Sparse signal representations using the tunable Q-factor wavelet transform

Ivan W. Selesnick

Polytechnic Institute of New York University, Brooklyn, NY 11201, USA

ABSTRACT

The tunable Q-factor wavelet transform (TQWT) is a fully-discrete wavelet transform for which the Q-factor, Q, of the underlying wavelet and the asymptotic redundancy (over-sampling rate), r, of the transform are easily and independently specified. In particular, the specified parameters Q and r can be real-valued. Therefore, by tuning Q, the oscillatory behavior of the wavelet can be chosen to match the oscillatory behavior of the signal of interest, so as to enhance the sparsity of a sparse signal representation. The TQWT is well suited to fast algorithms for sparsity-based inverse problems because it is a Parseval frame, easily invertible, and can be efficiently implemented using radix-2 FFTs. The TQWT can also be used as an easily-invertible discrete approximation of the continuous wavelet transform.

Keywords: wavelet transform, constant Q transform, sparse signal representation

1. INTRODUCTION

For the sparse wavelet representation of a signal, the Q-factor of the wavelet transform should be chosen so as to match the signal's oscillatory behavior. This paper reviews a new wavelet transform, the 'tunable Q-factor wavelet transform' (TQWT) presented in Ref. 17, for which the Q-factor is continuously tunable. Therefore, the wavelet can be chosen according to the oscillatory behavior of the signal, so as to enhance the sparsity of a sparse representation. The TQWT is well suited for iterative algorithms for sparse representation because it is a fully-discrete self-inverting transform which can be efficiently implemented using radix-2 FFTs.

This paper will illustrate sparse signal representations using the TQWT obtained by ℓ_1-norm minimization.

2. TUNABLE Q-FACTOR WAVELET TRANSFORM (TQWT)

Constant-Q signal analysis and synthesis[13] has long been a subject of interest, in part motivated by the approximately constant-Q property of biological hearing in mammals. Constant-Q signal analysis refers to the analysis of a signal using a set of band-pass filters, all with the same Q-factor. The Q-factor of a band-pass filter is defined as the ratio of its center frequency to its bandwidth,

$$Q = \frac{f_c}{\text{BW}}.$$

Therefore, the frequency responses of a constant-Q set of filters appear to have equal width when illustrated on a log-frequency axis. The temporal response of a band-pass filter with a high Q-factor exhibits more sustained oscillatory behavior that that of a filter with a low Q-factor. The Q-factor in a sense counts the number of oscillations in the temporal response, with lower amplitude oscillations being down-weighted.[16]

Nowadays, constant-Q transforms are commonly known as wavelet transforms. In practice, the term 'constant-Q transform' appears to be more often used in audio applications[4,15] and other applications for which a high Q-factor is needed. On the other hand, most wavelet transforms used in applications are based on a low Q-factor; for example, the dyadic Daubechies wavelets (and other dyadic wavelets) have low Q-factors ($Q \approx 1$). However, several (non-dyadic) wavelet and related transforms have been developed for use over a range of Q-factors.[2,3,9,10,12,14]
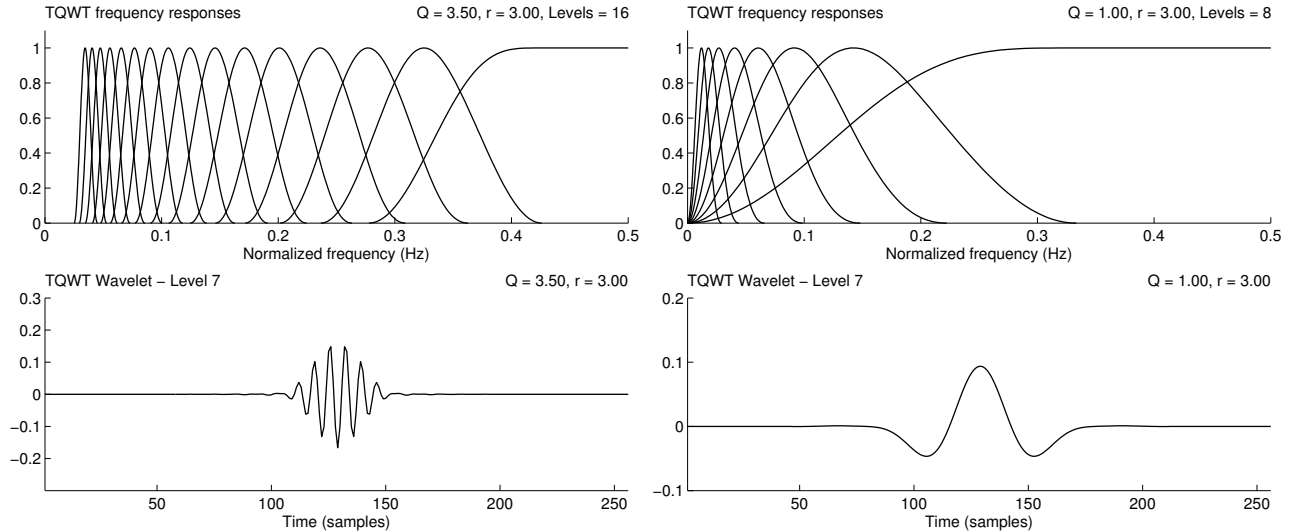

Figure 1. TQWT frequency responses and wavelets for two Q-factors: $Q = 3.5$ (left) and $Q = 1.0$ (right).

Wavelet transforms (constant-Q transforms) with low Q-factors are useful for the efficient representation of piecewise smooth signals (such as a scan line of an image) whereas wavelet transforms with high Q-factors are useful for the efficient representation of oscillatory signals (such as audio, or EEG rhythms).

The 'tunable Q-factor wavelet transform' (TQWT) is a recently developed wavelet transform designed so that the Q-factor is easily and continuously adjustable. The motivation for the TQWT is the efficient (sparse) representation of signals that exhibit some degree of oscillatory behavior. It is thought that, by adjusting the Q-factor of the wavelet transform so as to match the oscillatory behavior of the signal under analysis, a more efficient (sparse) signal representation can be obtained. The enhanced sparsity should in turn improve the performance of sparsity-based signal processing algorithms for applications such as denoising, deconvolution, classification, signal separation, etc.

Intended specifically for *discrete*-time signal processing, the TQWT is a fully-discrete transform. The derivation of the TQWT in Ref. 17 follows closely that of the classic dyadic wavelet transform filter banks, with the exception that in place of the dyadic down-sampling, the TQWT employs continuous-valued rate-changers for discrete-time signals.* Therefore, the derivation of the TQWT depends on a definition of continuous-valued rate-changers; a particular definition is given in Ref. 17.

The TQWT introduced in Ref. 17 is essentially parameterized by two parameters: the Q-factor Q and the asymptotic over-sampling rate r. The parameter r is the redundancy factor of the wavelet transform in the limit as the number of levels (scales) goes to infinity. The parameter r can also be understood as a measure of how much spectral overlap exists between adjacent band-pass filters. Keeping Q constant, but increasing r, results in the band-pass filters constituting the TQWT maintaining the same Q-factor, but having increasing overlapping in the frequency domain. Hence, by using a large value for r, one can use the TQWT to approximate the continuous wavelet transform (the TQWT coefficients sample the time-frequency plane more finely with respect to frequency). A high degree of spectral overlap between adjacent band-pass filters may also be desirable for modeling the cochlear of mammals, which can be roughly thought of as a continuum of band-pass filters.

Figure 1 illustrates the TQWT for two different sets of TQWT parameters. On the left, the parameters are set to $Q = 3.5$ and $r = 3.0$; on the right, the parameters are set to $Q = 1.0$ and $r = 3.0$. The figure illustrates the frequency responses of the band-pass filters constituting the TQWT. It can be seen in the figure that for $Q = 1.0$ the band-pass filters are quite wide, like the classic dyadic wavelet transform, and relatively few levels

---

*By a continuously-valued rate changer we mean one which produces output samples at a rate of αf_s where f_s denotes the number of input samples per unit time, where α is a positive real scalar.

(scales, filters) are needed to traverse the spectral content of the signal of interest. In contrast, for $Q = 3.5$ each band-pass filter is narrower and therefore more levels (filters) are needed so as to cover the spectrum of a signal. As Fig. 1 illustrates, the Q-factor is easily observed in the frequency responses of the TQWT. Figure 1 also illustrates the 'wavelet' of the TQWT in each case. The wavelets are simply the temporal responses (impulse responses) of the the band-pass filters. In the figure, the wavelet (temporal response) is illustrated at level 7. The role of the Q-factor is also easily observed in the shape of the wavelet. For low Q-factor, the wavelet has few sign changes and consists of few oscillatory cycles (basically, a single cycle for $Q = 1.0$ in Fig. 1). For higher Q-factor, the wavelet has more sign changes and consists of more oscillatory cycles, which is more suitable for the efficient representation and processing of oscillatory signals.

We will denote the TQWT linear operator (matrix) by Φ so that $\mathbf{w} = \Phi\mathbf{x}$ represents the computation of the wavelet coefficients $\mathbf{w}$ of a signal $\mathbf{x}$. The vector of wavelet coefficients $\mathbf{w}$ can be considered as a concatenation of the subbands of the wavelet transform into a single vector $\mathbf{w}$. The TQWT is an over-complete transform, so $\mathbf{w}$ is longer than $\mathbf{x}$, and Φ is a rectangular matrix (taller than it is wide).

The TQWT has the property that it can be inverted using Φ^*, that is

$$\Phi^*\Phi = \mathbf{I}. \tag{1}$$

Therefore, the TQWT is a so-called 'Pareval frame' or 'tight-frame', meaning that it satisfies an energy preservation property,

$$\|\Phi\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2.$$

That is, the energy of the signal is preserved by the TQWT.

We note, however, that the TQWT is not a 'uniform tight frame', meaning that the analysis functions (wavelets) do not all have the same ℓ_2 norm. That is, the rows of Φ do not all have the same ℓ_2 norm. However, within a single subband, the norms are the same; the norm of the wavelet depends on the subband index, not on its temporal position.

The TQWT can be implemented using radix-2 FFTs. A Matlab toolbox and a set of C programs are available for implementing the TQWT and for sparse signal representation/approximation using the TQWT.

3. SPARSE REPRESENTATIONS

Sparse signal processing exploits sparse representations of signals for applications such as denoising, deconvolution, signal separation, classification, etc. Therefore, transforms for the sparse representation of signals is a key ingredient for sparse signal processing. In some sense, the utilized transform should be chosen so as to maximize the sparsity of the signal(s) of interest. For signals that have some degree of oscillation, the transform should be chosen to match the oscillatory behavior. For example, when using the short-time Fourier transform (STFT), the window length should be chosen appropriately. When using the TQWT, the Q-factor should be chosen appropriately. In this section we illustrate by example the role of the Q-factor in using the TQWT for sparse signal representation.

Figure 2 illustrates two test signals that we will use in the following. Test signal 1 is a somewhat oscillatory signal comprised of three pulses, each pulse being a windowed sinusoid; the three pulses exhibit different frequencies. Test signal 2 is a less oscillatory signal consisting of biphasic pulses. It can be said that test signal 2 is a 'low Q-factor' signal whereas test signal 1 is a 'high Q-factor' signal (relative to signal 2).

3.1 Example 1: Test signal 1

When the TQWT is applied to test signal 1, we obtain the TQWT coefficients (subbands) illustrated in Fig. 3. As in Fig. 1, the parameters are set to $Q = 3.5$ and $r = 3.0$ (left) and to $Q = 1.0$ and $r = 3.0$ (right). The distribution of the energy of the wavelet coefficients is also illustrated in Fig. 3. Several observations can be made regarding Fig. 3. Note that for the high Q-factor TQWT (left) sixteen levels of the TQWT are computed, whereas for the low Q-factor TQWT (right) eight levels of the TQWT are computed. This is because, as noted above, a high Q-factor wavelet transform requires more levels to cover the same frequency range as a low Q-factor transform. Note that the last subband (subbands 17 and 9 on left and right) are the low-pass subbands. These
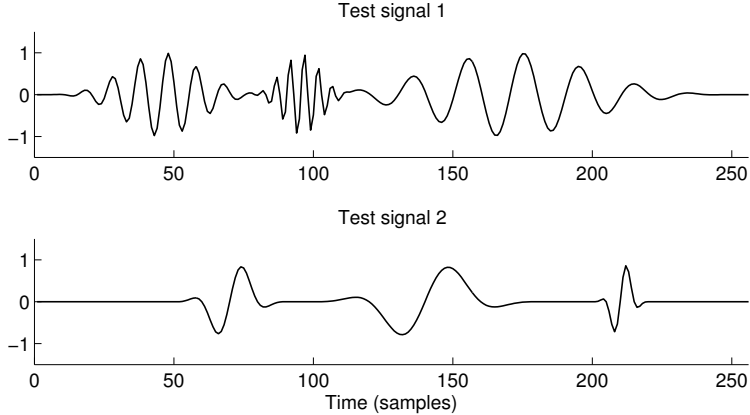
Figure 2. Test signals.

subbands are essentially zero, due to the band-pass behavior of the test signals. Note also that the subbands of the high Q-factor TQWT (left) are sampled at a lower rate than those of the low Q-factor TQWT (right).

Of particular relevance, is the comparative sparsity of the low and high Q-factor TQWT representations. It is illustrated in Fig. 3, that for test signal 1 (a 'high Q-factor' signal), the high Q-factor TQWT gives a sparser representation than the low Q-factor TQWT. That is, *sparsity can be enhanced by matching the Q-factor of the wavelet transform to the oscillatory behavior of the signal under analysis.* The enhanced sparsity (of Fig. 3 right vs left) is apparent in that each subband of the high Q-factor TQWT has fewer significant coefficients in comparison to corresponding subbands of the low Q-factor TQWT.

The TQWT is over-complete, therefore, the wavelet coefficients are not unique — other sets of wavelet coefficients can be obtained that also represent a given signal exactly. We can exploit the non-uniqueness of the wavelet coefficients to obtain a sparser set of coefficients. For example, we can obtain a sparser set of wavelet coefficients by via constrained ℓ_1 norm minimization (basis pursuit). The ℓ_1 norm of a vector $\mathbf{x}$ is defined as

$$\|\mathbf{x}\|_1 = \sum_n |\mathbf{x}(n)|.$$

3.2 Basis pursuit

Given a signal of interest $\mathbf{x}$, a sparse set of wavelet coefficients $\mathbf{w}$ can be found via ℓ_1 norm minimization. The problem can be formulated as a constrained minimization problem:

$$\operatorname*{argmin}_{\mathbf{w}} \quad \|\boldsymbol{\lambda} \odot \mathbf{w}\|_1 \tag{2a}$$

$$\text{such that} \quad \Phi^* \mathbf{w} = \mathbf{x} \tag{2b}$$

This approach is known as *basis pursuit*[6] (BP). This problem makes sense only when the system of equations (2b) are underdetermined, which is the case whenever Φ represents an over-complete transform.

The notation $\boldsymbol{\lambda} \odot \mathbf{w}$ in (2a) denotes element-wise multiplication of the equal-size vectors $\boldsymbol{\lambda}$ and $\mathbf{w}$. When all elements of vector $\boldsymbol{\lambda}$ are the same value λ, then (2) can be written as

$$\operatorname*{argmin}_{\mathbf{w}} \quad \lambda \|\mathbf{w}\|_1 \tag{3a}$$

$$\text{such that} \quad \Phi^* \mathbf{w} = \mathbf{x} \tag{3b}$$

which is the more common form of BP. We use the form (2a) (a weighted ℓ_1 norm) because the rows of Φ do not all have the same norm (the norm of the wavelet varies from subband to subband). More generally, it is useful to allow non-uniform weighting of $\mathbf{w}$ for its generality.
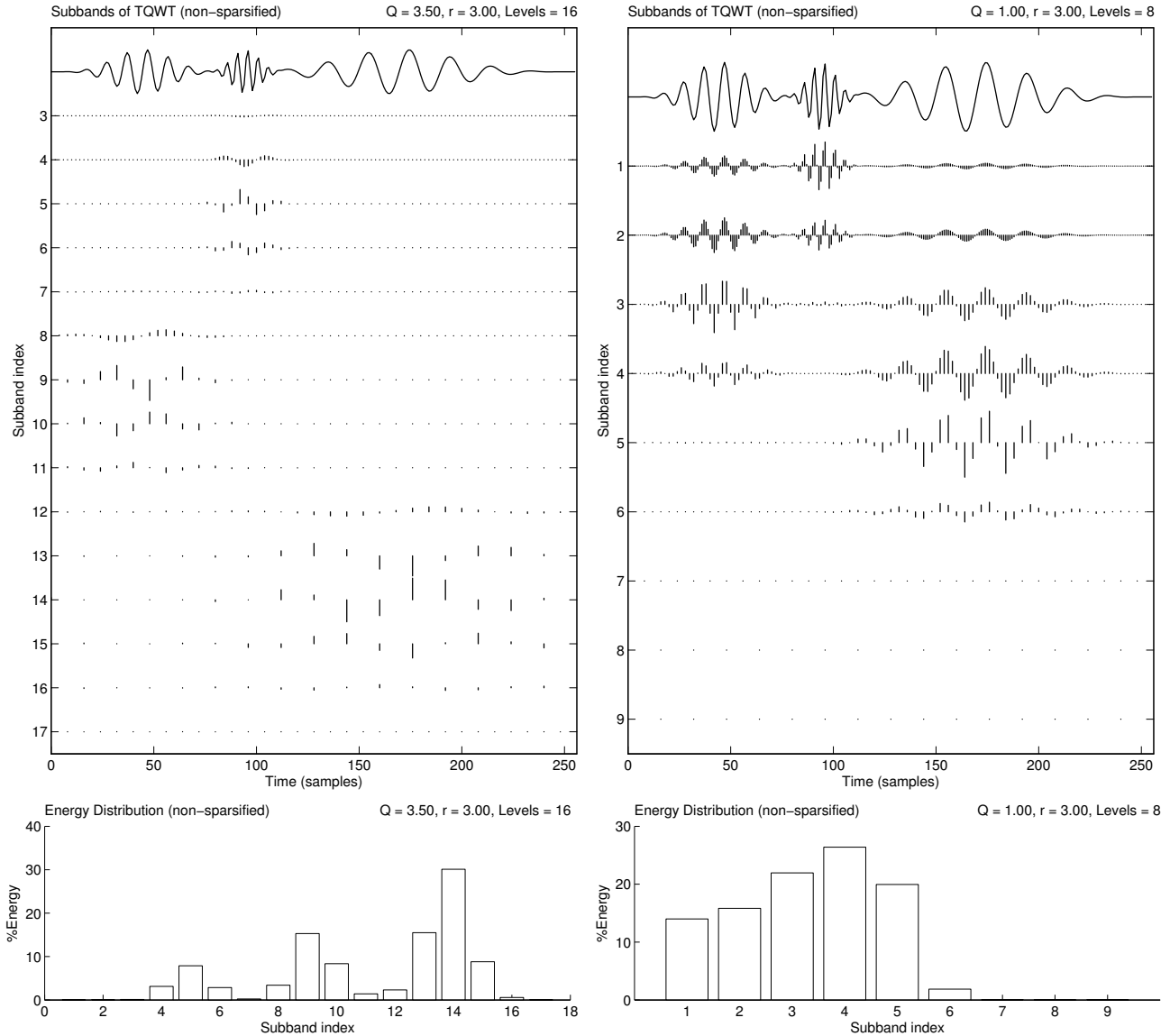
Figure 3. The coefficients (subbands) obtained by applying the TQWT to test signal 1. The TQWT parameters are set to $Q = 3.5$, $r = 3.0$ (left) and to $Q = 1.0$, $r = 3.0$ (right).

To take into account the scale-dependent norm of the wavelet, we set $\boldsymbol{\lambda}$ in (2a) as

$$\boldsymbol{\lambda}(k) = \|\psi(k)\|_2,$$

where $\psi(k)$ is the 'wavelet' corresponding to wavelet coefficient $\mathbf{w}(k)$. By the wavelet $\psi(k)$, we simply mean the k-th row of matrix Φ. (Note, the matrix Φ is never explicitly computed nor stored in memory. Instead the fast TQWT is used.) Hence, in subbands for which the wavelets have smaller norm, each wavelet coefficient is weighted less, accordingly.

The BP problem can not be solved in closed form; instead an iterative numerical algorithm is necessary. Appendix B gives an algorithm for solving the BP problem. The algorithm, derived using augmented Lagrangian and variable splitting, is a variant of SALSA.[1]
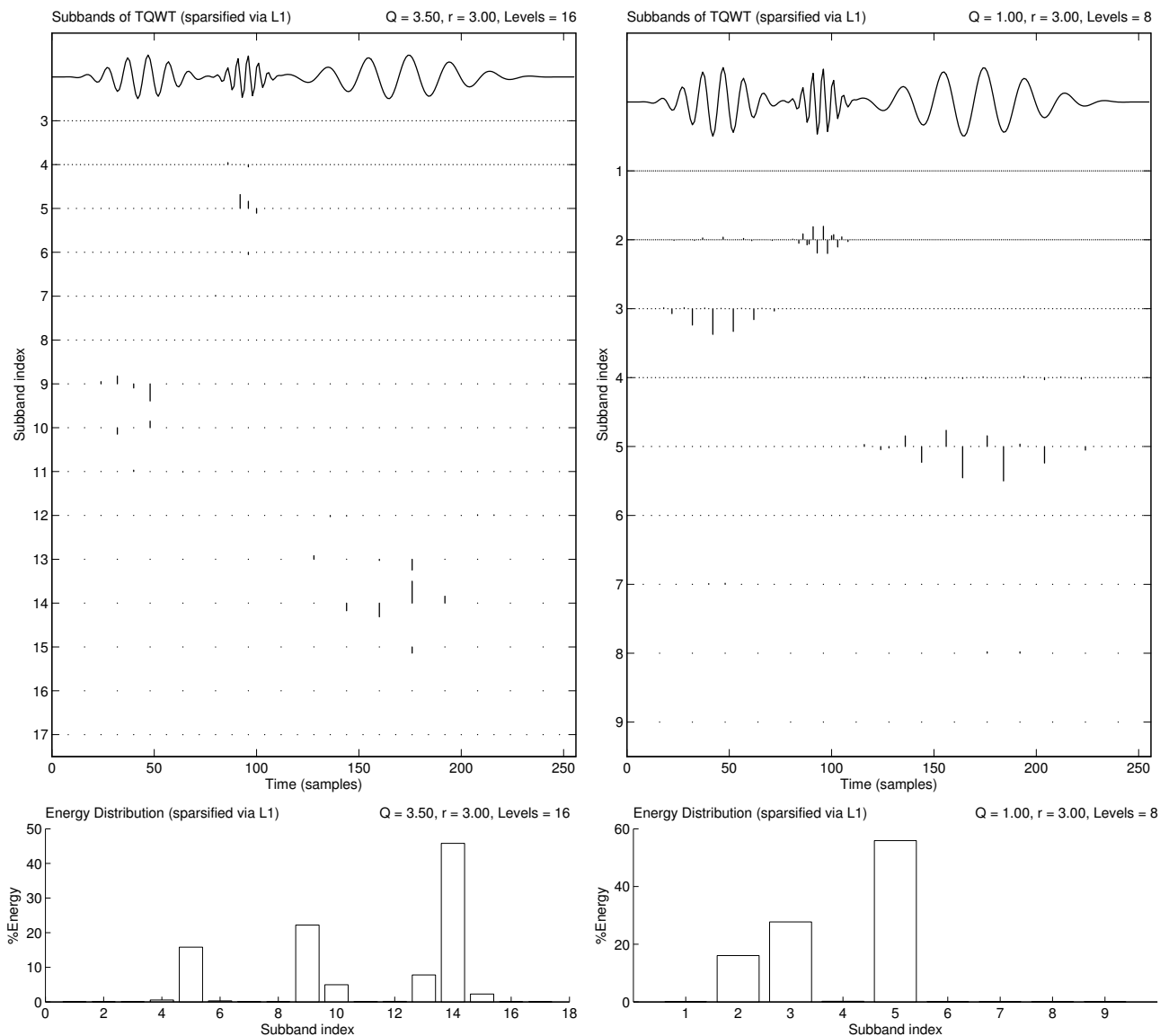
Figure 4. Coefficients (subbands) obtained via ℓ_1 norm minimization (basis pursuit) for the sparse representation of test signal 1. In comparison with Fig. 3, the TQWT coefficients are much sparser.

3.3 Example 1 continued

To obtain TQWT coefficients that are sparser than those illustrated in Fig. 3, we can use the basis pursuit approach — we can solve the constrained ℓ_1 norm minimization problem (2) using the iterative SALSA algorithm (14) in Appendix B.

The sparse wavelet coefficients (subbands) obtained via BP are illustrated in Fig. 4. Due to the constraint (2b), when the inverse TQWT (Φ^*) is applied to these coefficients, the signal $\mathbf{x}$ is obtained exactly. Comparing Fig. 4 with Fig. 3, it is clear that the TQWT coefficients obtained by BP (ℓ_1 minimization) are substantially more sparse than the coefficients obtained by simply applying the transform Φ to the signal $\mathbf{x}$.

Also, note that the energy in Fig. 4 is more concentrated into fewer subbands compared to Fig. 3. This is made clear in the bar graphs in the lower panel of the figures.
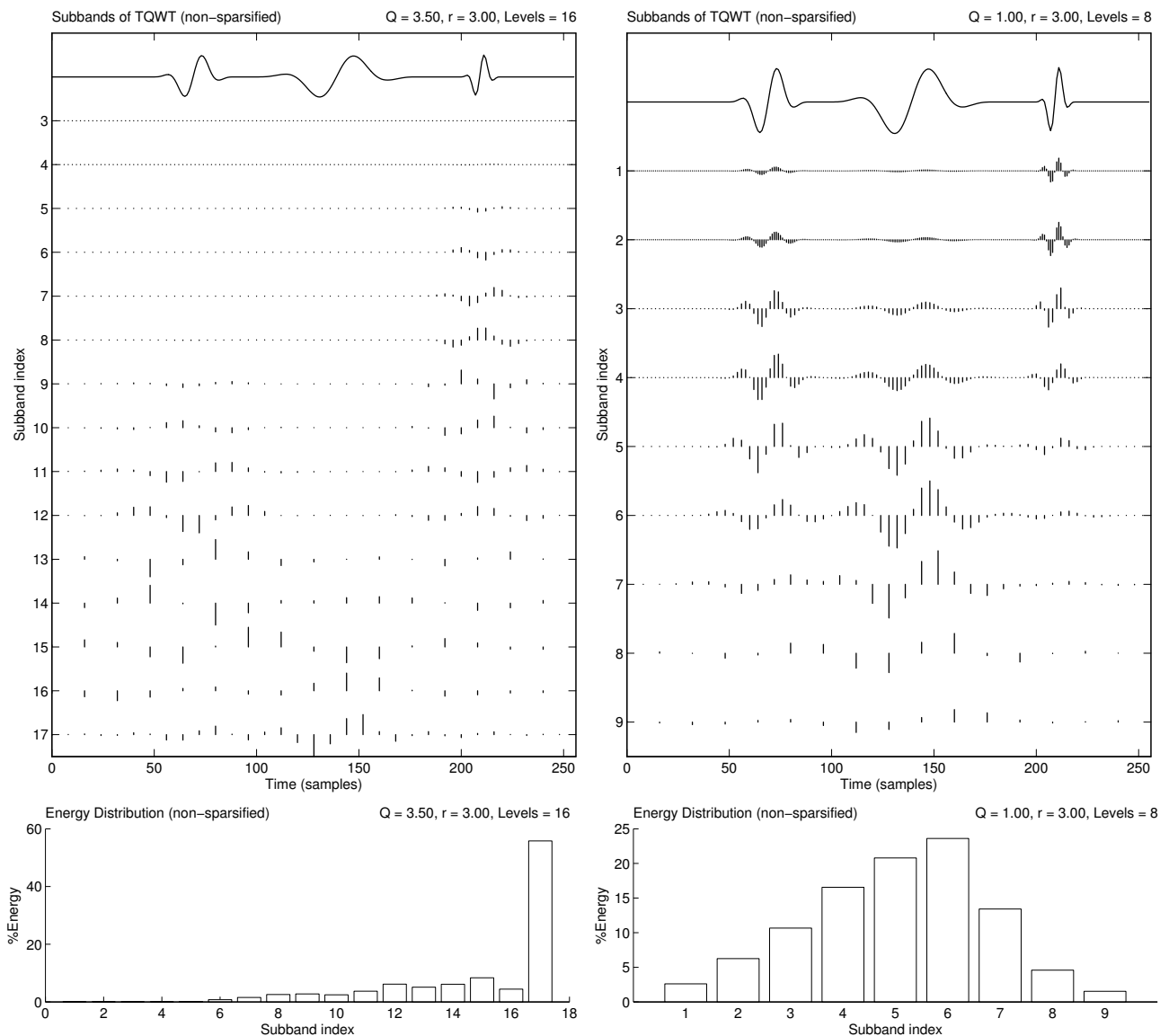
Figure 5. The coefficients (subbands) obtained by applying the TQWT to test signal 2. The TQWT parameters are set to $Q = 3.5$, $r = 3.0$ (left) and to $Q = 1.0$, $r = 3.0$ (right).

3.4 Example 2: Test signal 2

When the TQWT is applied to test signal 2, we obtain the TQWT coefficients illustrated in Fig. 5. As in Example 1, the parameters are set to $Q = 3.5$ and $r = 3.0$ (left) and to $Q = 1.0$ and $r = 3.0$ (right). It can be seen in Fig. 5 that neither representation is particularly sparse. This is partly due to the TQWT being an over-sampled transform. However, it can be observed that the number of scales over which the energy is spread is different between the two transforms. Consider the third of the the three pulses constituting test signal 2. For the high Q-factor TQWT, Fig. 5 (left), the energy in the wavelet domain due to this pulse is spread over about 10 subbands (subbands 5–15). On the other hand, for the low Q-factor TQWT, Fig. 5 (right), the energy due to this pulse is spread over about 5 subbands (subbands 1–5).

To obtain sparse TQWT coefficients for test signal 2, we use the basis pursuit approach (constrained ℓ_1 norm minimization). The wavelet coefficients obtained via BP are illustrated in Fig. 6. The wavelet representations
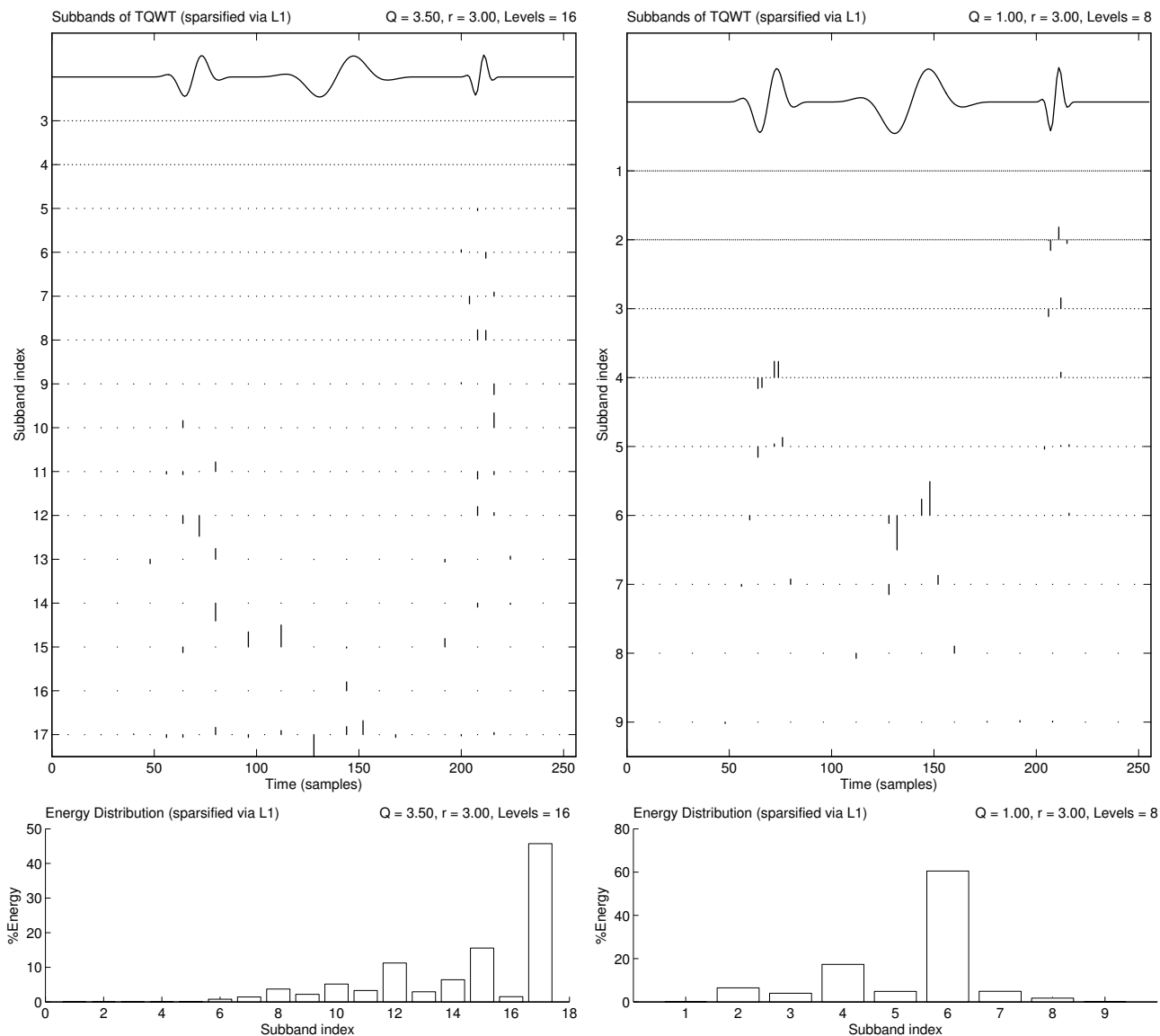
Figure 6. Coefficients (subbands) obtained via ℓ_1 norm minimization (basis pursuit) for the sparse representation of test signal 2. In comparison with Fig. 5, the TQWT coefficients are much sparser.

illustrated in Fig. 6 are clearly more sparse than those illustrated in Fig. 5. Also note that the coefficients of the low Q-factor TQWT (left) are more sparse than the coefficients of the high Q-factor TQWT (right). Specifically, note that the third pulse of test signal 2 is represented in the low Q-factor transform (left) over essentially three subbands (2–5); whereas the same pulse is represented in the high Q-factor transform (right) over about 10 subbands (6-15). The sparse low Q-factor TQWT coefficients are more concentrated than the sparse high Q-factor TQWT coefficients.

4. SIGNAL SEPARATION BASED ON SIMULTANEOUS Q-FACTORS

As described in Ref. 16, some signals may not be sparsely represented using any single Q-factor. In such cases, the use of two simultaneous distinct Q-factors may better represent the behavior of a given signal. For signals that can be represented as a sum of two separate signal components, each having its own characteristic Q-factor,

two wavelet transforms with corresponding Q-factors can be used. A dual Q-factor representation that jointly utilizes two TQWTs, with low and high Q-factors respectively, permits the sparse representation of such signals. In addition, a sparse dual Q-factor representation can be used to *separate* the low and high Q-factor signal components. The general approach of using a dual sparse signal representations for nonlinear signal separation is known as *morphological component analysis* (MCA) as described in Ref. 19. Several algorithms have been described for MCA, see Refs. 5, 7, 8, 11, 18–20

Given an observed signal $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, the goal of MCA is to estimate $\mathbf{x}_1$ and $\mathbf{x}_2$ individually. Assuming $\mathbf{x}_1$ and $\mathbf{x}_2$ can be sparsely represented using transforms Φ_1 and Φ_2 respectively, there are several ways to perform MCA of a given signal $\mathbf{x}$. Given $\mathbf{x}$, one approach is to find sparse $\mathbf{w}_1$ and sparse $\mathbf{w}_2$ such that $\mathbf{x} = \Phi_1^*\mathbf{w}_1 + \Phi_2^*\mathbf{w}_2$. Following the basis pursuit approach, the coefficient vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ can be estimated by ℓ_1-norm minimization. The problem can be formulated as:

$$\underset{\mathbf{w}_1,\mathbf{w}_2}{\operatorname{argmin}} \quad \|\boldsymbol{\lambda}_1 \odot \mathbf{w}_1\|_1 + \|\boldsymbol{\lambda}_2 \odot \mathbf{w}_2\|_1 \tag{4a}$$

$$\text{such that} \quad \mathbf{x} = \Phi_1^*\mathbf{w}_1 + \Phi_2^*\mathbf{w}_2 \tag{4b}$$

Then $\mathbf{x}_1$ and $\mathbf{x}_2$ are estimated as $\hat{\mathbf{x}}_1 = \Phi_1^*\mathbf{w}_1$ and $\hat{\mathbf{x}}_2 = \Phi_2^*\mathbf{w}_2$. A SALSA-type algorithm to solve (4) is given by (17) in Appendix C.

We have found that in order to obtain an accurate decomposition it is important to set the parameters (4) appropriately. First, the parameters of each wavelet transform Φ_i must be set appropriately — these parameters are the Q-factor, redundancy, and the number of levels of each of the two wavelet transforms. Some discussion of the influence of these parameters is given in Ref. 16. Secondly, the weighting parameters $\boldsymbol{\lambda}_i$ in (4a) must be set appropriately. We have used

$$\boldsymbol{\lambda}_1(k) = \theta \, \|\psi_1(k)\|_2, \qquad \boldsymbol{\lambda}_2(k) = (1 - \theta) \, \|\psi_2(k)\|_2$$

where $\psi_1(k)$ represents the 'wavelet' corresponding to wavelet coefficient $\mathbf{w}_1(k)$; that is, the k-th row of Φ_1. Similarly for $\psi_2(k)$. The scalar parameter θ has the purpose of tuning the relative energy of the two estimated components $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$. Normally, θ is set to a value around one half, however, adjusting θ slightly can improve the accuracy of the decomposition.

When it is known or expected that a sought component $\mathbf{x}_i$ is oscillatory, then the energy in the low-pass band of the wavelet transform Φ_i is expected to be negligible. Therefore, it can be useful to set a high value of $\boldsymbol{\lambda}_i(k)$ for low-pass wavelet coefficients in order to minimize the low-pass subband energy.

Figure 7 illustrates an example of dual Q-factor signal decomposition (referred to as 'resonance decomposition' in Ref. 16). The test signal in Fig. 7 is the sum of the two test signals illustrated in Fig. 2; this test signal is therefore not sparsely represented with any single Q-factor. Using a sparse representation using two simultaneous TQWTs, the individual Q-factor components can be estimated using the algorithm (17). The decomposition is somewhat dependent on the selection of the TQWT parameters. To obtain the result illustrated in Fig. 7 we used the parameters $Q_1 = 4.0$, $r_1 = 3.0$ and $Q_2 = 1.0$, $r_2 = 3.0$ for the two wavelet transforms Φ_1 and Φ_2. For each transform, we used the maximum number of levels (scales). We used $\theta = 0.45$. In addition, we used a high value for $\boldsymbol{\lambda}_i(k)$ in (4a) for the low-pass wavelet coefficients in order to minimize the low-pass energy; reflecting our prior knowledge that the signal components are oscillatory (band-pass). This extra penalty of low-pass coefficients influences the decomposition particularly when the signal $\mathbf{x}$ is short enough so as to impose a limit on the number of wavelet transform levels.

5. CONCLUSION

Using the tunable Q-factor wavelet transform (TQWT), the Q-factor of the wavelet transform can be matched to the oscillatory behavior of the signal under analysis. In particular, by oscillatory behavior we refer to the degree to which the oscillations are sustained. By matching the Q-factor of the wavelet transform to the signal, the sparsity of a sparse representation can be enhanced. This is illustrated by way of example in Figs. 3–6.

The use of the Q-factor as a parameter for describing the relative behavior of signals also leads to a new type of signal decomposition: the decomposition of a signal into two (or more) Q-factor components (or 'resonance
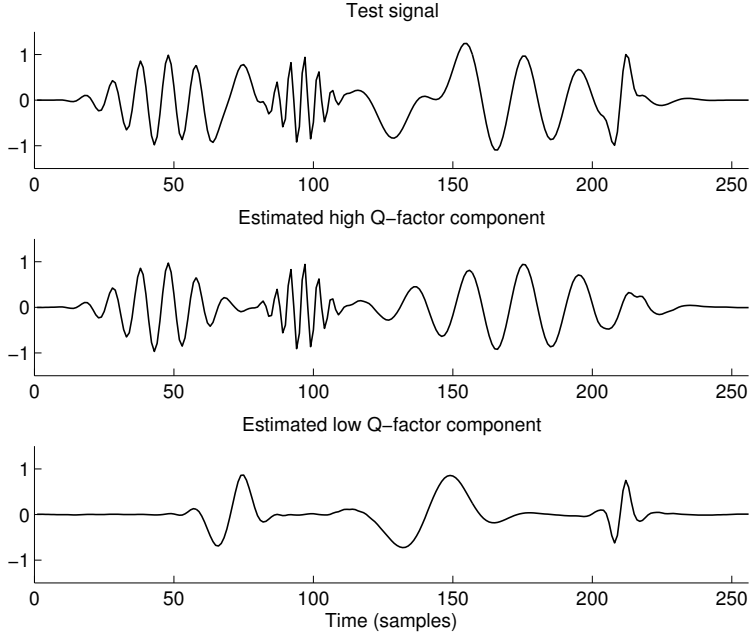
Figure 7. Example: Dual Q-factor decomposition. The test signal is the sum of the two test signals illustrated in Fig. 2. The estimated high Q-factor and low Q-factor signal components are quite similar to the individual components.

components' as in Ref. 16). Sparse signal representation with dual Q-factor TQWTs enables this (necessarily nonlinear) signal decomposition, as illustrated in Fig. 7.

APPENDIX A. AUGMENTED LAGRANGIAN

The combination of *augmented Lagrangian* and variable splitting has been demonstrated in Ref. 1 to be an effective tool for solving linear inverse problems with sparse regularization. It is notable due to (1) its flexibility in handling various problems and (2) its fast convergence in many practical examples.

For the constrained optimization problem

$$\operatorname*{argmin}_{\mathbf{z}} \quad E(\mathbf{z}) \tag{5a}$$

$$\text{such that} \quad \mathbf{Cz} - \mathbf{b} = \mathbf{0}, \tag{5b}$$

the augmented Lagrangian is given by

$$\mathcal{L}_A(\mathbf{z}, \boldsymbol{\lambda}, \mu) = E(\mathbf{z}) + \boldsymbol{\lambda}^t(\mathbf{Cz} - \mathbf{b}) + \mu \|\mathbf{Cz} - \mathbf{b}\|_2^2.$$

Version-2 of the *augmented Lagrangian method* (ALM), as described in Ref. 1, is an iterative algorithm for solving (5). ALM is given by the following update equations,

$$\text{initialize: } \mu > 0, \mathbf{d}$$

$$\mathbf{z} \leftarrow \operatorname*{argmin}_{\mathbf{z}} \ E(\mathbf{z}) + \mu \|\mathbf{Cz} - \mathbf{d}\|_2^2 \tag{6a}$$

$$\mathbf{d} \leftarrow \mathbf{d} - (\mathbf{Cz} - \mathbf{b}) \tag{6b}$$

$$\text{repeat}$$

This method is also known as the *method of multipliers* (MM). The iterative algorithm (6) is referred to as ALM/MM in Ref. 1.

APPENDIX B. SPARSE REPRESENTATION USING SALSA

The algorithm SALSA[1] can be adapted to the basis pursuit problem as follows. The basis pursuit problem (2) can be written as

$$\underset{\mathbf{w},\mathbf{u}}{\operatorname{argmin}} \quad \|\boldsymbol{\lambda} \odot \mathbf{u}\|_1 \tag{7a}$$

$$\text{such that} \quad \Phi^* \mathbf{w} = \mathbf{x} \tag{7b}$$

$$\mathbf{u} - \mathbf{w} = \mathbf{0}. \tag{7c}$$

We will use the 'partly' augmented Lagrangian,

$$\mathcal{L}_A(\mathbf{w},\mathbf{u},\boldsymbol{\lambda},\mu) = \|\boldsymbol{\lambda} \odot \mathbf{u}\|_1 + \boldsymbol{\lambda}^*(\mathbf{u} - \mathbf{w}) + \mu\|\mathbf{u} - \mathbf{w}\|_2^2 + \boldsymbol{\lambda}_2(\Phi^* \mathbf{w} - \mathbf{x}).$$

Using ALM/MM, we obtain the algorithm,

$$\text{initialize: } \mu > 0, \mathbf{d}$$

$$\mathbf{w}, \mathbf{u} \leftarrow \begin{cases} \underset{\mathbf{w},\mathbf{u}}{\operatorname{argmin}} & \|\boldsymbol{\lambda} \odot \mathbf{u}\|_1 + \mu\|\mathbf{u} - \mathbf{w} - \mathbf{d}\|_2^2 \\ \text{such that} & \Phi^* \mathbf{w} = \mathbf{x} \end{cases} \tag{8a}$$

$$\mathbf{d} \leftarrow \mathbf{d} - (\mathbf{u} - \mathbf{w}) \tag{8b}$$

$$\text{repeat}$$

We may alternate between $\mathbf{w}$ and $\mathbf{u}$ minimization as discussed in Ref. 1, to obtain the algorithm,

$$\text{initialize: } \mu > 0, \mathbf{d}$$

$$\mathbf{u} \leftarrow \underset{\mathbf{u}}{\operatorname{argmin}} \|\boldsymbol{\lambda} \odot \mathbf{u}\|_1 + \mu\|\mathbf{u} - \mathbf{w} - \mathbf{d}\|_2^2 \tag{9a}$$

$$\mathbf{w} \leftarrow \begin{cases} \underset{\mathbf{w}}{\operatorname{argmin}} & \|\mathbf{u} - \mathbf{w} - \mathbf{d}\|_2^2 \\ \text{such that} & \Phi^* \mathbf{w} = \mathbf{x} \end{cases} \tag{9b}$$

$$\mathbf{d} \leftarrow \mathbf{d} - (\mathbf{u} - \mathbf{w}) \tag{9c}$$

$$\text{repeat}$$

The solution to the minimization problem in (9a) can be expressed explicitly and compactly as soft-thresholding. That is, the minimizer $\mathbf{u}$ of $\|\boldsymbol{\lambda} \odot \mathbf{u}\|_1 + \|\mathbf{u} - \mathbf{y}\|_2^2$ is given by $\mathbf{u} = \text{soft}(\mathbf{y}, 0.5\boldsymbol{\lambda})$. Here $\text{soft}(y, T)$ is the soft-threshold rule with threshold T, which is defined as

$$\text{soft}(y, T) = y \max(0, 1 - T/|y|), \qquad y \in \mathbb{C}, \ T \in \mathbb{R}_+$$

For vector arguments, the soft-threshold rule is applied element-wise to each component pair (y_i, T_i).

The minimization problem in (9b) is a constrained least squares problem; the solution can be expressed explicitly in matrix form. With these simplifications, the algorithm (9) can be written as,

$$\text{initialize: } \mu > 0, \mathbf{d}$$

$$\mathbf{u} \leftarrow \text{soft}(\mathbf{w} + \mathbf{d}, 0.5\boldsymbol{\lambda}/\mu) \tag{10a}$$

$$\mathbf{w} \leftarrow (\mathbf{u} - \mathbf{d}) + \Phi\,(\Phi^*\Phi)^{-1}(\mathbf{x} - \Phi^*(\mathbf{u} - \mathbf{d})) \tag{10b}$$

$$\mathbf{d} \leftarrow \mathbf{d} - (\mathbf{u} - \mathbf{w}) \tag{10c}$$

$$\text{repeat}$$

To simplify the computational redundancy in algorithm (10), define $\mathbf{v} = \mathbf{u} - \mathbf{d}$. This gives the update equations:

$$\text{initialize: } \mu > 0, \mathbf{d}$$
$$\mathbf{v} \leftarrow \text{soft}(\mathbf{w} + \mathbf{d}, 0.5\boldsymbol{\lambda}/\mu) - \mathbf{d} \tag{11a}$$
$$\mathbf{w} \leftarrow \mathbf{v} + \Phi\,(\Phi^*\Phi)^{-1}(\mathbf{x} - \Phi^*\mathbf{v}) \tag{11b}$$
$$\mathbf{d} \leftarrow \mathbf{w} - \mathbf{v} \tag{11c}$$
$$\text{repeat}$$

A slight rearrangement of (11) further simplifies the update equations:

$$\text{initialize: } \mu > 0, \mathbf{d}$$
$$\mathbf{v} \leftarrow \text{soft}(\mathbf{w} + \mathbf{d}, 0.5\boldsymbol{\lambda}/\mu) - \mathbf{d} \tag{12a}$$
$$\mathbf{d} \leftarrow \Phi\,(\Phi^*\Phi)^{-1}(\mathbf{x} - \Phi^*\mathbf{v}) \tag{12b}$$
$$\mathbf{w} \leftarrow \mathbf{d} + \mathbf{v} \tag{12c}$$
$$\text{repeat}$$

The update equations (12) constitute an iterative algorithm for the basis pursuit problem (2).

Note that $\mathbf{w}$ in (12c) always satisfies $\Phi^*\mathbf{w} = \mathbf{x}$, because

$$\Phi^*\mathbf{w} = \Phi^*\left[\Phi(\Phi^*\Phi)^{-1}(\mathbf{x} - \Phi^*\mathbf{v}) + \mathbf{v}\right] = \Phi^*\Phi(\Phi^*\Phi)^{-1}(\mathbf{x} - \Phi^*\mathbf{v}) + \Phi^*\mathbf{v} = (\mathbf{x} - \Phi^*\mathbf{v}) + \Phi^*\mathbf{v} = \mathbf{x}.$$

Tight Frame Case. Consider the problem (2) where the columns of Φ^* form a tight frame, meaning that

$$\Phi^*\,\Phi = c\,\mathbf{I}. \tag{13}$$

Then algorithm (12) becomes

$$\text{initialize: } \mu > 0, \mathbf{d}$$
$$\mathbf{u} \leftarrow \text{soft}(\mathbf{w} + \mathbf{d}, 0.5\boldsymbol{\lambda}/\mu) - \mathbf{d} \tag{14a}$$
$$\mathbf{d} \leftarrow \frac{1}{c}\Phi(\mathbf{x} - \Phi^*\mathbf{u}) \tag{14b}$$
$$\mathbf{w} \leftarrow \mathbf{d} + \mathbf{u} \tag{14c}$$
$$\text{repeat}$$

Note that algorithm (14) requires no matrix inverses; only forward and inverse transforms, Φ and Φ^*.

APPENDIX C. MORPHOLOGICAL COMPONENT ANALYSIS USING SALSA

Note that (4) is a special case of (2) with

$$\Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}, \quad \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}.$$

Here we will assume that Φ_i are self-inverting transforms (Parseval tight frames), meaning that

$$\Phi_1^*\Phi_1 = \mathbf{I}, \qquad \Phi_2^*\Phi_2 = \mathbf{I}. \tag{15}$$

That is, the columns of Φ_i form tight frames. Using (15) we can write

$$\Phi^*\Phi = \begin{bmatrix} \Phi_1^* & \Phi_2^* \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} = \Phi_1^*\Phi_1 + \Phi_2^*\Phi_2 = 2\,\mathbf{I}$$

therefore, Φ satisfies (13) with $c = 2$ (the columns of Φ^* form a tight frame). Therefore, the optimization problem (4) can be solved using algorithm (14) with $c = 2$. The algorithm becomes

initialize: $\mu > 0, \mathbf{d}$

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \leftarrow \text{soft}\left(\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}, \begin{bmatrix} 0.5\boldsymbol{\lambda}_1/\mu \\ 0.5\boldsymbol{\lambda}_2/\mu \end{bmatrix}\right) - \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} \tag{16a}$$

$$\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} \leftarrow \frac{1}{2}\begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}\left(\mathbf{x} - \begin{bmatrix} \Phi_1^* & \Phi_2^* \end{bmatrix}\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}\right) \tag{16b}$$

$$\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \tag{16c}$$

repeat

Algorithm (16) can be rewritten as:

initialize: $\mu > 0, \mathbf{d}_i$

$$\mathbf{u}_i \leftarrow \text{soft}(\mathbf{w}_i + \mathbf{d}_i, 0.5\boldsymbol{\lambda}_i/\mu) - \mathbf{d}_i, \qquad i = 1, 2 \tag{17a}$$

$$\mathbf{c} \leftarrow \mathbf{x} - \Phi_1^*\mathbf{u}_1 - \Phi_2^*\mathbf{u}_2 \tag{17b}$$

$$\mathbf{d}_i \leftarrow \frac{1}{2}\Phi_i\mathbf{c} \qquad\qquad i = 1, 2 \tag{17c}$$

$$\mathbf{w}_i \leftarrow \mathbf{d}_i + \mathbf{u}_i \qquad\qquad i = 1, 2 \tag{17d}$$

repeat

The update equations (17) constitute an iterative algorithm for the basis pursuit form of MCA.

C.1 Why not a quadratic cost function?

For sparse representation and signal decomposition via constrained optimization it is important that the cost functions (2) and (4) be non-quadratic. To illustrate this, it is informative to consider the following quadratic cost function as an alternative to (4),

$$\underset{\mathbf{w}_1, \mathbf{w}_2}{\text{argmin}} \quad \lambda_1\|\mathbf{w}_1\|_2^2 + \lambda_2\|\mathbf{w}_2\|_2^2 \tag{18a}$$

$$\text{such that} \quad \mathbf{x} = \Phi_1^*\mathbf{w}_1 + \Phi_2^*\mathbf{w}_2. \tag{18b}$$

The solution $\mathbf{w}_1$ and $\mathbf{w}_2$ to (18) can be found in closed form as follows. The Lagrangian is given by

$$\mathcal{L} = \lambda_1\|\mathbf{w}_1\|_2^2 + \lambda_2\|\mathbf{w}_2\|_2^2 + \boldsymbol{\mu}^t(\Phi_1^*\mathbf{w}_1 + \Phi_2^*\mathbf{w}_2 - \mathbf{x})$$

where $\boldsymbol{\mu}$ is a vector of Langrange multipliers. Taking the derivative of $\mathcal{L}$ with respect to $\mathbf{w}_1$, $\mathbf{w}_2$, and $\boldsymbol{\mu}$ gives

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}_1} = 2\lambda_1\mathbf{w}_1 + \Phi_1\boldsymbol{\mu}, \qquad \frac{\partial\mathcal{L}}{\partial\mathbf{w}_2} = 2\lambda_2\mathbf{w}_2 + \Phi_2\boldsymbol{\mu} \tag{19}$$

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\mu}} = \Phi_1^*\mathbf{w}_1 + \Phi_2^*\mathbf{w}_2 - \mathbf{x}. \tag{20}$$

Setting the derivatives to zero gives

$$\mathbf{w}_1 = -\frac{1}{2\lambda_1}\Phi_1\boldsymbol{\mu}, \qquad \mathbf{w}_2 = -\frac{1}{2\lambda_2}\Phi_2\boldsymbol{\mu} \tag{21}$$

$$\mathbf{x} = \Phi_1^*\mathbf{w}_1 + \Phi_2^*\mathbf{w}_2. \tag{22}$$

Substituting (21) into (22) gives

$$\mathbf{x} = -\frac{1}{2\lambda_1}\Phi_1^*\Phi_1\boldsymbol{\mu} - \frac{1}{2\lambda_2}\Phi_2^*\Phi_2\boldsymbol{\mu}$$

and using (15) gives

$$\mathbf{x} = -\frac{1}{2}\left(\frac{1}{\lambda_1} + \frac{1}{\lambda_2}\right)\boldsymbol{\mu}. \tag{23}$$

Solving (23) for the Lagrange multipliers $\boldsymbol{\mu}$ gives

$$\boldsymbol{\mu} = -\frac{2}{\left(\frac{1}{\lambda_1} + \frac{1}{\lambda_2}\right)}\mathbf{x}. \tag{24}$$

Using (24) in (21) gives an explicit expression for the optimal $\mathbf{w}_1$ and $\mathbf{w}_2$,

$$\mathbf{w}_1 = \frac{1/\lambda_1}{1/\lambda_1 + 1/\lambda_2}\,\Phi_1\mathbf{x}, \qquad \mathbf{w}_2 = \frac{1/\lambda_2}{1/\lambda_1 + 1/\lambda_2}\,\Phi_2\mathbf{x}.$$

Therefore, using (15) again, the estimated components $\hat{\mathbf{x}}_1 = \Phi_1^*\mathbf{w}_1$ and $\hat{\mathbf{x}}_2 = \Phi_2^*\mathbf{w}_2$, are given by

$$\hat{\mathbf{x}}_1 = \frac{1/\lambda_1}{1/\lambda_1 + 1/\lambda_2}\,\mathbf{x}, \qquad \hat{\mathbf{x}}_2 = \frac{1/\lambda_2}{1/\lambda_1 + 1/\lambda_2}\,\mathbf{x}.$$

Note that both $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ are simply scaled versions of $\mathbf{x}$. That is, minimizing the quadratic cost function (18) yields no signal separation whatsoever. The cost function must be non-quadratic in order to achieve the sought type of decomposition. This is to be expected because for a quadratic cost function with linear constraints, the minimizer will be a linear function of the signal $\mathbf{x}$; yet as described in Ref. 16, the sought decomposition must be a nonlinear function of the $\mathbf{x}$.

REFERENCES

1. M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. Image Process.*, 19(9):2345 –2356, September 2010.
2. I. Bayram and I. W. Selesnick. Frequency-domain design of overcomplete rational-dilation wavelet transforms. *IEEE Trans. Signal Process.*, 57(8):2957–2972, August 2009.
3. İ. Bayram and I. W. Selesnick. Overcomplete discrete wavelet transforms with rational dilation factors. *IEEE Trans. Signal Process.*, 57(1):131–145, January 2009.
4. J. C. Brown and M. S. Puckette. An efficient algorithm for the calculation of a constant Q transform. *J. Acoust. Soc. Am.*, 92(5):2698–2701, November 1992.
5. J.-F. Cai, S. Osher, and Z. Shen. Split Bregman methods and frame based image restoration. *Multiscale Model. Simul.*, 8(2):337–369, 2009.
6. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.
7. M. Elad, J. Starck, P. Querre, and D. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *J. of Appl. and Comp. Harm. Analysis*, 19(3):340–358, November 2005.
8. M. J. Fadili, J.-L. Starck, J. Bobin, and Y. Moudden. Image decomposition and separation using sparse representations: An overview. *Proc. IEEE*, 98(6):983 –994, June 2010.
9. T. Gulzow, T. Ludwig, and U. Heute. Spectral-subtraction speech enhancement in multirate systems with and without non-uniform and adaptive bandwidths. *Signal Processing*, 83(8):1613 – 1631, 2003.
10. A. Karmakar, A. Kumar, and R. K. Patney. Design of optimal wavelet packet trees based on auditory perception criterion. *IEEE Signal Processing Letters*, 14(4):240–243, April 2007.
11. G. Kutyniok. Data separation by sparse representations. In Y. C. Eldar and G. Kutyniok, editors, *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012. To appear.
12. Q. Li. An auditory-based transform for audio signal processing. In *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2009.
13. T. Petersen and S. Boll. Critical band analysis-synthesis. *IEEE Trans. on Acoust., Speech, Signal Proc.*, 31(3):656–663, June 1983.

14. Y. Ren, M. T. Johnson, and J. Tao. Perceptually motivated wavelet packet transform for bioacoustic signal enhancement. *J. Acoust. Soc. Am.*, 124:316–327, 2008.

15. C. Schörkhuber and A. Klapuri. Constant-Q transform toolbox for music processing. In *7th Sound and Music Conf.*, July 2010.

16. I. W. Selesnick. Resonance-based signal decomposition: A new sparsity-enabled signal analysis method. *Signal Processing*, 91(12):2793 – 2809, 2011.

17. I. W. Selesnick. Wavelet transform with tunable Q-factor. *Signal Processing, IEEE Transactions on*, 59(8):3560–3575, August 2011.

18. J.-L. Starck, M. Elad, and D. Donoho. Redundant multiscale transforms and their application for morphological component analysis. *Advances in Imaging and Electron Physics*, 132:287–348, 2004.

19. J.-L. Starck, M. Elad, and D. Donoho. Image decomposition via the combination of sparse representation and a variational approach. *IEEE Trans. Image Process.*, 14(10), 2005.

20. J.-L. Starck, Y. Moudden, J. Bobina, M. Elad, and D.L. Donoho. Morphological component analysis. In *Proceedings of SPIE*, volume 5914 (Wavelets XI), 2005.