in (31).

tracked programmatically during the forward pass and be later retrieved for backpropagation in a 5-D binary tensor, $M^{(l)}$, whose entries are hot (i.e. 1) if $a^{(l)}_{kx'y'} \equiv a^{(l-1)}_{kxy}$, and cold (i.e. 0) otherwise:

$$M^{(l)}_{k,(x,y),(x',y')} = \begin{cases} 1 & \text{if } a^{(l)}_{kx'y'} \equiv a^{(l-1)}_{kxy}, \\ 0 & \text{otherwise.} \end{cases}$$

Regardless of implementation, this must be **tracked**, and not only compared by value, as several input values can be equal by comparison to the activation. Summarized, we get an equation we can compute:

$$\frac{\partial C}{\partial a^{(l-1)}_{kxy}} = \sum_{x',y'} \frac{\partial C}{\partial a^{(l)}_{kx'y'}} M^{(l)}_{k,(x,y),(x',y')}. \tag{38}$$

7.3.4 Average Pooling Layers

An average pooling layer is identical to that a convolutional layer, except its kernels' entries are all fixed and equal to the inverse of the kernel volume, i.e.

$$w^{(l)}_{jkab} \equiv \frac{1}{k_1 k_2} \; \forall \, j, k, a, b. \tag{39}$$

Thus, its forward and backward pass are described by eq. (31) and (35), respectively, with the weights as in eq. (39).

7.4 Image Matching

Let X, Y be two 3-D images of equal sizes containing $i = 1, \ldots, N_{\text{vox}}$ real and positive intensities x_i, y_i, respectively. Let $\bar{x}, \bar{y}$ be the average intensity of each respective image. Then a pixel-by-pixel matching metric of X and Y is the Pearson Correlation Coefficient (PCC) $0 \leq r \leq 1$, where

$$r \equiv \frac{\sum_i (x_i - \bar{x})(y_i - \bar{x})}{\sqrt{(\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{x})^2}}$$

7.5 Q-matrices for SAR-calculations

The Q-matrices for the Nova head coil were calculated using data from numerical simulations which was distributed by Nova. The technical details of the numerical simulations is proprietary information reserved by Nova. However, the simulations were conducted using

- the human model Hugo[47] as the subject, representing a 38–year-old male, who is 187cm tall with an approximate weight of 114kg[48].
- Remcom[49] simulation software.
- the finite-difference time-domain method (FDTD)[50].

The SAR averaging volume was for $m = 10$g of tissue, i.e. the calculated local SAR was $\text{SAR}_{10\text{g}}$, with the averaging technique applied as in [29].

References

[1] M. E. Ladd, P. Bachert, M. Meyerspeer, E. Moser, A. M. Nagel, D. G. Norris, S. Schmitter, O. Speck, S. Straub, and M. Zaiss. Pros and cons of ultra-high-field mri/mrs for human application. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 109:1–50, 2018.

[2] Q. X. Yang, J. Wang, X. Zhang, C. M. Collins, M. B. Smith, H. Liu, X.-H. Zhu, J. T. Vaughan, K. Ugurbil, W. Chen, and et al. Analysis of wave behavior in lossy dielectric samples at high field. *Magnetic Resonance in Medicine*, 47(5):982–989, 2002.

[3] C. M. Collins, W. Liu, W. Schreiber, Q. X. Yang, and M. B. Smith. Central brightening due to constructive interference with, without, and despite dielectric resonance. *Journal of Magnetic Resonance Imaging*, 21(2):192–196, 2005.

[4] C. B. Sørensen. Simulated parallel transmission b1+-shimming at 7 tesla by deep neural networks. December 2019. [Internal publication, please contact the author at `chrisbso@stud.ntnu.no` for access].

[5] M. V. Vaidya, C. M. Collins, D. K. Sodickson, R. Brown, G. C. Wiggins, and R. Lattanzi. Dependence of b1- and b1+ field patterns of surface coils on the electrical properties of the sample and the mr operating frequency. *Concepts in Magnetic Resonance Part B: Magnetic Resonance Engineering*, 46(1):25–40, 2016.

[6] D. I. Hoult. The principle of reciprocity in signal strength calculations: A mathematical guide. *Concepts in Magnetic Resonance*, 12(4):173–187, 2000.

[7] J. Pauly, D. Nishimura, and A. Macovski. A k-space analysis of small-tip-angle excitation. *Journal of Magnetic Resonance (1969)*, 81(1):43–56, 1989.

[8] R. Schneider. *Selective excitation MR imaging with parallel transmission (pTx)*. PhD thesis, 2015.

[9] R. W. Brown, Y.-C. N. Cheng, E. M. Haacke, M. R. Thompson, and R. Venkatesan. *Magnetic Resonance Imaging: Physical Principles and Sequence Design (2nd Edt.)*. John Wiley & Sons, Inc., 2014.

[10] F. Padormo, A. Beqiri, J. V. Hajnal, and S. J. Malik. Parallel transmission for ultrahigh-field imaging. *NMR in Biomedicine*, 29(9):1145–1161, 2015.

[11] M. A. Bernstein, K. F. King, and X. J. Zhou. *Handbook of MRI pulse sequences*. Elsevier, Acad. Press, 2005.

[12] R. Cusack and N. Papadakis. New robust 3-d phase unwrapping algorithms: Application to magnetic field mapping and undistorting echoplanar images. *NeuroImage*, 16(3):754–764, 2002.

[13] J. Gallier. Notes on spherical harmonics and linear representations of lie groups, Nov 2013. `https://www.cis.upenn.edu/~cis610/sharmonics.pdf`. Retrieved 28/03/2020.

[14] P. Hudson. Pushing the boundaries in gradient and shim design for mri. *Electronic Thesis and Dissertation Repository*, page 99, 2011.

[15] V. Aboites. Legendre polynomials: a simple methodology. *Journal of Physics: Conference Series*, 1221, 2019.

[16] M. Arioli and S. Gratton. Linear regression models, least-squares problems, normal equations, and stopping criteria for the conjugate gradient method. *Computer Physics Communications*, 183(11):2322–2336, 2012.

[17] D. Brenner, D. H. Y. Tse, E. Pracht, T. Feiweier, R. Stirnberg, and T. Stoecker. 3dream – a three-dimensional variant of the dream sequence. 05 2014.

[18] K. Nehrke and P. Börnert. Dream - a novel approach for robust, ultrafast, multislice b1 mapping. *Magnetic Resonance in Medicine*, 68(5):1517–1526, 2012.

[19] A. Beqiri. *Parallel Transmission MRI for Optimised Cardiac Imaging and Improved Safety*. PhD thesis, 08 2015.

[20] P. Ehses, D. Brenner, R. Stirnberg, E. D. Pracht, and T. Stöcker. Whole-brain b 1 -mapping using three-dimensional dream. *Magnetic Resonance in Medicine*, 2019.

[21] K. Nehrke, M. J. Versluis, A. Webb, and P. Börnert. Volumetric b1 mapping of the brain at 7t using dream. *Magnetic Resonance in Medicine*, 71(1):246–256, 2013.

[22] M. A. Cloos, N. Boulant, M. Luong, G. Ferrand, E. Giacomini, D. L. Bihan, and A. Amadon. kt-points: Short three-dimensional tailored rf pulses for flip-angle homogenization over an extended volume. *Magnetic Resonance in Medicine*, 67(1):72–80, 2011.

[23] D. H. Y. Tse, C. J. Wiggins, D. Ivanov, D. Brenner, J. Hoffmann, C. Mirkes, G. Shajan, K. Scheffler, K. Uludağ, B. A. Poser, and et al. Volumetric imaging with homogenised excitation and static field at 9.4 t. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 29(3):333–345, 2016.

[24] K. Setsompop, L. L. Wald, V. Alagappan, B. Gagoski, F. Hebrank, U. Fontius, F. Schmitt, and E. Adalsteinsson. Parallel rf transmission with eight channels at 3 tesla. *Magnetic Resonance in Medicine*, 56(5):1163–1171, 2006.

[25] K. Setsompop, L. Wald, V. Alagappan, B. Gagoski, and E. Adalsteinsson. Magnitude least squares optimization for parallel radio frequency excitation design demonstrated at 7 tesla with eight channels. *Magnetic Resonance in Medicine*, 59(4):908–915, 2008.

[26] D. H. Y. Tse, C. J. Wiggins, and B. A. Poser. High-resolution gradient-recalled echo imaging at 9.4t using 16-channel parallel transmit simultaneous multislice spokes excitations with slice-by-slice flip angle homogenization. *Magnetic Resonance in Medicine*, 78(3):1050–1058, 2016.

[27] V. Gras, A. Vignaud, A. Amadon, D. L. Bihan, and N. Boulant. Universal pulses: A new concept for calibration-free parallel transmission. *Magnetic Resonance in Medicine*, 77(2):635–643, 2016.

[28] M. D. Greef, O. Ipek, A. J. E. Raaijmakers, J. Crezee, and C. A. T. V. D. Berg. Specific absorption rate intersubject variability in 7t parallel transmit mri of the head. *Magnetic Resonance in Medicine*, 69(5):1476–1485, Mar 2012.

[29] I. Graesslin, H. Homann, S. Biederer, P. Börnert, K. Nehrke, P. Vernickel, G. Mens, P. Harvey, and U. Katscher. A specific absorption rate prediction concept for parallel transmission mr. *Magnetic Resonance in Medicine*, 68(5):1664–1674, Sep 2012.

[30] G. Eichfelder and M. Gebhardt. Local specific absorption rate control for parallel transmission by virtual observation points. *Magnetic Resonance in Medicine*, 66(5):1468–1476, 2011.

[31] D. H. Y. Tse, M. S. Poole, A. W. Magill, J. Felder, D. Brenner, and N. J. Shah. Encoding methods for b1 mapping in parallel transmit systems at ultra high field. *Journal of Magnetic Resonance*, 245:125–132, 2014.

[32] Multirate signal processing. *Multirate and Wavelet Signal Processing Wavelet Analysis and Its Applications*, page 1–28, 1998.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[34] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks, 2011. `https://www.utc.fr/~bordesan/dokuwiki/_media/en/glorot10nipsworkshop.pdf`. Retrieved 25/03/2020.

[35] M. Nielsen. Neural networks and deep learning, Des 2019. `http://neuralnetworksanddeeplearning.com/index.html`. Retrieved 26/03/2020.

[36] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[37] S. Varma and S. Das. *Introduction to Deep Learning*. 04 2018. `https://srdas.github.io/DLBook/`. Retrieved 28/03/2020.

[38] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[39] J. D. Ianni, Z. Cao, and W. A. Grissom. Machine learning rf shimming: Prediction by iteratively projected ridge regression. *Magnetic Resonance in Medicine*, 80(5):1871–1881, 2018.

[40] M. S. Vinding, B. Skyum, R. Sangill, and T. E. Lund. Ultrafast (milliseconds), multidimensional rf pulse design with deep learning. *Magnetic Resonance in Medicine*, 82(2):586–599, 2019.

[41] interpn: Interpolation for 1-d, 2-d, 3-d, and n-d gridded data in ndgrid format. https://se.mathworks.com/help/matlab/ref/interpn.html.

[42] Spm - statistical parametric mapping. https://www.fil.ion.ucl.ac.uk/spm/software/spm12.

[43] Mutual information as an image matching metric, 2016. https://matthew-brett.github.io/teaching/mutual_information.html. Retrieved 06/04/2020.

[44] D. Grainger. Safety guidelines for magnetic resonance imaging equipment in clinical use, Nov 2014. http://www.ismrm.org/smrt/files/con2033065.pdf. Retrieved 02/06/2020.

[45] D. Burstein. Stimulated echoes: Description, applications, practical hints. *Concepts in Magnetic Resonance*, 8(4):269–278, 1996.

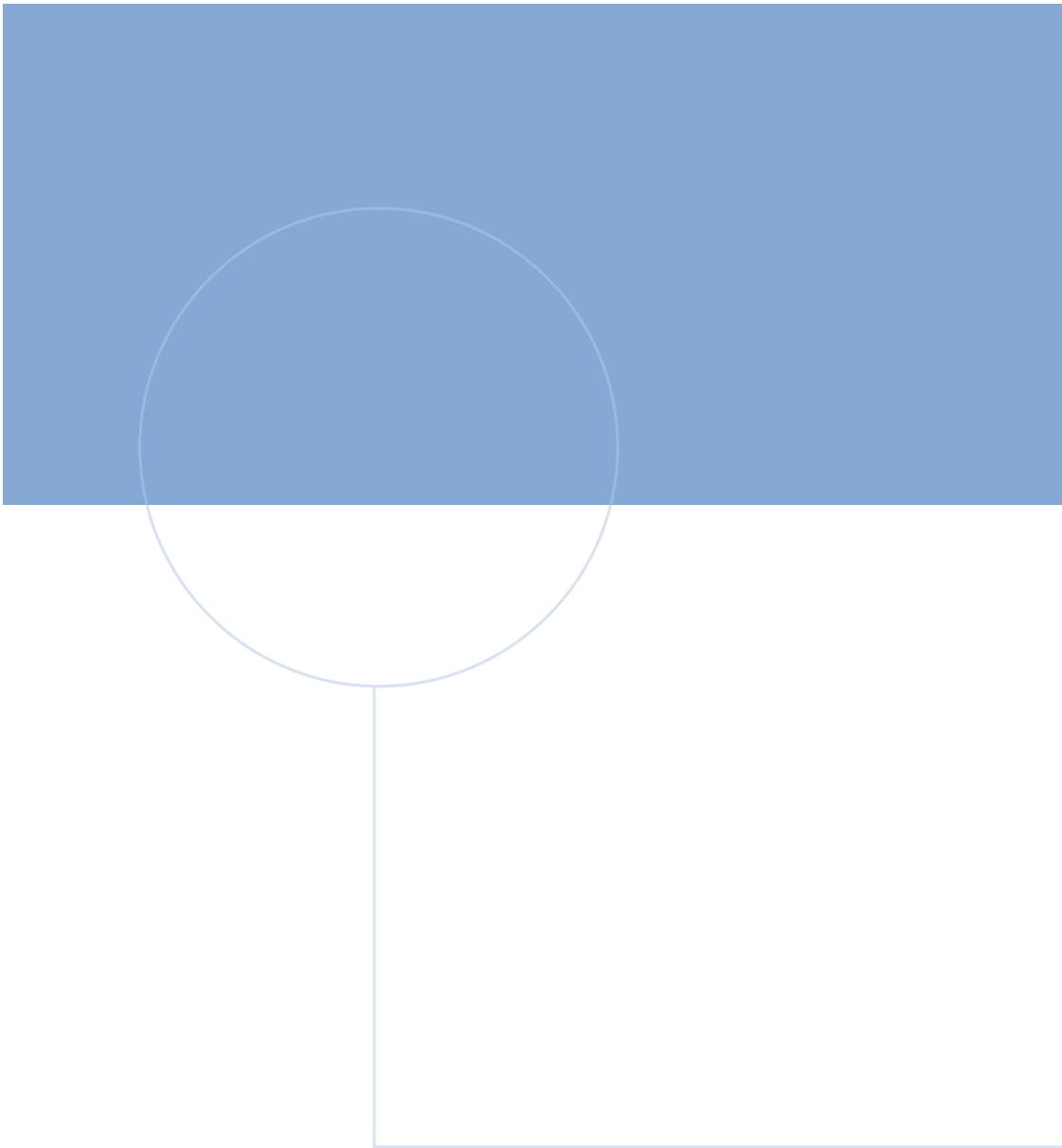
[46] M. Kaushik. Part 1: Backpropagation for convolution with strides, May 2019. https://medium.com/@mayank.utexas/backpropagation-for-convolution-with-strides-8137e4fc2710. Retrieved 03/02/2020.

[47] V. Spitzer, M. J. Ackerman, A. L. Scherzinger, and D. Whitlock. The visible human male: A technical report. *Journal of the American Medical Informatics Association*, 3(2):118–130, Jan 1996.

[48] A. Barchanski, M. Clemens, E. Gjonaj, H. D. Gersem, and T. Weiland. Large-scale calculation of low-frequency-induced currents in high-resolution human body models. *IEEE Transactions on Magnetics*, 43(4):1693–1696, 2007.

[49] Electromagnetic simulation software & em modeling. https://www.remcom.com/. Retrieved 02/06/2020.

[50] B. Archambeault, O. M. Ramahi, and C. Brench. The finite-difference time-domain method. *EMI/EMC Computational Modeling Handbook*, page 35–67, 1998.



NTNU

Norwegian University of
Science and Technology