Chapter 6

Simulator applications

The simulator can be applied in a number of diverse applications which span both MRI and FMRI fields. These applications include the simulation and removal of various imaging artefacts (such as $B_0$ inhomogeneities, rigid-body motion etc.), development of new pulse sequences, evaluation and validation of software tools for FMRI analysis methods (e.g. motion correction, registration, statistical analysis of images, etc.), or it can be used as an educational tool.

In this chapter five different applications of the simulator are presented. These are: quantifying the performance of motion correction software; investigating the performance of Independent Component Analysis (ICA) as a tool for quantifying motion-related artefacts; evaluating the impact of stimulus correlated motion artefacts; comparing different acquisition techniques for eddy-current compensation; and reproducing and extending experiments in neuronal current imaging.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

6.1 Quantifying the performance of a motion correction algorithm

In this section we focus on quantifying the performance of a motion correction algorithm. The motion correction algorithm in question is called MCFLIRT [6], [36] and is a part of FSL (FMRIB’s Software Library, www.fmrib.ox.ac.uk/fsl) [67].

6.1.1 Introduction

Motion correction is an important issue in FMRI analysis as even the slightest patient motion (1 - 2 degrees of rotation or a few mm translation) during a scan can mean that a voxel location does not correspond to the same physical point in the volume as the same voxel in a subsequent volume, given a spatial resolution of a few mm in the acquisition process. In addition, the change in signal intensity due to motion can be far greater than the BOLD effect itself, particularly at tissue boundaries, at the edge of the brain or near major ventricles. Before analysis can take place to get the activation patterns of different parts of the brain, the acquired images must be motion corrected in order for the physical and image coordinate systems to be coincident, and for the artefacts to be removed.

At present, most FMRI motion correction methods assume a simple model of rigid-body image transformation. In order for this approach to be accurate two assumptions are necessary. First is that the motion happened instantaneously and between the acquisition of the two neighbouring volumes (3D images) i.e. just after the acquisition of one volume finished and before the acquisition of the next volume started. This is descriptively shown in Figure 6.1, Level1. With this assumption, artefacts such as blurring and slice misalignment do not occur.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

Figure 6.1: Figure describes at which points in the pulse sequence we can assume that the motion has happened. The black line represents the image acquisition process in time. Red lines describe the motion occurring instantaneously between the acquisition of the volumes (Level1), blue lines describe the motion occurring instantaneously between the acquisition of the slices (Level2), and the green lines represent that motion is happening continuously throughout the whole image acquisition, including the acquisition of each of the slices (Level3).

The second assumption is that there is no interaction between motion and $B_0$-inhomogeneities. These motion correction methods typically compute the image transformation that will match a 3D image at time point $t$ to a reference template 3D image by optimising some cost function constructed to express image similarity. Commonly these type of methods are classified as image-similarity-based motion correction algorithms. Some commonly used ones are the one of Friston et. al. ([25], [26]) in the Statistical Parametric Mapping (SPM) package, the one described by Woods et. al. ([70]) in the Automated Image Registration (AIR) package and the one described by Jenkinson et.al [36], [6] in the FMRIB’s Software Library (FSL) package [67]

Motion correction algorithms are one of the necessary parts of every FMRI data analysis package. However, most of the motion correction techniques that are developed are based on analysis of empirically acquired FMRI data, so that the ground truth is not known and therefore the precise accuracy of the methods difficult to estimate (e.g. see [32, 25, 41, 36]).

Accuracy testing of motion correction algorithms is usually done by the use of simulated data. These data are usually constructed by taking a real image from the scanner and applying 3D transformations to it in order to generate
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

the motion corrupted time-series. Due to this approach, some of the important motion artefacts are missed out and as a result the data is often simple and unrealistic making the full evaluation of the performance of the motion correction methods very difficult.

The simulator has the capability to simulate realistic images with complex motion-related artefacts such as spin history or blurring. In addition, the simulator does this in a controlled way, i.e. various artefacts can be simply turned “on” or “off”. Due to these properties, the simulator can be used for testing image analysis softwares. In this section we focus on quantifying the performance of the motion correction algorithm MCFLIRT [6], [36].

An accuracy assessment of this software was done by Bannister [6], however the simulated data used for the assessment was empirically derived. A high-resolution EPI volume (2 × 2 × 2mm voxel dimensions) was duplicated 180 times and each volume was transformed by an affine matrix corresponding to real motion estimates taken from an experimental study. The images were then sub-sampled to 4 × 4 × 6mm voxels. It is expected that the effect of interpolation when applying the transformations to the images will be minimised with this approach of applying the transformations to a high-resolution images first and then sub-sampling them. However the effects of the interpolation can never be completely avoided.

Furthermore, the images were generated by applying one affine transformation for each of the time points, as if motion happened instantaneously between the acquisition of volumes acquired at the neighbouring time-points (Level1 motion). This approach is sufficient for testing the accuracy of the MCFLIRT algorithm in ideal conditions but is not sufficient to evaluate the performance of the algorithm in realistic scanning conditions and with data that has realistic and more complex motion effects.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

By using the simulator it is possible to address both of these issues. Firstly, the problem of interpolation would be avoided. Secondly, the simulator is capable of simulating all of the motion artefacts represented in Figure 2.18 and on any of the levels represented in Figure 6.1. This provides excellent grounds for evaluation of the performance of the MCFLIRT algorithm in realistic, but still controllable situation. In order to show the approach used to do this, a brief description of the MCFLIRT algorithm is needed and is given in the next section.

6.1.2 MCFLIRT Algorithm

The MCFLIRT algorithm is an image-similarity-based motion correction algorithm. In order to correct the motion-corrupted images, first a reference image is chosen from the time-series. Which volume is taken to be the reference image is a user-defined option. The two most common options are the middle image (default option in MCFLIRT) and the first image. All of the other images are then registered to the reference image. The registration process is computed by constructing a cost function which quantifies the dissimilarity between the two images and then searches for the transformation which gives the minimum cost value.

The method used in MCFLIRT for the process of minimisation is known as golden section search [6]. Golden section search is performed on each of the 6 rigid-body parameters in turn. It is performed for the rotational parameters first as these have a greater impact on the overall movement of the object than the translational parameters (e.g. 2 degrees rotation centered in the middle of the brain results in approximately 4mm translation at the edge of the brain) [6]. Searching in each of the 6 parameter directions stops when a specified tolerance has been reached. The default tolerances in MCFLIRT are 0.057 degrees for the
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

rotations and 0.02mm for the translations.

In the MCFLIRT algorithm a cost function is selected by the user and can be: mutual information, correlation ratio, normalised correlation, normalised mutual information or least squares. Each cost function is regularised via the introduction of geometric smoothing. This is done in order to avoid small discontinuities as the transformation parameters are varied smoothly which can create local minima “traps”. The default cost function used in MCFLIRT is normalised correlation.

When calculating the dissimilarity between the reference image ($I_r$) and the image we want to register to the reference image ($I$), interpolation needs to be applied to $I$. This is a process that calculates the intensity in the image $I$ at points between the original image voxel points. The interpolation choices in MCFLIRT are: trilinear, sinc and nearest neighbour. Different choices of the interpolation function are used for different stages of the algorithm.

The registration process is done in a few different stages. The first stage is performed on data which is sub-sampled to 8mm isotropic voxels to allow the gross intensity patterns in the images to drive the registration. The optimisation during this stage is performed using trilinear interpolation and the stopping tolerance for the golden search process is specified to be relatively large (0.45 degrees for the rotations, and 0.16mm for the translations). The second stage is performed on 4mm isotropic voxels, using trilinear interpolation and the same tolerance level as the one defined during the first stage. During the third stage the size of the voxels is maintained at 4mm, the interpolation is still trilinear but the tolerance is reduced to the default values (0.057 degrees for the rotations and 0.02mm for the translations). These three stages are the default stages of the MCFLIRT algorithm. An extra fourth stage is optional, and is the same as the third stage except that the sinc interpolation method is used instead of the trilinear interpolation method.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

Once the minimisation process is finished and all of the transformation parameters are estimated, a transformation is applied to the image. This transformation can be applied using either trilinear or sinc or the nearest neighbour interpolation method (the default in MCFLIRT is trilinear interpolation).

6.1.3 Methods

In this work, two different aspects of MCFLIRT performance are evaluated:

- Accuracy of the estimated motion parameters,
- Accuracy of the application of the final transformation.

Methods which were used to do so are described in this section. Firstly, simulated data used for the evaluation purposes is described. Following from there, various MCFLIRT options and test measures are described for both of the two different aspects of MCFLIRT performance.

Simulated data

Data was simulated in such a way as to incorporate various levels of complexity at different stages. More specifically, the five different groups of simulated data were:

Group 1: Level1 motion, no $B_0$ inhomogeneities, without noise. This group of simulations will be affected by the position displacement and also by the spin history artefact (which is unavoidable when using realistic motion sequences). Motion parameters (rotations and translations) were extracted from a patient study done by Hill et al. [34] and are shown in Figure 6.2.

The step-wise look of the motion parameters is due to the fact that the motion is defined, for this simulation, to happen instantaneously and between the
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

![Figure 6.2: Level1 motion used in the simulations for Group1 and Group2.](image)

acquisition of neighbouring volumes. Simulations were firstly done for motion paradigms where only one parameter is changing and all others are zero. The waveform of the motion parameter that is changing was taken from Figure 6.2. In addition to this, simulations were done when all of the motion parameters are changing together (following the pattern represented in 6.2).

**Group2: Level1 motion, with $B_0$ inhomogeneities, without noise.** This group of simulations will, in addition to all of the artefacts of Group1, also add $B_0$ inhomogeneities. Motion parameters and the simulation method is the same as the one described for the previous group.

**Group3: Level2 motion, with $B_0$ inhomogeneities, without noise.** This group will, in addition to all of the artefacts of Group2, also be affected by the slice misalignment. The motion parameters are shown in Figure 6.3. The difference between this and the motion file used for Group1 and Group2 is that this motion is also happening instantaneously, but between the acquisition of the consecutive slices. The motion parameters for each of the slices were interpolated
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

![Graphs showing motion parameters](image)

Figure 6.3: Level2 motion used in the simulations for Group3.

using piecewise cubic spline interpolation from motion parameters used in the Group1.

**Group4: Level3 motion, with $B_0$ inhomogeneities, without noise.** This group will, in addition to all of the artefacts of Group3, also be affected by blurring in the images. The motion parameters used for this group are shown in Figure 6.4. They represent continuous motion. The motion parameters for each of the time points were interpolated using piecewise cubic spline interpolation from motion parameters used in Group1. Simulations were done for small, medium and large motion. Medium motion is the one represented in Figure 6.4 (with the maximal motion of 0.8mm for the translations and 0.8 degrees for the rotations), small motion is the medium motion divided by 5 (maximal motion of 0.16mm for the translations and 0.16 degrees for the rotations), and the large motion is the medium motion multiplied by 3 (maximal motion of 2.4mm for the translations and 2.4 degrees for the rotations).
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

![Graphs showing motion in different directions](image)

Figure 6.4: Level3 motion used in the simulations for Group4 and Group5.

**Group5: Level3 motion, with $B_0$ inhomogeneities, with noise.** This group will, in addition to all of the artefacts of Group4, be affected by the noise in the images. The only difference in the simulations from the previous group is that the various noise levels are added to the image. Standard Signal to Noise (SNR) values for single shot EPI images range from 100 to 200 [13]. Simulations were done for SNR=100, SNR=150 and SNR=200.

By separating the groups of simulations like this we hope that we can measure the influence that each of the motion effects has on MCFLIRT performance.

All of the simulations were performed using the McGill virtual brain with the parameter values given in Table 4.1, at 3T magnetic field strength, and an EPI pulse sequence which we generated with parameters: TE = 30ms, TR = 3s, flip angle 90°, maximum gradient strength 35mT/m, rise times 0.2ms, BW = $100KH_z$ crushers of maximum magnitude, output voxel size $4 \times 4 \times 7$mm, 25 slices and 20 volumes. In the cases when the $B_0$ inhomogeneity field was simulated, a set of 9 basis volumes are passed into the simulator. Slice acquisition order ran in the -z direction.

Once the data sets were generated, they were motion-corrected using various
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

options of MCLFIRT. As mentioned at the beginning of this section, two different aspects of motion correction are looked at: Accuracy of the estimated motion parameters; and Accuracy of the application of the final transformation. These are, together with test measures used to assess the accuracy of the results, described in the two following sections.

Accuracy of the estimated motion parameters

The accuracy of the transformation estimation was tested for a specific choice of cost functions, interpolation functions and other options as described in the following.

Two different cost functions are investigated: normalised correlation (the current default option) and correlation ratio. These two were chosen as they were found to be the two most successful in the accuracy assessment done by Bannister [6] (with a slight preference towards the normalised correlation).

Regarding the number of stages in the optimisation process, the current default one (three stage process using trilinear interpolation) was compared to the four stage process with an extra optimisation step using the sinc interpolation (as described in the previous section). The tolerance levels chosen for the golden sec-

<table>
<thead>
<tr>
<th>Stages</th>
<th>Ref Image</th>
<th>Tolerance</th>
<th>Cost function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrilF</td>
<td>3</td>
<td>First</td>
<td>default</td>
</tr>
<tr>
<td>TrilM</td>
<td>3</td>
<td>Middle</td>
<td>default</td>
</tr>
<tr>
<td>TrilMTol</td>
<td>3</td>
<td>Middle</td>
<td>reduced</td>
</tr>
<tr>
<td>TrilMCr</td>
<td>3</td>
<td>Middle</td>
<td>default</td>
</tr>
<tr>
<td>SincF</td>
<td>4</td>
<td>First</td>
<td>default</td>
</tr>
<tr>
<td>SincM</td>
<td>4</td>
<td>Middle</td>
<td>default</td>
</tr>
<tr>
<td>SincMTol</td>
<td>4</td>
<td>Middle</td>
<td>reduced</td>
</tr>
<tr>
<td>SincMCr</td>
<td>4</td>
<td>Middle</td>
<td>default</td>
</tr>
</tbody>
</table>

Table 6.1: Eight different combinations of MCFLIRT options that were tested in the experiments for testing the accuracy of the estimated parameters.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

Motion search were also looked at. The results generated by the current default one (0.057 degrees for the rotations and 0.002mm for the translations) were compared to results obtained using the new reduced one (0.0057 degrees for the rotations and 0.0002mm for the translations).

In addition to this, a comparison was done for two different choices of the reference image. The choice of the middle image in the time-series (the current default option) is compared to the choice of the first image in the time-series.

In summary, results were generated for eight different choices of motion correction options which are shown in Table 6.1. In order to assess the accuracy of these options two different test measures are used.

Test measure for Group1 and Group2: For the first two groups of the simulated data sets motion was instantaneous and happened between the acquisition of neighbouring volumes. From this it follows that the volumes themselves are only affected by the affine transformations and therefore can be completely corrected by applying the MCFLIRT algorithm. As the motion parameters for the simulations are known, it is possible to compare them with those estimated by the MCFLIRT algorithm. In order to calculate the difference between these two affine transformations (the true one, and the one estimated by MCFLIRT) a root-mean-square (RMS) average metric is employed [35].

The RMS average metric is calculated on the transformed coordinates of the volume. More specifically, when comparing two transformations, $T_1$ and $T_2$, each mapping volume A to volume B (the reference volume), if the transformations are identical then the points in volume B will be the same. However, in general the transformations differ and so map $x_A$ to two different points: $x_{B1}$ and $x_{B2}$. The vector difference between these points, $\Delta x = x_{B2} - x_{B1}$, represents the error in the transformation. It is the average magnitude of this error that is of interest.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

This method of calculating the error is more suited than the other commonly used method of calculating the error directly on the transformation parameters. The reason for this is that when calculating RMS on rotation parameters directly we get the difference in degrees which is the same everywhere no matter how big the brain is. When calculating the difference in transformed coordinates over the whole brain we take that into account (for example for some brains 2 degrees in the centre is 4mm on the edge, and for some it is 3mm or 5mm etc.). In addition, for measuring the average error over the brain volume a spherical volume is chosen. This is done because it is the simplest and the closest approximation of the brain volume. Using cubic volume generates larger errors as the rotational error increases with increasing the distance from the centre of the cube (specially affected are the corners as they are the furthest).

In general, the error vector can be written as \( \Delta x = Mx \) and then can be used to give the squared error:

\[
E^2 = |\Delta x|^2 = (\Delta x)^\top (\Delta x) = x^\top M^\top Mx.
\]

The normalised RMS error is then given by:

\[
E_{RMS}^2 = \frac{\int_V E^2 \, dx}{\int_V dx}.
\]

Changing the variables into the spherical coordinates, and integrating over the whole spherical volume the RMS error becomes:

\[
E_{RMS} = \sqrt{\frac{1}{5} R^2 \text{Trace}(A^\top A) + (t + Ax_c)^\top (t + Ax_c)}
\]

where \( x_c \) is the center of the volume of interest, \( R \) is the radius of the spherical
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

The volume encompassing the brain,

\[ M = T_2 T_1^{-1} - I = \begin{bmatrix} A & t \\ 0 & 0 & 0 \end{bmatrix} \]  (6.6)

and \( T_1 \) and \( T_2 \) are the transformations that are being compared i.e. MCFLIRT estimated transformation and the true transformation used to create the simulations. This measure gives an estimate in real units (mm) of the difference in transformations. More details on solving this equation are described in [35]. Once the RMS difference for each of the volumes is known, a mean value and a standard deviation for all of the volumes is calculated.

**Test measure for Group3, Group4 and Group5:** Simulations in Group3, Group4 and Group5 need to be treated differently to the simulations in Group1 and Group1. The object is moving *during* the acquisition of the volumes, creating effects in 3D images which cannot be fully compensated for by a pure affine transformation. This is because the slices will not be aligned anymore and no single 3D transformation can fix that. MCFLIRT only handles affine transformations on 3D images in its correction process, ignoring the other effects which are due to, say, slice-misalignment or blurring. Therefore validating MCFLIRT with respect to this kind of motion is unfair as MCFLIRT does not claim to be doing these corrections anyway. What is, however, possible to do is to evaluate its performance in these kind of realistic situations.

RMS difference is calculated in a very similar fashion as described above for the previous test measure. The first important difference here is that there is no one affine transformation that can correct the image (this is due to the slice-misalignment). Therefore, the true motion parameters are used to construct a separate transformation for each of the slices in the volume, and RMS difference is calculated for each of these slices individually. The second important difference
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

is that the RMS is calculated numerically and not analytically like in the previous case. One MCFLIRT and one true transformation were applied on each voxel of slice and the distance between the transformed voxels (in mm) was calculated. This distance is then squared, and the sum over all of the squared distances calculated. The mean of this sum is found and its squared root taken:

\[ E_{RMS} = \sqrt{\frac{1}{N} \sum_{x \in S} \| T_1(x) - T_2(x) \|^2} \] (6.7)

where \( S \) stands for a slice, \( N \) is the number of voxels in the slice and \( T_1 \) and \( T_2 \) are the transformations that are being compared. Once the RMS difference for each of the slices is known, a mean value and a standard deviation for all of the slices over all of the volumes is calculated.

**Accuracy of the application of the final transformation**

The accuracy of application of the transformation matrix was tested for different interpolation techniques: trilinear (the current default in MCFLIRT), nearest neighbour and sinc interpolation. Results were generated for three different choices of motion correction options which are shown in Table 6.2. In order to

<table>
<thead>
<tr>
<th>Stages</th>
<th>Ref Image</th>
<th>Tolerance</th>
<th>Cost function</th>
<th>Final interp</th>
</tr>
</thead>
<tbody>
<tr>
<td>NearN</td>
<td>4</td>
<td>First</td>
<td>default</td>
<td>Norm corr</td>
</tr>
<tr>
<td>Tril</td>
<td>4</td>
<td>First</td>
<td>default</td>
<td>Norm corr</td>
</tr>
<tr>
<td>Sinc</td>
<td>4</td>
<td>First</td>
<td>default</td>
<td>Norm corr</td>
</tr>
</tbody>
</table>

Table 6.2: Three different combinations of MCFLIRT options that were tested in the experiments for testing the accuracy of the application of the transformation.

assess the accuracy of MCFLIRT the following test measure is used:

**Test measure:** This is a measure applied directly on the corrected images and not on the parameter estimates as it was the case with the measures used when
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

accuracy of the estimated motion parameters was tested. A simple difference between the motion corrected image $I_1$ and the motionless image $I_2$ was calculated and divided by the averaged motionless image as shown in the equation:

$$E = 100 \frac{(I_1 - I_2)}{I_2^u} \quad (6.8)$$

where 100 transforms the result into percentages. In order to get further insight into the characteristics of this error, the minimum, 10th percentile, mean, median, 90th percentile and the maximum value of the error over all of the volumes are summarised.

6.1.4 Results and discussion

**Accuracy of the estimated motion parameters**

Figures 6.5, 6.6 and 6.7 show the results of motion correction for Group1 (Level1 motion, no $B_0$ inhomogeneities) and Group2 (Level1 motion, with $B_0$ inhomogeneities) of simulated data. On the left hand side of the figures, the calculated RMS difference between the true and MCFLIRT estimated transformations is shown. Green lines represent the results for the case with no $B_0$ inhomogeneities (Group1), while black lines represent the case with $B_0$ inhomogeneities (Group2). On the right hand side of the figures, the value of the estimated motion parameter is shown. The true value which was used in the simulations is in blue, while each of the other lines represents motion correction done with a combination of options described in the legend.

It can be observed from figures 6.5, 6.6 and 6.7 that the overall accuracy of the MCFLIRT estimation of the motion parameters in the case of single motion parameter change for Group1 and Group2 (the motion happens instantaneously and between the acquisition of the neighbouring volumes) is good. The mean error does not exceed 0.08mm (for translations), and 0.3mm (for rotations).
Figure 6.5: RMS mean error results for the MCFLIRT motion correction algorithm when applied to data simulated with only one motion parameter changing: Translation in x (top row); Translation in y (middle row); Translation in z (bottom row). The data belongs to Group1 (green lines) and Group2 (black lines). The left hand side plots show the error bars which indicate the standard error. The right hand side shows the MCFLIRT estimation of the motion parameter that is changing. The true values of that motion parameter (the ones used in the simulations) are shown with a blue line in each of the plots.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

Figure 6.6: RMS mean error results for the MCFLIRT motion correction algorithm when applied to data simulated with only one motion parameter changing: Rotation about x (top row); Rotation about y (middle row); Rotation about z (bottom row). The data belongs to Group1 (full lines) and Group2 (dashed lines). The left hand side plots show the error bars which indicate the standard error. The right hand side shows the MCFLIRT estimation of the motion parameter that is changing. The true values of that motion parameter (the ones used in the simulations) are shown with a blue line in each of the plots. Note that the scale for the error plots is larger than the scale used in the translation error plots shown in Figure 6.5.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

Figure 6.7: RMS mean error results for the MCFLIRT motion correction algorithm when applied to data simulated with all of the motion parameters changing. The top plot shows the error bars which indicate the standard error for Group1 (full lines) and Group2 (dashed lines). Other plots show MCFLIRT estimation of the motion parameters. The true values (the ones used in the simulations) are shown with a blue line in each of the plots.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

It can also be seen from Figure 6.7 that each individual estimated motion parameter from the simulated data when all of the motion parameters are changing, was matched very well with the estimated motion parameters from the data when only one motion parameter is changing (shown in figures 6.5, 6.6).

A few differences can be seen between different motion correction options. Firstly, regarding the choice of the cost function, it was shown that the motion correction using the corelation ratio cost function did not perform as well as the one using the normalised correlation. This confirms the results of validation done by Bannister [6], and further justifies the use of the normalised correlation cost function in the MCFLIRT algorithm.

Regarding the number of stages in the optimisation process, the current default one (three stage process using trilinear interpolation) was compared to the four stage process with an extra optimisation step using the sinc interpolation. The results suggest considerable improvement when using four stages in the motion correction process. This improvement was stable across all of the different parameters, including the situation when the motion was quite complex, involving a change in all of the parameters.

The results regarding the tolerance levels chosen for the golden section search, suggests that the current default tolerance level (0.057 degrees for the rotations and 0.002mm for the translations) performed as well as the reduced tolerance level (0.0057 degrees for the rotations and 0.0002mm for the translations). This suggests that the current default tolerance level in the MCFLIRT is sufficient for the parameter estimation.

In addition to this, results regarding the different choices of the reference image (first image versus the middle image in the time series) are also shown. The choice of a reference image did not seem to impact results significantly or in any predictable way. It is possible that the shape of the motion paradigm influences
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

the preferable reference image. It would be interesting to further analyse the relationship between the two.

In addition to the conclusions drawn about the various motion correction options, a significant observation was made regarding the \( B_0 \) inhomogeneities. It can be seen in all of the figures (Figures 6.5, 6.6 and 6.7) that the presence of the magnetic-susceptibility-induced \( B_0 \) inhomogeneities (black lines) did not make a significant difference in the estimation of the motion parameters. This is quite an interesting and new observation. The error did increase in most of the cases, but this increase was not as big as what might be intuitively expected.

The results for the simulated data of Group4 and Group5 are shown in Figure 6.8, on the left and right hand side respectively. For the left plot, RMS error was calculated for the simulations of different motion intensities: red line representing the error for the large motion, blue line representing the error for the average motion, and black line representing the error for the small motion. It can be seen from the plot that as the motion level increases the size of the error also increases as expected. An interesting observation is that the amount of this increase is (almost) linearly proportional to the increase in the motion level. More specifically, the Medium motion and the Small motion are different by a factor of 5 while their respective mean RMS errors are (on average) different by a factor of 4.19. Also, the Medium motion and the Large motion are different by a factor of 3 while their respective mean RMS errors are (on average) different by a factor of 2.45.

The right hand side plot in Figure 6.8 shows the RMS error calculated for the simulations of Group5 with different noise levels: SNR=100 (red line), SNR=150 (black line) and SNR=200 (green line). The results show that the difference in noise level did not significantly influence the accuracy of estimated parameters.

A summary of the impact of different levels of complexity of the data on the
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

Figure 6.8: RMS mean error results for the MCFLIRT motion correction algorithm when applied to: simulated data of Group4, Large, Medium and Small motion, no noise (left); simulated data of Group5, Medium motion, SNR=100, SNR=150, SNR=200 (right). Both Group4 and 5 are simulated with the motion that varies continuously throughout the image acquisition and with $B_0$ inhomogeneities.

RMS error results is shown in Figure 6.9.

Different levels of motion had different impacts on the accuracy of parameter estimation. Motion that happens instantaneously between the volume acquisitions (Level1), was the easiest to correct and the error was the smallest. The presence of $B_0$ inhomogeneities did create a difference but this difference was not prominent.

Motion that happens instantaneously between the slice acquisitions (Level2), adds an extra motion artefact to the images: slice misalignment. From the results it is concluded that the slice misalignment created a big difference (by a factor of 1.7) in parameter estimation.

Motion that happens continuously throughout the image acquisition (Level3), adds (on the top of Level2 motion) an extra motion artefact to the images: blurring. From the results however it is concluded that blurring does not considerably change the accuracy of parameter estimation. The accuracy does not change much either when noise is added to the images (SNR=100).
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

Figure 6.9: RMS mean error results for the MCFLIRT motion correction algorithm when applied to simulated data of Group1 (green line), Group2 (black line), Group3 (red line), Group4 (blue line), Group5 (magenta line).

Accuracy of the application of the final transformation

Results of the experiment testing the accuracy of MCFLIRT algorithm regarding the application of the estimated transformation at the final stage of the algorithm are represented in Table 6.3. The error was calculated for Group 1 and 2 simulations (one representative for the translation data, one for the rotation data and one for “All”, when all the parameters are changing) and is represented in the first half of the table. The error calculated for the group 3, 4 and 5 simulations is shown at the bottom of the table. The error was calculated using the test measure described in the Methods section.

Results presented in Table 6.3 show that the choice of the interpolation function when resampling the image (in the final stage of the MCFLIRT algorithm)
### 6.1. Quantifying the Performance of a Motion Correction Algorithm

<table>
<thead>
<tr>
<th>Group, Type</th>
<th>min</th>
<th>10th perc</th>
<th>50th perc</th>
<th>mean</th>
<th>90th perc</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group1, Tx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NearN</td>
<td>0</td>
<td>0.000178</td>
<td>0.014675</td>
<td>0.192475</td>
<td>0.448458</td>
<td>18.131998</td>
</tr>
<tr>
<td>Tril</td>
<td>0</td>
<td>0.000138</td>
<td>0.014644</td>
<td>0.166135</td>
<td>0.429214</td>
<td>17.592768</td>
</tr>
<tr>
<td>Sinc</td>
<td>0</td>
<td>0.000014</td>
<td>0.010066</td>
<td>0.112660</td>
<td>0.306298</td>
<td>14.459094</td>
</tr>
<tr>
<td>Group2, Tx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NearN</td>
<td>0</td>
<td>0.000209</td>
<td>0.019885</td>
<td>0.194751</td>
<td>0.444200</td>
<td>19.272722</td>
</tr>
<tr>
<td>Tril</td>
<td>0</td>
<td>0.000185</td>
<td>0.019118</td>
<td>0.165766</td>
<td>0.417982</td>
<td>18.684446</td>
</tr>
<tr>
<td>Sinc</td>
<td>0</td>
<td>0.000015</td>
<td>0.013760</td>
<td>0.115542</td>
<td>0.306869</td>
<td>14.970655</td>
</tr>
<tr>
<td>Group1, Rx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NearN</td>
<td>0</td>
<td>0.188789</td>
<td>0.010270</td>
<td>0.188789</td>
<td>0.429271</td>
<td>25.985754</td>
</tr>
<tr>
<td>Tril</td>
<td>0</td>
<td>0.000231</td>
<td>0.010820</td>
<td>0.190849</td>
<td>0.469642</td>
<td>15.738885</td>
</tr>
<tr>
<td>Sinc</td>
<td>0</td>
<td>0.000036</td>
<td>0.007971</td>
<td>0.141556</td>
<td>0.370623</td>
<td>13.398987</td>
</tr>
<tr>
<td>Group2, Rx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NearN</td>
<td>0</td>
<td>0.000236</td>
<td>0.012326</td>
<td>0.197366</td>
<td>0.447132</td>
<td>27.397284</td>
</tr>
<tr>
<td>Tril</td>
<td>0</td>
<td>0.000226</td>
<td>0.013160</td>
<td>0.187879</td>
<td>0.460936</td>
<td>24.340437</td>
</tr>
<tr>
<td>Sinc</td>
<td>0</td>
<td>0.000034</td>
<td>0.009653</td>
<td>0.145426</td>
<td>0.379373</td>
<td>20.820518</td>
</tr>
<tr>
<td>Group1, All</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NearN</td>
<td>0</td>
<td>0.000937</td>
<td>0.044183</td>
<td>0.490567</td>
<td>1.121255</td>
<td>50.736828</td>
</tr>
<tr>
<td>Tril</td>
<td>0</td>
<td>0.000960</td>
<td>0.046705</td>
<td>0.454624</td>
<td>1.160782</td>
<td>27.538780</td>
</tr>
<tr>
<td>Sinc</td>
<td>0</td>
<td>0.000151</td>
<td>0.033376</td>
<td>0.308924</td>
<td>0.834392</td>
<td>19.972572</td>
</tr>
<tr>
<td>Group2, All</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NearN</td>
<td>0</td>
<td>0.011182</td>
<td>0.059673</td>
<td>0.516598</td>
<td>1.190839</td>
<td>51.670368</td>
</tr>
<tr>
<td>Tril</td>
<td>0</td>
<td>0.001290</td>
<td>0.062503</td>
<td>0.456827</td>
<td>1.164064</td>
<td>29.679621</td>
</tr>
<tr>
<td>Sinc</td>
<td>0</td>
<td>0.000179</td>
<td>0.046147</td>
<td>0.325090</td>
<td>0.870691</td>
<td>28.055516</td>
</tr>
<tr>
<td>Group3, All</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NearN</td>
<td>0</td>
<td>0.000970</td>
<td>0.060904</td>
<td>0.507177</td>
<td>1.149266</td>
<td>48.778713</td>
</tr>
<tr>
<td>Tril</td>
<td>0</td>
<td>0.000981</td>
<td>0.063039</td>
<td>0.449183</td>
<td>1.119420</td>
<td>31.585869</td>
</tr>
<tr>
<td>Sinc</td>
<td>0</td>
<td>0.000138</td>
<td>0.047097</td>
<td>0.317496</td>
<td>0.828243</td>
<td>30.682161</td>
</tr>
<tr>
<td>Group4, All</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NearN</td>
<td>0</td>
<td>0.000970</td>
<td>0.060969</td>
<td>0.507427</td>
<td>1.149931</td>
<td>48.751251</td>
</tr>
<tr>
<td>Tril</td>
<td>0</td>
<td>0.000988</td>
<td>0.063085</td>
<td>0.449275</td>
<td>1.119950</td>
<td>31.431250</td>
</tr>
<tr>
<td>Sinc</td>
<td>0</td>
<td>0.000138</td>
<td>0.047163</td>
<td>0.317567</td>
<td>0.827249</td>
<td>30.543518</td>
</tr>
<tr>
<td>Group5, All</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NearN</td>
<td>0</td>
<td>0.074081</td>
<td>0.413926</td>
<td>0.760446</td>
<td>1.396072</td>
<td>49.125244</td>
</tr>
<tr>
<td>Tril</td>
<td>0</td>
<td>0.068711</td>
<td>0.380343</td>
<td>0.681543</td>
<td>1.336535</td>
<td>31.405405</td>
</tr>
<tr>
<td>Sinc</td>
<td>0</td>
<td>0.059400</td>
<td>0.386406</td>
<td>0.599571</td>
<td>1.225205</td>
<td>30.497026</td>
</tr>
</tbody>
</table>

Table 6.3: Representing the error (in percentages of the average image intensity) which occurred when the final transformation was applied to the images. Results for three different interpolation techniques and for different levels of simulated data complexity (increasing from the top to the bottom) are given.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

was important. Use of sinc interpolation was the best option out of the three interpolation techniques (trilinear interpolation, nearest neighbour, and sinc). The second best option was the use of the trilinear interpolation (which is the current default option in MCFLIRT) while the use of the nearest neighbour proved to be the worst.

It can also be observed from Table 6.3 that the overall accuracy of the MCFLIRT application of the final transformation decreases as the simulations become more realistic. When only one motion parameter is changing, and the motion is happening instantaneously between the volume acquisitions (Group1, top of the table) correction of translations was done more accurately than correction of rotations, which was expected. For the case of a single parameter changing, the error levels go even as high as 19% but the minimum, mean, maximum and the 90th percentile are overall less than 0.3% (when using sinc interpolation) of the average image intensity. The situation is worse when all of the parameters are changing with the error increasing more than twice compared to the previous cases. In this case, the 90th percentile of the error is 0.83% (when using sinc interpolation) of the average image intensity. As the simulations become more realistic the error increases. In the most realistic case of continuous motion (Level3 motion) with $B_0$ inhomogeneities and noise (SNR 100), 90th percentile of the error increases to 1.22%.

It can also be observed from Table 6.3 that the magnetic-susceptibility-induced $B_0$ inhomogeneity did not make a big impact when evaluating the accuracy of the resampling techniques in the final stage of the MCFLIRT algorithm (difference between the groups 1 and 2 in the table). This is slightly suprising and very similar to what is found in the previous section.
6.1. QUANTIFYING THE PERFORMANCE OF A MOTION CORRECTION ALGORITHM

6.1.5 Conclusions

The MCFLIRT algorithm showed good performance when tested for various types of simulated data. This was specifically true for estimating the motion parameters. It was found that four stages of optimisation when estimating the motion parameters, with the fourth stage using sinc interpolation, performed significantly better than the three stage option which is the current default in MCFLIRT. In addition to this, it was found that the final transformation is significantly more accurate when applied using the sinc interpolation than when applied using the trilinear interpolation which is the current default option in MCFLIRT.

It was found that by increasing the complexity levels of the simulated data (with non-rigid deformations due to motion) the MCFLIRT performance became worse which is logical and expected as MCLFIRT assumes simple rigid-body transformation of the object. It was found that slice misalignment strongly impacts the parameter estimation while, $B_0$ inhomogeneity, blurring and noise do not.

In addition to finding more accurate choices of some of the existing MCFLIRT options, the results suggest that the main step forward in improving the MCFLIRT algorithm is including the model for correction of the distortion due to slice misalignment.
6.2 Quantifying motion artefacts with ICA

This section describes an application of the simulator in investigating the performance of ICA as an aid to motion correction. The results presented in this section were done in collaboration with Christian Beckmann and are published in Drobnjak et al. [21].

6.2.1 Introduction

FMRI data contain many motion-related artefacts which can limit its usefulness. Many approaches exist to correct for rigid-body motion effects but they struggle to do so in cases when motion has more complex impact on the data (e.g. spin history, slice mis-alignment, or interaction of motion and the $B_0$ field inhomogeneities). Independent Component Analysis (ICA) on FMRI data can be used to spatially and temporally identify these artefacts, and then by doing ICA denoising (or some other advanced pre-processing method) remove the artefacts from the data.

However, in order for ICA-based tools for motion correction to be used, the accuracy and usefulness of ICA for motion artefact identification and quantification needs to be tested with data which contains realistic artefacts and for which the ground truth is known. This section shows the performance of ICA on simulated data which includes rigid-body motion effects for in-plane rotations, including the interactions with $B_0$ inhomogeneities.

6.2.2 Methods

The data was simulated using an EPI pulse sequence with 4mm in-plane resolution ($64 \times 64$ voxels), nine 3mm slices, TR=3s, TE=30ms for 98 volumes. The BrainWeb partial volume tissue estimates were used as the object model. $T_2^*$
time courses were derived by fitting the equation $S = S_0 \exp(-TE/T_2^*)$ to an experimentally acquired FMRI data set (from Montreal Neurological Institute, McGill University), where $S$ is the measured signal intensity, $TE$ is the echo time and $S_0$ is the baseline intensity (determined from the average value of $S$ under baseline conditions). The BOLD changes modelled by this $T_2^*$ change include stimulus-related activations as well as those of no interest (“physiological noise”).

Figure 6.10: Rotation about the z-axis (degrees). Time is in seconds.

Figure 6.10 shows the change in motion parameter (rotation about the z-axis up to 0.8 degrees) over time. Four separate simulations were generated: (1) no motion; (2) motion represented in Figure 6.10 which is an average of 0.046 degrees per TR; (3) three times this motion which is 0.138 degrees per TR; and (4) five times this motion which is 0.23 degrees per TR. Rician noise (SNR=100) was added to all of the simulation outputs and a motion correction algorithm (MCFLIRT), was applied to all of them. ICA was carried out using MELODIC [7].
6.2.3 Results and discussion

There were 14 different IC components estimated by MELODIC in each of the simulations. Figure 6.11 shows two example components due to motion. Both of the components show effects of the global rotational motion as well as the motion-$B_0$ field interaction, which can be seen around the edges of the brain and the $B_0$-effected areas. The first component shows more localization in the $B_0$-affected areas and therefore is predominantly associated with the motion-$B_0$ interactions. Furthermore, the time courses of the two components are different, as the first closely resembles the original input motion, while the second one does not. This suggests that the influence of the motion on the data is much more complex and does not have a simple relationship with the motion parameter time-course.

In Figure 6.12 the total standard deviation explained by the ICA components in the 4 datasets can be seen. The results in this figure show that the non-motion related components had a very similar standard deviation in each case, while the level of residual motion artefact (after motion correction) increased steadily with
6.2. QUANTIFYING MOTION ARTEFACTS WITH ICA

Figure 6.12: A bar-chart which shows the absolute value of the total standard deviation (std) in the simulated data. The four bars represent std explained with ICs found from the four simulations (average motion levels of: 0 deg/TR; 0.046 deg/TR; 0.138 deg/TR; and 0.23 deg/TR). Dark blue bars represent the std associated with all the non-motion-related IC components, while light blue, green, orange and brown bars represent the std of individual motion-related ICs.

<table>
<thead>
<tr>
<th>IC1</th>
<th>IC2</th>
<th>IC3</th>
<th>IC4</th>
<th>IC5</th>
<th>IC6</th>
<th>IC7</th>
<th>IC8</th>
<th>IC9</th>
<th>IC10</th>
<th>IC11</th>
<th>IC12</th>
<th>IC13</th>
<th>IC14</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.8</td>
<td>0.83</td>
<td>0.79</td>
<td>0.72</td>
<td>0.77</td>
<td>0.64</td>
<td>0.62</td>
<td>0.47</td>
<td>0.75</td>
<td>0.68</td>
<td>0.21</td>
<td>0.29</td>
<td>0.58</td>
</tr>
<tr>
<td>0.2</td>
<td>0.7</td>
<td>0.77</td>
<td>0.60</td>
<td>0.45</td>
<td>0.60</td>
<td>0.49</td>
<td>0.58</td>
<td>0.46</td>
<td>0.61</td>
<td>0.45</td>
<td>0.2</td>
<td>0.37</td>
<td>0.33</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5</td>
<td>0.69</td>
<td>0.39</td>
<td>0.39</td>
<td>0.51</td>
<td>0.26</td>
<td>0.5</td>
<td>0.24</td>
<td>0.52</td>
<td>0.33</td>
<td>0.11</td>
<td>0.38</td>
<td>0.34</td>
</tr>
</tbody>
</table>

Figure 6.13: The table is showing the correlation coefficients between thresholded IC spatial maps for the non-motion data set and the 3 motion-affected data sets. Only IC1 shows poor correlation since it had similar spatial and temporal characteristics as the motion components and was not effectively separated from the motion components. Note that a correlation value of 0.2 or higher has less than a 0.01 probability of occurring by chance (based on Fishers r-to-z transformation).

Increasing motion levels. Similarly the number of estimated ICA components was increasing with increasing motion.

Furthermore, the non-motion-related ICs are consistently estimated in each case, as shown by the correlations between them in Table 6.13. Rows 2, 3, 4 of this table show correlation coefficients of the non-motion data with the three motion-corrupted datasets (ascending motion levels).
6.2. QUANTIFYING MOTION ARTEFACTS WITH ICA

6.2.4 Conclusions

Degraded quality of FMR images due to motion is often observed in real experiments [31] (e.g. in patients) and is hard to correct, making FMRI less reliable than often required in clinical practice. We have shown quantitatively how ICA can be used to identify motion-related components, how the variation (std) of these components is related to amount of motion, and how the temporal characteristics of some of the motion components is not linearly related to the motion parameter changes. Furthermore, we have shown that the remaining, non-motion-related components are reliably estimated even when large amounts of motion are present. The complicated relationship between motion-artefact and motion parameters can only be explored because of the use the simulator. This complicated relationship also demonstrates that simple motion-artefact-reduction (e.g. using motion parameters as regresses) is not sufficient to correct for such artefacts.
6.3 EVALUATING STIMULUS-CORRELATED MOTION ARTIFACTS

6.3 Evaluating Stimulus-Correlated Motion Artifacts

In this section we show an application of the use of the simulator in modelling the impact of stimulus-correlated motion in FMRI data. The results represented in this section are published in Drobnjak et al. [22].

6.3.1 Introduction

Degraded quality of FMR images due to stimulus-correlated motion (SCM) is often observed in real experiments [31], [34], specifically in patients, and is hard to correct with existing motion correction methods. Rigid body motion correction algorithms, such as MCFLIRT, that were described in Section 2 cannot remove fully the SCM effects.

Most methods use rigid-body motion correction algorithms as the first step. During the statistical analysis of the FMRI data they apply a second step that uses the estimated motion parameters from the first step (or from some other method such as ICA) to remove the motion-induced fluctuation from the voxels by regressing the estimated motion parameters from the FMRI signal [26]. This method has been proved to be very good with the motion in general, but struggles to cope with the case when the motion is correlated with the experimental stimulus because when regressing the estimated motion parameter lots of the signal gets regressed as well.

In this section the simulator is applied in order to qualitatively show the effects that stimulus-correlated motion has on the FMRI data. We present simulations of two FMRI datasets, each corrupted with a different kind of head motion: motion correlated with the experimental paradigm; and motion uncorrelated with the
experimental paradigm. We show the different effects of these motion sequences on activation maps and how rigid-body motion correction algorithms that do not account for artefacts cannot recover the true activation maps.

### 6.3.2 Methods

In order to simulate BOLD activation (Figure 6.14), realistic $T_2^*$ values that contain activation, resting state networks and neuronal noise were used. They were derived by fitting the equation $S = S_0 \exp(-TE/T_2^*)$ to an experimentally acquired FMRI data set (from Montreal Neurological Institute, McGill University), where $S$ is the measured signal intensity, $TE$ is the echo time and $S_0$ is the baseline intensity (determined from the average value of $S$ under baseline conditions). The experiment that the data was generated from involved auditory naming with the paradigm consisting of six blocks ‘on’, and six blocks ‘off’. During ‘on’ blocks, subjects hear a description of an object and had to think of the name of the object.

Simulations were generated with and without the presence of the $B_0$ field distortions (top two versus bottom two rows in both (a) and (b) of the Figure 6.14); and with (i) no motion, (ii) motion that is uncorrelated with the experiment paradigm and (iii) motion that is correlated with the experiment paradigm (described with values under each image). The two motion sequences shown in the figure were used to generate seven different motion sequences each, by scaling the rotation angles by 0, 1/50, 1/20, 1/10, 1/5, 1/2, and 1. For each of these sequences the amount of motion experienced by the object voxel that undergoes the maximum motion (expressed in percentages of the image voxel) is evaluated and is shown under each activation image.

Thermal noise was included with Contrast to Noise Ratio $CNR \sim 1$. Analysis
was carried out using FEAT (FMRI Expert Analysis Tool) Version 5.42, part of FSL (FMRI’s Software Library, www.fmrib.ox.ac.uk/fsl). The following pre-statistics processing was applied; spatial smoothing using a Gaussian kernel of FWHM 5mm; mean-based intensity normalisation of all volumes by the same factor; highpass temporal filtering. Time-series statistical analysis was carried out using FILM with local autocorrelation correction [71]. Z (Gaussianised T/F) statistic images were thresholded using clusters determined by \( Z > 2.3 \) and a (corrected) cluster significance threshold of \( P = 0.01 \) determined by the Gaussian Random Field theory [72].

Motion correction was done separately, prior to this processing, by applying the inverse of the known input rotation and translation to each image, using trilinear interpolation. Note that this corresponds to the ‘ideal’ case of rigid-body parameter estimation, and is not affected by the image artefacts.

### 6.3.3 Results and discussion

The BOLD changes modelled in these images includes changes of interest but also those of no interest (“physiological noise”). In Figure 6.14 (a), it can be seen as expected, uncorrelated motion has acted as extra random noise and, depending on the strength of motion, the activations shrank or were completely lost. On the other hand, motion that is correlated with the experimental paradigm created many false activations which can be seen in the images in Figure 6.14 (b). As the motion became larger, the maximum rotation ranging from \( 0.08^\circ \) to \( 4^\circ \), the size and the number of false activations increased as well. This is particularly emphasised in the cases with the \( B_0 \) distortion in which case false activations are also seen at the edges of the areas affected by the \( B_0 \) artefact.

As the motion amplitude increases the number of erroneous activation clusters
Figure 6.14: Images showing the effects of motion of different magnitudes on BOLD changes. On the right hand side of both (a) and (b) are figures describing the experimental paradigms together with the varying motion parameter (rotation about z-axis). Simulations generated with the average amount of motion (ranging between -0.7 and 0.7 degrees) experienced maximum 30% and 35% (in image voxels) movements, in the case (a) and (b) respectively. Figure (a) shows the resulting activation images when the motion is uncorrelated with the experimental paradigm in which case it acts like added noise and therefore, as expected, with increase in motion some of the activations are lost inspite of the motion correction. Figure (b) shows the resulting activation images when the motion is correlated with the experimental paradigm. As expected, some false activations are seen on the edges of the images and also on the edges of the $B_0$ affected areas (in the case of simulations with $B_0$ perturbation - bottom two rows).
6.3. EVALUATING STIMULUS-CORRELATED MOTION ARTIFACTS

increases as well. Some of the erroneous activation clusters look quite realistic and like real activations (e.g. the activation site just next to the $B_0$ distorted area). Such real looking activations in a real experiment, which are solely due to motion, often get confused with the real activations. In order to avoid any danger, researchers tend to model out all of the images that look suspicious. This causes a loss of valuable data, and is costly regarding the time and the resources.

A follow up application after the one described in this section would be to use the simulator in order to predict these erroneous activation sites so that they could be differentiated from the real activation sites. The simulator would use the motion parameters estimated by the use of an existing motion correction algorithm and, together with all of the subject-, scanner-, and experiment-specific properties, predict where the false activations (if any) would be in the images. If the approach proves to be successful and commonly used in practice, it could possibly save quite a lot of data that would otherwise be lost and would help in increasing the chance of the accurate and precise results in the FMRI. It would specifically be helpful in individual clinical FMRI experiments when there is only one chance to do the experiment.

6.3.4 Conclusions

These simulations demonstrated that stimulus uncorrelated motion reduces the number and the size of activations but does not induce erroneous activations. This behavior was the same with or without the presence of the $B_0$ magnetic field inhomogeneities. It was shown that in the case of average motion the number of the activation sites and their size was not significantly different from the true one.

On the other hand, it was concluded that stimulus-correlated motion both
6.3. EVALUATING STIMULUS-CORRELATED MOTION ARTIFACTS

reduces the number and the size of true activations and creates erroneous activations. The number, position and extent of the erroneous activations was dependent on the presence of the magnetic field inhomogeneities. It was also shown that in the case of average motion the number of the activation sites and their size was significantly different from the true one. This conclusion confirms the results that stimulus-correlated motion affects the data significantly more than the stimulus-uncorrelated motion.

Possible extensions of this application were also suggested. These would serve as a tool to predict the erroneous activations, and eliminate them from the real data. The proposed method would possibly be able to save large amounts of motion corrupted data which would otherwise be lost.
6.4 Eddy Current Effects

This work was performed in collaboration with Rita G. Nunes and Stuart Clare from FMRIB's Physics Group. It was published in [56].

A study where the eddy current effects were simulated for the standard and optimised sequences is presented. The ability to correct for the effect of eddy currents using an affine registration method is analysed.

6.4.1 Introduction

In order to be able to infer on fibre structure in the brain, a set of diffusion-weighted (DW) images need to be collected under a set of different diffusion gradient directions. The presence of the eddy current fields induced by the diffusion gradients during the readout window can lead to significant geometric distortions in echo planar images (EPI) [39]. Misalignment of the DW images will, in turn, lead to errors in the estimates of the diffusivities and fibre orientations [39].

In order to minimise the effect of eddy currents at source, modifications to the standard single spin-echo sequence, shown in Figure 6.15 (a), have been suggested. One of the modifications is the doubly-refocused sequence, developed by Reese et al.[64], and is shown in Figure 6.15 (b). When gradients are switched on and off, they induce eddy current fields of opposite polarity. In the case when the time constant of the eddy current fields is much longer than the duration of the gradients, the opposite eddy current fields will cancel out. By applying diffusion gradients with a shorter duration than the ones in the standard sequence, the cancellation of eddy currents having lower time constants can be achieved, enabling the minimisation of the eddy currents at source [64].

Despite its advantages regarding the minimisation of eddy currents, the doubly-refocused sequence is still not often used. This is because it is generally assumed
that the geometric distortions observed with the single spin-echo DW images should be correctable using affine registration methods. This would be true provided that the eddy current effects could be considered both time invariant throughout the acquisition window and spatially invariant from slice to slice. This assumption, together with concerns regarding potential signal-to-noise loss due to the presence of an extra imperfect 180° pulse have prevented the wider implementation of the doubly-refocused sequence. The standard single spin-echo sequence still remains, therefore, the most commonly used.

While the presence of higher order eddy current terms has been experimentally demonstrated \cite{66}, the intrinsically low signal-to-noise of DW images, together with the presence of subject motion complicates their experimental quantification. In order to be able to isolate the aspects related only to eddy current fields and to remove all other confounds, the simulator was employed. The simulator has made it possible to quantitatively evaluate the extent to which affine registration
can correct for eddy current distortions.

6.4.2 Methods

Gradient-echo EPI images were generated. The matrix size was $128 \times 128$ corresponding to an in-plane resolution of $2 \times 2 \text{mm}^2$. A slice thickness of $3 \text{ mm}$ was chosen. The bandwidth was $100 \text{ kHz}$, the readout window $163 \text{ ms}$ and the echo time $90 \text{ ms}$. The effect of the eddy currents produced by each of the two sequences (single spin-echo and doubly-refocused) was simulated by superimposing a sum of exponentially decaying terms to the EPI readout gradients:

$$G_{\text{eddy}}(t) = \sum_i \pm \varepsilon G_{\text{diff}} \exp\left[-(t - t_i)/\tau\right]$$ (6.9)

each of these terms corresponds to the switching either on (-) or off (+) of each of the positive or negative diffusion gradients at time $t_i$. Two diffusion gradients were considered for the standard sequence each with a duration of $28 \text{ ms}$. For the doubly-refocused sequence four diffusion gradients with a duration of $14 \text{ ms}$ were considered. The amplitude of the diffusion gradients $G_{\text{diff}}$ was $21.3 \text{ mT/m}$ while the maximum amplitudes achieved by the read and phase-encode EPI gradients were $9.1 \text{ mT/m}$ and $4.0 \text{ mT/m}$ respectively.

A set of eddy currents were simulated with different relative amplitudes $\varepsilon$ and time constant $\tau$. Time constants of $5, 10, 30, 50, 70, 90$ and $100 \text{ ms}$ were considered. For the read out direction, eddy currents with amplitudes of $0.01, 0.04, 0.07, 0.1\%$ relative to those of the diffusion gradients were used. These amplitudes were rescaled for the phase-encode direction accounting for the lower strength of the phase-encode blips in comparison to the read out gradients.

The diffusion gradients could be applied either along the read ($xx$) axis, producing skews in the images or along the phase-encode direction ($yy$) producing scaling [39].
6.4. EDDY CURRENT EFFECTS

Although the difference in contrast displayed by both the T2-weighted image and the DW images should also have an influence in the efficiency of the registration step, no attempt was made to model anisotropic diffusion. To provide the distorted images with a realistic contrast, the mean attenuation produced in each tissue was instead considered. The k-space data corresponding to CSF, white and grey matter were separately generated and an appropriate attenuation factor applied to each of them before summation. These were based on a b-value of 1000 s mm$^{-2}$ and a simple exponential decay was assumed: $S = S_0 \exp[-b < D>]$. The mean diffusivities $< D >$ for each tissue were: 3.2$\times 10^{-3}$ mm$^2$/s (CSF), 0.8$\times 10^{-3}$ mm$^2$/s (grey matter), 0.7$\times 10^{-3}$ mm$^2$/s (white matter) [47].

The distorted attenuated images were then registered to the undistorted T2-weighted image using an affine registration method [36]. Five degrees of freedom were allowed for: translation along $xz$ and $yy$, rotations around the $zz$ axis, skews in $xy$ and scaling along $yy$. If eddy current distortions could be fully corrected by performing affine registration, no difference should exist between the registered images and the undistorted attenuated image. To evaluate the ability to correct for eddy-current-induced distortions, the sum of squares of the difference image between each registered image and the undistorted DW image was therefore calculated.

6.4.3 Results and discussion

The undistorted T2-weighted (a) and DW (b) simulated images are shown in Figure 6.16. Images generated for the single spin-echo sequence with eddy currents along the read (c) and phase encode direction (d) are also displayed.

The distortions are clearly visible when looking at the difference images (e)
6.4. EDDY CURRENT EFFECTS

Figure 6.16: Simulated images displaying eddy current effects. (a) T2-weighted image; (b) DW image; (c) DW image with an eddy current along the read direction (standard sequence, $\varepsilon = 0.01\%$, $\tau = 10$ ms); (d) DW image with an eddy current along the phase-encode direction (standard sequence, $\varepsilon = 0.01\%$, $\tau = 10$ ms); (e) Subtraction of image (b) from (c); (f) Subtraction of the image obtained after registering (c) to (a); (g) Subtraction of image (b) from (d); (h) Subtraction of the image obtained after registering (d) to (a).

and (g), where (b) was subtracted from each of the distorted images. Although registration did enable a reduction in these differences, residual distortions are still visible in both (f) and (h).

From the residuals plotted in Figure 6.17, it can be noted that the affine registration method does not seem to be able to cope effectively with eddy currents of large amplitudes and time constants much lower than the duration of the gradients. Unless a more effective correction method is used, minimising eddy currents at source should therefore be preferable. The simulation results also confirm previous experimental observations [63] as the residuals corresponding to the doubly-refocused sequence are significantly lower. There can be several causes for the residuals observed. One factor could be the presence of higher order eddy currents, not corrected using affine registration. On the other hand, the presence of either skews or scaling, even if affine, in the distorted images would
6.4. EDDY CURRENT EFFECTS

Figure 6.17: Sum of squares of the residuals for both sequences after affine registration, for eddy currents along the read (a) and the phase encode (b) directions. The sum of square of the residuals was obtained when comparing each registered image with the undistorted DW image. Each curve corresponds to a different time constant in ms as indicated in the legend. Note that the vertical scales are quite different in the left hand and the right hand figures.

distribute the overall signal over a wider number of pixels. This signal cannot be totally recovered by performing registration even if the images become perfectly re-aligned and should lead to errors when estimating diffusivities and/or fibre orientations. Another factor that can influence the quality of the registration is the presence of eddy-current-induced ghosting.

As stated in Nunes [56] “In order to be able to improve these simulations, in the future the gradient waveforms used could be altered so as to resemble more closely those used when acquiring diffusion images on the actual scanner. An-
other aspect which was not taken into account was that DW images are normally acquired using a spin-echo sequence. At the moment POSSUM does not simulate the effect of 180° pulses, but this feature is expected to be added in the future. A further step of the simulation which could be improved, is the way in which the effect of the eddy currents is added to the gradient waveforms. Instead of simply adding a sum of exponential decays, the actual gradient waveforms could be filtered so as to remove the high-frequency components. In this way the ramping up and down of the gradients could be made to be smoother as observed in practice.

Under these conditions the scanner simulator employed here could be extremely useful, as different aspects of the problem could be studied in isolation. In a first stage, this simulator could be used to consider the problem of correcting for high levels of distortion in the images without the additional difficulty of also having a low image SNR. Increased levels of noise could then be added so as to provide better tests for the correction methods developed. The problem could then be made to be even more complex, so as to better replicate the experimental conditions, by progressively introducing the effects of subject motion and field inhomogeneities."

6.4.4 Conclusions

With the use of the simulator it was demonstrated that using a standard affine registration method is not sufficient to correct for strong eddy current effects. By using the doubly-refocused sequence, the level of eddy currents is substantially reduced compared to the more widely used single spin-echo approach. The same results were achieved experimentally by Nunes [56]. Furthermore, it was also demonstrated that for the doubly-refocused sequence the registration step per-
forms equally well over a wide range of amplitudes and time constants. Possible additions to the simulator together with the novel areas of future work in which this simulator could be useful in this application have also been suggested.
6.5 Direct MR imaging of neuronal activity

The work in this section is done in collaboration with Dr Gaby Pell from the Brain Research Institute, Melbourne, Australia. Dr Pell is investigating the characteristics of an MRI technique that could potentially be used for direct detection of neuronal activity [61, 60]. The use of the simulator is likely to be of considerable use in this area. The main focus of the work presented in this section is to show the ways that the simulator could assist work in the area of direct imaging. In order to do that, two of the experiments carried out by Pell et al. and presented in their paper on “Optimisation of MR sensitivity to transient and weak currents in a conductor” [60] have been simulated. The experimental results obtained by Pell et al. are compared with the simulation.

6.5.1 Introduction

As described in Chapter 2, blood oxygenation level dependent (BOLD) fMRI has become one of the most widely used methods for the non-invasive imaging of neuronal activity in the human brain. However, this method does not measure the neuronal activity directly from the neurons themselves. It measures the haemodynamic changes that occur in the blood vessels that surround the neurons. The changes that occur in the blood vessels are coupled with the neuronal activity, but the exact nature of the coupling is complex and still not fully understood [49, 3]. The accepted view is that the haemodynamic changes are relatively sluggish and variable and therefore an imprecise measure of neuronal activity [5].

In addition, due to its dependence on the vasculature, the BOLD response is much slower than the actual neuronal response thus limiting the temporal resolution to seconds rather than milliseconds which are needed in order to image the transient activity of the neurons. Coarse temporal resolution also results
in the inability of FMRI to temporally resolve cascaded communication between sequentially activated brain regions. Furthermore, the spatial extent of the BOLD response is significantly larger than the presumed localised area of active neurons responsible for the change in the blood flow [60]. In the imaging voxels with large vessel effects this can result not only in the enlargement of the activation area but also in its slight shift (up to a centimeter) from the true region of activation [5].

Due to the above mentioned limitations of BOLD FMRI techniques, a need for new, more sophisticated methods to image brain activity has emerged. In the past few years, there has been significant research done in the field of direct neuronal imaging [12, 11, 73, 42, 43, 5, 60] which investigate new MRI techniques in directly detecting the localised magnetic effects arising from neuronal currents. These techniques rely on transient electrical activity perturbing the local magnetic field during a portion of the acquisition sequence and altering the phase and magnitude of the MR signal. The development of these techniques could potentially provide a neuronal mapping method with a direct access to the neuronal activity and with an excellent spatial (millimeters) and temporal (milliseconds) resolution.

In order for the neuronal current imaging to be used in practice, more research needs to be done regarding the sensitivity of the MR signal to the neuronal currents. In order to address this issue, Pell et al. investigate the sensitivity of MR magnitude signal change to: 1) the location of the transient magnetic field within the imaging voxel; 2) the current amplitude; 3) the timing of the current pulse within the imaging sequence; and 4) imaging parameters [60].

By using the simulator it is possible to analyse the sensitivity of MR magnitude signal change to any of the above mentioned factors. Physical experiments are time- and material- consuming and very often it is quite difficult to eliminate or minimise the artefacts that can appear during scanning. By using the simula-
tor as an aid when running the experiments it is possible to avoid those issues. This section demonstrates how the simulator was used to analyse the sensitivity of MR magnitude signal change to: 1) the 0.1mA current amplitude and the phase encode direction; and 2) the timing of current pulse within the imaging sequence.

Simulations are performed in such a way as to closely follow the experimental set up described in Pell et al. [60]. This was done in order to be able to compare whether the two independently acquired set of results (simulated results and the scanner acquired results) match well. In order to present the work fully, the relevant elements of the experimental set-up used for acquiring the scanner results by Pell et al. [60] are described throughout the section.

6.5.2 Methods

The phantom for the scanner-based experiments was constructed from a hollow glass sphere, filled with SF96/50 silicone oil. A carbon fibre conductor (diameter 0.015mm) was placed inside the glass sphere and a Perspex support structure was constructed to hold it in its place [60]. Passing current through the conductor is equivalent to the activity in a neuron (or a bundle of neurons aligned in parallel and oriented in the same way). The $T_1$ and $T_2$ relaxation times of the oil are 1420ms and 530ms respectively. These were measured by Pell et al.[60].

The phantom (object) used for the simulations was a simple sphere which is presented in the top row of Figure 6.19. The sphere had a very thin conductor going through its centre. This sphere represents an area around one of the two legs of the conductor in the phantom used in the experiments of Pell et al.. When the current goes through a conductor, a magnetic field is created around the conductor, perturbing the magnetic field of the scanner. The newly created
changes in the magnetic field are calculated using the Biot-Savart law. The perturbed field is presented in the bottom row of Figure 6.19. $T_1$ and $T_2$ values used in the simulator were the same as the ones estimated by Pell et al. for the silicone oil.

The experiments by Pell et al. were performed on a 3T system with gradient slew rate of 150T/m/s and a maximum gradient strength 40mT/m. These values were incorporated into the pulse sequence generator of the simulator. Pell et al. used a hardware- system to generate current in the conductor with accurate and precise control of the amplitude and duration. An EPI sequence was used in the experiments (Pell et al. also used other sequences in their paper) and it had the following parameters: $128 \times 128$ matrix, 6mm slice thickness, 90° excitation flip
angle, sampling time of 8μs, imaging voxel size of 1.56 × 1.56 × 6mm, $TE = 85\text{ms}$, $TR = 1\text{s}$, slice select gradient is $G_y$ (coronal slices).

**Sensitivity to the current and the phase encode direction**

This experiment was designed to determine the change in the signal magnitude and the spatial characteristics of this change when a current pulse of amplitude 0.1mA is applied. The current pulse duration was 20ms, and it was placed immediately after the 90° RF pulse. In addition to the image when the current pulse is on, an image when the current pulse is off was also generated. The image representing the signal change is obtained by subtraction of these two images.
Sensitivity to the timing of the current pulse within the imaging sequence

This experiment was designed to investigate the change in the signal magnitude when a current pulse is “walked through” an EPI pulse sequence. The current pulse amplitude was 7.8mA, its duration was 5ms, and its position within the pulse sequence was changed in each of the 260 volumes that were acquired in the following fashion: during the acquisition of the first volume, the current was applied immediately after the 90° RF pulse; during the acquisition of the second volume no current pulse was applied; during the acquisition of the third volume, the current pulse was placed 1ms later than that during the acquisition of the first volume; during the acquisition of the fourth volume, again, no current pulse was applied; during the acquisition of the fifth volume, the current pulse was placed 1ms later than that during the acquisition of the third volume; and so on. The timing of the onset of the current pulse was increased in 1ms increments for every alternate volume acquisition. A time-series is then extracted from the region of maximal signal change within the image. The experiment is discussed in more detail in Pell et al. [60].

6.5.3 Results and discussion

Sensitivity to the current and the phase encode direction

Figure 6.20 shows the subtraction of the image simulated with the current pulse on, and the image simulated with the current pulse off. The subtraction represents the signal change that occurred due to the current pulse. The image is shown with extreme colour contrast, mainly to point out the overall areas of positive (white) and negative (black) signal change. These show the “splitting” of the signal along the x-axis (in the case of phase encode in the x-direction) and along
the x-z diagonal (in the case of phase encode in the z-direction).

This kind of signal splitting has been noticed also by Pell et al. [60] and the comparison of their result to the result obtained by the simulator is shown in Figure 6.21. Note that the results of Pell et al. are statistical results showing all of the statistically significant voxels, which is not the case with the results of the simulator which were generated free of any noise or other artefacts.

In order to get more insight into the magnitude of the signal change and its spatial distribution, percentages of the change were calculated for each voxel individually (for one coronal slice) and shown in Figure 6.22. Results are shown for the case when the phase encode is in the x-direction. The voxel at the position
Figure 6.21: A comparison between Pell et al. results (the left hand side) and the simulator results (the right hand side) are shown. The middle column is a 24 x 24 voxels patch around the left carbon fibre position extracted from the images on the left hand side. Phase encode in the x-direction (top row): Signal is seen to “split” along the phase encoding direction (x). Phase encode in the z-direction (bottom row): Signal is seen to “split” along the x-z diagonal.

Figure 6.22: Signal change image in a 9 by 9 patch surrounding the central image voxel (the one containing the conductor) for phase encode in the x-direction. Red values represent intensity change (as percentages) for each of the image voxels independently.
(65,65) is the voxel through which contain the wire with the current goes. This voxel experienced the maximum of the signal change (10.8%). The signal change rapidly decreases as we are moving away from the wire with maximum signal change of 4.9% in the next layer and 1.9% change two layers of voxels away from the central voxel. This is equivalent to 1.56mm and 3.12mm respectively (as the in-plane voxel size is 1.56 × 1.56mm). As expected, the decay happens in a symmetrical way which is due to the symmetry in the magnetic field inhomogeneities induced by the wire.

**Sensitivity to the timing of the current pulse within the imaging sequence**

Figure 6.23 shows the results of the “walk-through” experiment which was described in more detail in the Methods section. Sensitivity of the signal to the timing of the current pulse within the imaging sequence is demonstrated. The figure shows the signal in percentages: 100% is the value when no change occurred, 80% is the value when a 20% signal change occurred etc. The large signal change between any two consecutive points represents the interleaving of the on and off states of the current pulse. In the figure, signal values (bottom row) and the pulse sequence events for the acquisition of one slice (top row) are aligned in such way as to represent that each signal value in the bottom row corresponds to a position in the top row which is where the current pulse was centered. For example, the 120th volume is acquired during the 120th second of the total imaging time (as each TR is 1s) and during its acquisition the onset of the current pulse was at 60ms in the pulse sequence.

The results obtained by the simulator showed very similar behaviour to the experimental results. The signal rapidly increases after the application of the RF pulse. It then maintains a steady level until the pulse enters the read-out
6.5. DIRECT MR IMAGING OF NEURONAL ACTIVITY

Figure 6.23: Results of the “walk-through” experiment. On the left hand side is the experimental result acquired experimentally by Pell et al. (Figure courtesy of Pell et al.). On the right hand side of the figure is the simulator result. The top row shows the time-progression of the pulse sequence (one for each of the experiments). The red line indicates the center of the k-space. The bottom row shows the normalised signal (in %). Signal values at 100% correspond to volumes acquired when no current pulse was on. The rest of the signal values correspond to the volumes when the current pulse was on. The position of the value corresponds to the time the current pulse was on within the pulse sequence (top row). The key observation in both of the left and right hand plots is that the contrast is maximal when the current pulse is timed to overlap the readout of the center of k-space.

gradient. The signal change subsequently rises in a linear fashion. The largest signal change occurs when the current pulse coincides with the acquisition of the central line of the k-space. The signal change subsequently decreases and reaches very small levels before the end of the read-out window.

The magnitude of the signal change is, however, different in the two results. This can be possibly attributed to the fact that the Pell et al. result was generated using a partial k-space reconstruction whereas the simulator results were acquired with the full k-space. It was observed by Pell et al. [60] that the results for the full k-space showed lesser change in signal magnitude.
6.5.4 Conclusions

Two experiments done by Pell et al. [60] (investigating the sensitivity of the signal to the current pulse of 0.1mA amplitude for different phase encode directions, and sensitivity of the signal to the timing of the current pulse) were replicated by the use of the simulator. The results obtained were a very close match with the results obtained by Pell et al. The work on applying the simulator in direct neuronal imaging is still on-going, but the preliminary experiments that were described in this section are very encouraging.

The application of the simulator can be very useful as an aid in doing experiments in direct current imaging. One of the main reason is that it is easier, less costly and potentially more accurate (as some of the scanner artefacts can be avoided) to do simulations than real experiments which involve physical phantoms or human subjects. The simulator can be particularly useful when investigating the effect that more than one neuronal current would have on the signal (say randomly oriented in the object). This is one of the potential future experiments and directions for the use of the simulator in this field.

6.6 Summary

In this chapter five different applications of the simulator were investigated.

Firstly, the simulator was applied in testing a motion correction algorithm - MCFLIRT. It was shown that the overall performance of MCFLIRT algorithm was good. Different default options to the ones that are currently used in the software are suggested.

Secondly, the simulator was applied to investigate the performance of ICA as a tool for quantifying motion-related artefact. It was shown that ICA could make reasonable estimation of motion-related independent components, however more
investigation is needed in order to make more general conclusions.

Thirdly, the simulator was applied in evaluating stimulus-correlated motion artefacts in FMRI data. The results demonstrated that stimulus correlated motion affects the data significantly more than stimulus uncorrelated motion. The results also demonstrated that the areas of distortion due to the $B_0$ inhomogeneities were particularly badly influenced in SCM-affected images.

The fourth application of the simulator presented in this thesis was in simulating eddy currents artefacts. With the use of the simulator it was demonstrated that using a standard affine registration method is not sufficient to correct for strong eddy current effects. It was also shown that by using the doubly-refocused pulse sequence, the level of eddy currents is substantially reduced compared to the more widely used single spin-echo approach.

The fifth application of the simulator presented in this thesis is in direct neuronal current imaging. The influence of a neuronal current on the image generation was simulated. Results were a very close match with the results obtained experimentally by Fell et.al.[60].