Wavelet-based methods for the analysis of fMRI time series

Wink, Alle Meije

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2004

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Chapter 5

Extracting the Haemodynamic Response Function from fMRI Time Series using Fourier-Wavelet Regularised Deconvolution with Orthogonal Spline Wavelets

Abstract

We describe a method to extract the haemodynamic response function (HRF) from functional magnetic resonance imaging (fMRI) time series based on Fourier-wavelet regularised deconvolution (ForWaRD). The algorithm presented here is an extension of an earlier ForWaRD-based extraction method. We introduce the computation of shift-invariant discrete wavelet transforms (SI-DWT) in the frequency domain, and apply ForWaRD using orthogonal spline wavelets. The extraction of subject-specific HRFs is demonstrated, as well as the use of these HRFs in a subsequent brain activation analysis. Temporal responses are modelled by using the extracted HRF time signals in a new model for fMRI time signals. The resulting activation maps show the effectiveness of the proposed method.

5.1 Introduction

Functional magnetic resonance imaging (fMRI) is a versatile method for functional neuroimaging. Regional brain activation induces changes in blood oxygenation, generating a blood oxygenation level dependent (BOLD) contrast in MR images (Ogawa et al. 1990). An important tool for fMRI analysis is statistical hypothesis testing, where the fMRI signal is predicted using the stimulus pattern and a response model. Statistical parametric mapping (Friston et al. 1995c) uses a model of the noise (Gaussian), and hypothesis tests are based on the parameters of the model. This chapter presents a method to extract the haemodynamic response function (HRF) from fMRI data. The method is based on Fourier-wavelet regularised deconvolution, ForWaRD (Neelamani et al. 2004), using orthogonal spline wavelets.
ForWaRD combines the advantages of deconvolution in the frequency domain and regularisation in the combined frequency and wavelet domains, see Fig. 5.1. The output is given as a time series of image volumes, containing the HRF at each voxel location. Compared to other HRF extraction methods (Glover 1999, Miezin et al. 2000), the method requires only a few assumptions: the stimulus-response relation should be linear and time invariant (LTI), and the signal should be distinguishable from noise in the Fourier and/or wavelet representation. The use of ForWaRD to extract the HRF is treated in more detail in chapter 4.

The extraction method given in chapter 4 is extended by using a novel frequency-domain implementation of the shift-invariant discrete wavelet transform (SI-DWT). It may be efficient to compute the wavelet transform in the frequency domain (Westenberg and Roerdink 2000), as is the case for orthogonal spline wavelets, which do not have compact support.

The remainder of this chapter is organised as follows. Section 5.2 introduces some fMRI and wavelet terminology. Then in section 5.3 we treat the computation of the SI-DWT in the frequency domain, and section 5.4 presents the spline wavelets we used. In section 5.5, fMRI experiments are analysed using the proposed extraction method. In section 5.6 we present our conclusions.

5.2 SPM, Wavelets, and ForWaRD

5.2.1 SPM

Statistical parametric mapping (SPM) is the standard fMRI analysis tool. It assumes the temporal noise to be independent, identically distributed and Gaussian. SPM consists of the following steps: (i) estimate the parameters of the noise, (ii) compute a statistic at every voxel location, (iii) threshold the statistic values using the noise parameters and a multiple testing correction method. Assuming a linear, time invariant (LTI) stimulus-response relation, an fMRI data set of \( T \) time samples in \( N \) voxels is modelled as:

\[
Y_{[T \times N]} = X_{[T \times M]} \beta_{[M \times N]} + e_{[T \times N]},
\]

(5.1)

\( Y \) represents the fMRI data, \( X \) is the design matrix of \( M \) explanatory variables (modelled effects), \( \beta \) contains the weights of each of these variables, and \( e \) contains the residuals, i.e., the unmodelled part of the signals. Each column of \( X \) contains the modelled response to the stimuli of one type. An LTI response to one type of stimulus is given by a convolution of the time pattern of the stimuli and the appropriate HRF. A good HRF model is critical to the success of the estimation based on (5.1), because inappropriate modelling will lead to a non-Gaussian distribution of the values in \( e \).
5.2.2 Wavelets

A wavelet transform describes a signal $c^0$ as a sum of basis functions. Given a scaling function $\phi$ and a basic wavelet $\psi$, $c^0$ is split into an approximation (smooth) part $c^1$ and a detail part $d^1$, which are weighted sums of shifted dilates of $\phi$ and $\psi$, respectively. Multi-level transforms divide subsequent $c^j$ into $c^{j+1}$ and $d^{j+1}$, $j = 0, 1, \ldots, J, J \in \mathbb{N}$. The inverse transform recursively uses $c^j$ and $d^j$ to reconstruct $c^{j-1}$. An efficient algorithm for computing these wavelet transformations is the fast wavelet transform, FWT (Mallat 1989), which subsamples $c^j$ and $d^j$ at each level. It requires $O(N)$ computations for a signal of size $N$. However, the FWT is not shift-invariant, and is not useful for deconvolution. Using polyphase decomposition (subsample for all possible shifts) instead of subsampling the undisplaced filtered signal yields a shift-invariant discrete wavelet transform (SI-DWT), and its inverse (SI-IDWT). The output of a level $L$ transform has size $(L+1) \times N$, and the complexity is $O(N \log_2[N])$ (Mallat 1991).

5.2.3 ForWaRD

Using the LTI model of section 5.2.1, an HRF can be extracted from an fMRI time series by deconvolving the measured time signals with the stimulus pattern. In the frequency domain, deconvolution can in principle be done via pointwise division (Fourier inversion). However, noise is amplified at frequencies where the signal is small, introducing instability: small changes in the inputs induce large changes in the output. Regularisation suppresses the destabilising effects. If the destabilising factor is the noise, regularisation is tantamount to denoising.

A common regularisation technique for frequency domain deconvolution is shrinking the frequency coefficients after inversion. The signal of interest is usually smooth (low-frequency) and noise is usually erratic (high-frequency). Two familiar shrinkage
methods are Wiener shrinkage and Tikhonov shrinkage (Neelamani et al. 2004). Non-smooth parts of signals (such as steep edges) are not efficiently represented in the frequency domain, because they contain much high-frequency energy. As a result, noise at those frequencies is not shrunk. Using more shrinkage to remove noise introduces artifacts, such as ringing.

The ForWaRD method applies Fourier inversion, Fourier shrinkage, and wavelet-domain Wiener shrinkage. Wiener shrinkage reduces the magnitude of wavelet coefficients at indices where the true signal is weak, and preserves those coefficients where the true signal is strong. The true signal is unknown, so ForWaRD uses two wavelet transforms of a signal: one transform with basis functions $(\phi_1, \psi_1)$ to estimate the true signal by thresholding detail coefficients, and another transform with basis functions $(\phi_2, \psi_2)$, whose detail coefficients $d_{2j}, j = 1 \ldots J$ are shrunk. ForWaRD uses the SI-DWT to ensure shift-invariance. ForWaRD can be used to extract an HRF $h$ from a noisy signal $g$, assuming $g$ to be the convolution of the stimulus pattern $f$ with $h$, plus noise $e$. This is denoted as: $g = f * h + e$. The algorithm is summarised in Algorithm 5.1.

5.3 Computing the SI-DWT in the frequency domain

ForWaRD requires the SI-DWT, which was implemented in the time domain (Mallat 1991). Spline wavelets (Unser and Blu 2000) can be computed most efficiently in the frequency domain. An FWT subamples at each level (taking samples $\{0, 2, \ldots, N-2\}$), whereas an SI-DWT first does a polyphase decomposition (for $Q$ phases: take samples $\{0, Q, \ldots, N-Q\}$, $\{1, Q+1, \ldots, N-Q+1\}$, $\ldots$, $\{Q-1, 2Q-1, \ldots, N-1\}$), filters

![Figure 5.1](image_url)
each phase separately, and then does a monophase reconstruction, *i.e.*, it combines all phase signals back into one signal. It is possible to do this in the frequency domain. As described by Rioul and Duhamel (1992) and Vetterli and Herley (1992), subsampling by a factor \( Q \) (using only phase 0) in the frequency domain separates the frequencies into \( Q \) consecutive blocks and computes an average block. Computing phases \( \{1, \ldots, Q-1\} \) requires (i) a shift to each block to sample coefficients from the right block \( \{0, \ldots, Q-1\} \), (ii) a shift inside the block to sample the right coefficient \( \{0, \ldots, N/Q-1\} \), and (iii) averaging the \( Q \) blocks. The monophase reconstruction in the frequency domain distributes the phases again across the frequency blocks by, for each phase, applying the appropriate shifts to each block and concatenating the blocks. The signals created for each phase are added together to reconstruct one signal in the frequency domain. Note that a shift of \( k \) places in an \( N \)-point signal is a multiplication by \( e^{2\pi ik/N} \) in the frequency domain.

While filtering is cheap in the frequency domain, polyphase decompositions are quite costly, mainly because of the many multiplications required for shifting. We have optimised the SI-DWT in the frequency domain towards processing a matrix of many 1D signals, such as a time series of large images. The imaginary exponentials for the shifts, which are equal for all voxel locations, are precomputed once for all signals, and access to the phases and frequency blocks is accelerated by changing the dimensionality of the (initially 1D) signals according to the number of phases.

### 5.3.1 Efficient computation of the frequency-domain SI-DWT

The frequency-domain implementations of the SI-DWT and the SI-IDWT, respectively, are given in Algorithm 5.2. The input parameters of the transform are the Fourier transforms of (i) a signal \( s \), and (ii) the wavelet filters \( h \) and \( g \). Analogous to the FWT in the frequency domain, the SI-DWT produces the frequency representations of the approximation and the detail channels (the SI-DWT in the time domain can be obtained by applying the IFFT to these signals). The frequency-domain SI-IDWT transforms the approximation and the detail channels back into the Fourier transform of \( s \).

Computing the complex exponentials required for the polyphase and monophase transforms, respectively, is a costly step. The application described here uses a large number of 1D transforms with the same parameters, and computation time is reduced by computing those exponentials (step 1 of Algorithm 5.2) only once and reusing them. The exponentials required for the polyphase decomposition are given in Eq. (B.14) of the appendix. To further speed things up, the product of every required combination of exponentials in (B.14) is stored in a lookup table. The exponentials required for the monophase reconstruction, given in (B.15), are also stored in a lookup table.

According to (B.14), for each phase \( q \) (whose signal has length \( N/Q \)), each frequency block \( \ell \), \( \ell = 0 \ldots Q-1 \) is multiplied by \( e^{(2\pi i(\ell q))/Q} \) (to shift to the right frequency block for sampling) times \( e^{(2\pi ikq)/N} \) (to select the right phase) and then the blocks are averaged. If the complex exponentials are available, the polyphase decomposition amounts to adding a number of pointwise vector products. Using the combined shifts from the
lookup table, the number of multiplications is considerably reduced. The frequency
domain polyphase decomposition requires more computations than the time-domain
version (the signal needs to be processed \(Q\) times), but for long filters this is worth the
effort: the convolution step requires much less time than in the time domain (see also
section 5.3.2).

More efficiency is obtained by changing the dimensionality of each 1D signal before
the decomposition. A length-\(N\) signal is split into \(Q\) frequency blocks of length \(N/Q\).
After the polyphase decomposition these blocks contain the \(Q\) phases of length \(N/Q\).
The monophase reconstruction in the frequency domain also uses the polyphase de-
composition and the monophase reconstruction. The gain in speed here is also obtained
by precomputing the shift exponentials of (B.15), and by changing the dimensionality
of the signal.

The forward transform performs the polyphase decomposition, filters the phases,
and then performs the monophase reconstruction for each level of decomposition, and
then doubles the number of phases \(Q\). The filters are subsampled by a factor \(2\) in the
frequency domain. This corresponds to extracting the \(N/2\)-point Fourier transform of a
signal from the \(N\)-point Fourier transform (Westenberg and Roerdink 2000).

The inverse transform starts with the number of phases at the highest level of de-
composition, and \(Q\) is divided by two after each reconstruction step. The filters cannot
be subsampled progressively because the reconstruction starts at the coarsest level. Fast
access to the subsampled filters is achieved by changing the dimensionality of the filters.
To subsample by a factor \(2^j\), the \(N\)-point filter is split into \(N/2^j\) blocks of length \(2^j\),
and the first element of each block is used. After filtering, the 1D signal is restored.

5.3.2 Computation times: spline wavelets

The algorithm for computing the SI-DWT in the frequency domain is computationally
intensive. For compact support filters, the time-domain computation is more efficient.
When the wavelet basis functions either have exponential decay instead of compact
support or are defined directly in the frequency domain, the transform proposed in
Algorithm 5.2 above may be more efficient than the time-domain computation. The
time-domain versions of the transforms use convolution, of which the computation time
increases with the signal length as well as with the filter length. The length of wavelet
filters that do not have compact support increases with the signal length \(N\), so the com-
plexity of the convolutions is \(O(N^2)\). In the frequency domain the complexity is \(O(N)\),
because time-domain convolution is pointwise multiplication in the frequency domain.
We compared the computation times of the time-domain version and the frequency-
domain version of the SI-DWT, by computing the three-level SI-DWT of signals vary-
ing in length, using the symmetric orthogonal cubic spline wavelet basis (Mallat 1989).
For each length, 100 signals were transformed and reconstructed. Figure 5.2 shows
the computation times for signals of varying lengths. The graphs show that the frequency-
domain implementation is faster for signals of more than 64 points. Therefore, the
frequency-domain SI-DWT is preferred for long filters (like orthogonal spline wavelets), and the rest of this chapter uses the frequency-domain implementation.

![Graph showing computation times](image)

**Figure 5.2.** Computation times of the time-domain SI-DWT and SI-IDWT and the frequency-domain SI-DWT and SI-IDWT, respectively, with symmetric orthogonal cubic spline wavelet basis functions. Time domain: ×: SI-DWT, ○: SI-IDWT. Frequency domain: □: SI-DWT, *: SI-IDWT.

### 5.4 ForWaRD using spline wavelets

The efficient frequency-domain implementation of the SI-DWT inside ForWaRD facilitates the use of spline wavelets in the HRF extraction routine. We use orthonormal splines to preserve the signals’ energy during the transform. An implementation of fractional splines (Unser and Blu 2000) was used to generate the wavelet basis functions. Examples of spline wavelet basis functions are shown in Fig. 5.3. Our new version of ForWaRD with frequency-domain SI-DWT and orthonormal spline wavelets was used in the extraction program. In this chapter, we used causal splines with degree $\alpha = 4$ for $(\phi_1, \psi_1)$ and anticausal splines with degree $\alpha = 3$ for $(\phi_2, \psi_2)$.

### 5.5 Event-Related fMRI Experiments

The HRF extraction routine was used in the analysis of two event-related fMRI experiments of one subject, measured on different days. The subject had to make a fist at presentation of a visual stimulus, and then immediately relax.

Stimuli were presented on a white screen inside the scanner. A white disc was shown as the default, a red disc was the cue to make a fist. One experiment was performed with a fixed interstimulus interval (ISI) and one with a randomised ISI. Realignment,
normalisation, and statistical analysis were done with the SPM program (Friston et al. 1995c). Denoising was done with a wavelet-based technique (Wink and Roerdink 2004). We computed HRFs for the whole brain and in a region of interest, respectively, which were then used in covariance analyses to test for activation.

5.5.1 Fixed-ISI Experiment

The fixed-ISI data set consisted of 156 volumes of $64 \times 64 \times 46$ voxels with size $3.5 \times 3.5 \times 3.5$ mm$^3$. Cues were given every 24 s (8 scans $\times$ 3 s) starting at scan 2. HRFs were extracted by our method, and also by selective averaging (Dale and Buckner 1997), a simple and robust extraction method when the ISI is long. A first statistical analysis was done to detect activation synchronous to the stimuli. We used a design matrix with a set of 6 Fourier basis functions (3 sines, 3 cosines), modulated by a Hanning window, in the time interval of 8 scans after each stimulus, so as not to impose shape assumptions on the HRF. An SPM$\{F\}$ resulting from an $F$-test was computed, using false discovery rate (FDR) control (Genovese et al. 2002) with $q = 0.05$ for multiple hypothesis testing. With both ForWaRD (using 128 of the 156 scans) and selective averaging, we computed a whole-volume HRF and a regional HRF in a $7 \times 7 \times 7$-voxel region with high activity (see the region indicated by a ‘$<$’ in Fig. 5.4a). The post-stimulus volumes were multiplied by the $F$-values in the map, after thresholding with an FDR-parameter $q = 0.0001$, and averaged over the volume/region. Figures 5.5a-b show the HRFs. Selective averaging shows better results than ForWaRD, which remains below baseline in the ISI. This may be because the real HRF does not return to baseline within the measured interval, so in the LTI model the response decreases at every next stimulus. This results in an HRF with a lower baseline.
5.5.2 Random-ISI Experiment

Stimulus times for this experiment were random and the length of the random-ISI data set was 256 scans, the other parameters were unchanged. Post-stimulus image volumes were produced by ForWaRD. Due to response overlap, neither selective averaging nor the Fourier basis set could not be used. The design matrix $X$ was made by convolving the stimulus signal with the canonical HRF from the SPM’99 program (Friston et al. 1995c) and its time and dilation derivatives. HRFs were made from the SPM{$F$} (see Fig. 5.4b) and the post-stimulus volumes, see Fig. 5.5c. The regional HRF corresponds most to the previously extracted HRFs. Both HRFs return to baseline within the post-stimulus interval.

![Figure 5.4](image)

**Figure 5.4.** SPM{$F$} of the fixed-ISI experiment (a), SPM{$F$} of the random-ISI experiment (b).

![Figure 5.5](image)

**Figure 5.5.** HRFs extracted from the fixed-ISI data set by selective averaging (a) and by ForWaRD (b), and from the random-ISI time series by ForWaRD (c). $\times$: whole-volume, $\circ$, region-specific.
5.5.3 Using the extracted HRFs in activation tests

A covariance test was done on the random-ISI data using a model based on the fixed-ISI HRF, and vice versa. Extracted HRFs cannot not be used for covariance tests on the same data set: a model must be specified a priori, and inferences cannot be made from models that are determined by the data. We modelled the HRFs by fitting one function

\[
f_{H,D,P,L}(t) = \begin{cases} 
  H \sin\left(\frac{t-L}{P}\right) e^{-\frac{t-L}{D}}, & \text{if } t > L \\
  0, & \text{otherwise}
\end{cases}
\] (5.2)

to the HRFs extracted from the fixed-ISI data, and two such functions were used to the HRFs from the random-ISI data.

Function (5.2) models a damped oscillator with parameters \( H \) (eight), \( L \) (ag), \( P \) (period), and \( D \) (ilation), which is a plausible model for a delayed response such as the BOLD signal. With the HRFs from the random-ISI experiment we use two such functions: one to model the peak and one to model the undershoot. The fixed-ISI HRFs did not have enough points to model the undershoot. The fitted functions were used to build the design matrices. The maps in Fig. 5.7 resulting from a t-test show very similar shapes as those in Fig. 5.4, but here the detected activations are stronger. This indicates that the model used here captures all variance captured by those methods. The difference between this analysis and the previous is that only one basis function is used here, enabling a covariance test with stronger responses.

Table 4.2 of the previous chapter is extended in this chapter by adding the maximum variance ratio values found in the tests of this section. Recall that a high variance ratio indicates that much of the variance in the signal is explained by the model, and that the residual noise in the GLM (see Eq. (5.1)) is small. Table 5.1 shows that the HRFs

![Figure 5.6](image-url)
Extracting the HRF using ForWaRD with orthogonal spline wavelets

Figure 5.7. SPM(T)’s of the activation found by using the modelled HRFs: (a) fixed-ISI data, random-ISI whole-volume HRF, (b) fixed-ISI data, random-ISI regional HRF, (c) random-ISI data, fixed-ISI whole-volume HRF, (d) random-ISI data, fixed-ISI regional HRF.

Table 5.1. A comparison of the maximum variance ratio values found in this Chapter and those found in Chapter.

<table>
<thead>
<tr>
<th></th>
<th>ForWaRD (splines)</th>
<th>original ForWaRD</th>
<th>selective averaging</th>
<th>HRF_{spm}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>volume</td>
<td>region</td>
<td>volume</td>
<td>region</td>
</tr>
<tr>
<td>fixed-ISI</td>
<td>120</td>
<td>163</td>
<td>113</td>
<td>162</td>
</tr>
<tr>
<td>random-ISI</td>
<td>103</td>
<td>101</td>
<td>103</td>
<td>102</td>
</tr>
</tbody>
</table>
extracted from the random-ISI data set by ForWaRD with orthogonal spline wavelets yield better results than the HRFs previously extracted. Especially the undershoot of the whole-volume HRF is better detected with ForWaRD and orthogonal spline wavelets (see Fig. 5.5). The HRFs extracted from the fixed-ISI data set do not show measurable improvements. As indicated in Section 5.5.1 a possible explanation for this is that the ISI is too short to successfully capture the whole HRF.

The generally small differences between the results of both versions of ForWaRD may be explained by the fact that ForWaRD is quite robust to the choice of wavelet filters, as observed in the tests in Chapter 4. Another possible regularising factor is that not the coefficients themselves are used, but the functions $HRF_{par}$, and that different HRF time signals yield very similar fits.

5.6 Conclusion

We have presented an HRF extraction method for fMRI time series based on ForWaRD. The output of our algorithm is a post-stimulus time series, representing the HRF in every voxel. The existing ForWaRD method was extended by introducing a novel, frequency-domain, implementation of the SI-DWT. Computation time was reduced by precomputing and reusing the exponentials required for the polyphase decomposition and the monophase reconstruction. The efficiency of the algorithm was further increased by changing the dimensionality of the signals and filters according to the number of phases. Timings show that for signals longer than 64 points, the speed gain of the frequency-domain transform is considerable. This enabled us to efficiently use orthogonal spline wavelets. We also presented a model for the HRF that can be used in combination with the extracted coefficients to predict event-related fMRI responses. The modelled HRF appears to capture the same amount of variance in one basis function as the tested traditional methods, which require multiple basis functions.
**Extracting the HRF using ForWaRD with orthogonal spline wavelets**

Given: signal $s$ of length $N$, wavelet filters $h$ and $g$

$s \xrightarrow{\text{FFT}} S; \ h \xrightarrow{\text{FFT}} H; \ g \xrightarrow{\text{FFT}} G$

Forward transform (SI-DWT):

1. Compute shifts, the set of complex exponentials for the shifts
2. $C^0 := S; \ Q := 1$
3. for $j = 1$ to $J$
   4. shifts, $C^{j-1}$ polyphase(Q) → $\{C^{Q,q}\}_{q=0}^{Q-1}$ (see Eq. (B.14))
   5. for $q = 0$ to $Q - 1$
      6. $D^{Q,q} := C^{Q,q}G$ (pointwise multiplication)
      7. $C^{Q,q} := C^{Q,q}H$ (pointwise multiplication)
   8. shifts, $\{D^{Q,q}\}_{q=0}^{Q-1}$ monophase(Q) → $D^j$ (see Eq. (B.15))
   9. $H := \downarrow 2^j H; \ G := \downarrow 2^j G$
10. $Q := 2Q$
11. end for

Result: $C^j$ and $\{D^j\}_{j=1}^J$.

Inverse transform (SI-IDWT):

1. Compute shifts, the set of complex exponentials for the shifts
2. for $j = J$ downto 1
3. $Q := 2^{j-1}$
4. shifts, $C^j$ polyphase(Q) → $\{C^{Q,q}\}_{q=0}^{Q-1}$ (see Eq. (B.14))
   shifts, $D^j$ polyphase(Q) → $\{D^{Q,q}\}_{q=0}^{Q-1}$ (see Eq. (B.14))
5. $H := \downarrow 2^{j-1} H_s$
   $G := \downarrow 2^{j-1} G_s$
6. for $q = 0$ to $Q - 1$
   7. $C^{Q,q} := (C^{Q,q}H_s + D^{Q,q}G_s) / 2$
   8. end for
9. shifts, $\{C^{Q,q}\}_{q=0}^{Q-1}$ monophase(Q) → $C^{j-1}$ (see Eq. (B.15))
10. $Q := Q/2$
11. end for

$C^0 \xrightarrow{1\text{D-IFFT}} s$

† These shifts are the same as those used in the forward transform.

**Algorithm 5.2:** Frequency-domain SI-DWT and SI-IDWT in pseudo-code.