DUAL TREE COMPLEX WAVELETS

NICK KINGSBURY

Signal Processing Group, Dept. of Engineering
University of Cambridge, Cambridge CB2 1PZ, UK.
ngk@eng.cam.ac.uk
www.eng.cam.ac.uk/~ngk

September 2004

UNIVERSITY OF CAMBRIDGE

Dual Tree Complex Wavelets

Nick Kingsbury
Signal Processing Group, Dept. of Engineering,
University of Cambridge, Cambridge CB2 1PZ, UK.
Email: ngk@eng.cam.ac.uk Web: www.eng.cam.ac.uk/˜ngk

Abstract

We describe the Dual Tree Complex Wavelet Transform (DT CWT), a form of discrete wavelet transform which generates complex coefficients by using a dual tree of wavelet filters to obtain their real and imaginary parts. This introduces limited redundancy ($2^m : 1$ for m-dimensional signals) and allows the transform to provide approximate shift invariance and directionally selective filters (properties lacking in the traditional wavelet transform) while preserving the usual properties of perfect reconstruction and computational efficiency with good well-balanced frequency responses.

We analyse why the new transform can be designed to be shift invariant, and describe how to estimate the accuracy of this approximation and design suitable filters to achieve this. We describe why the DT CWT is particularly suitable for images and other multi-dimensional signals, and summarise some applications of the transform that take advantage of its unique properties, including denoising, sparse coding, and registration.

Features of the (Real) Discrete Wavelet Transform (DWT)

- **Good compression** of signal energy.
- **Perfect reconstruction** with short support filters.
- **No redundancy**.
- **Very low computation** – order-N only.

But

- **Severe shift dependence**.
- **Poor directional selectivity** in 2-D, 3-D etc.

The DWT is normally implemented with a tree of highpass and lowpass filters, separated by $2:1$ decimators.

Real Discrete Wavelet Transform (DWT) in 1-D

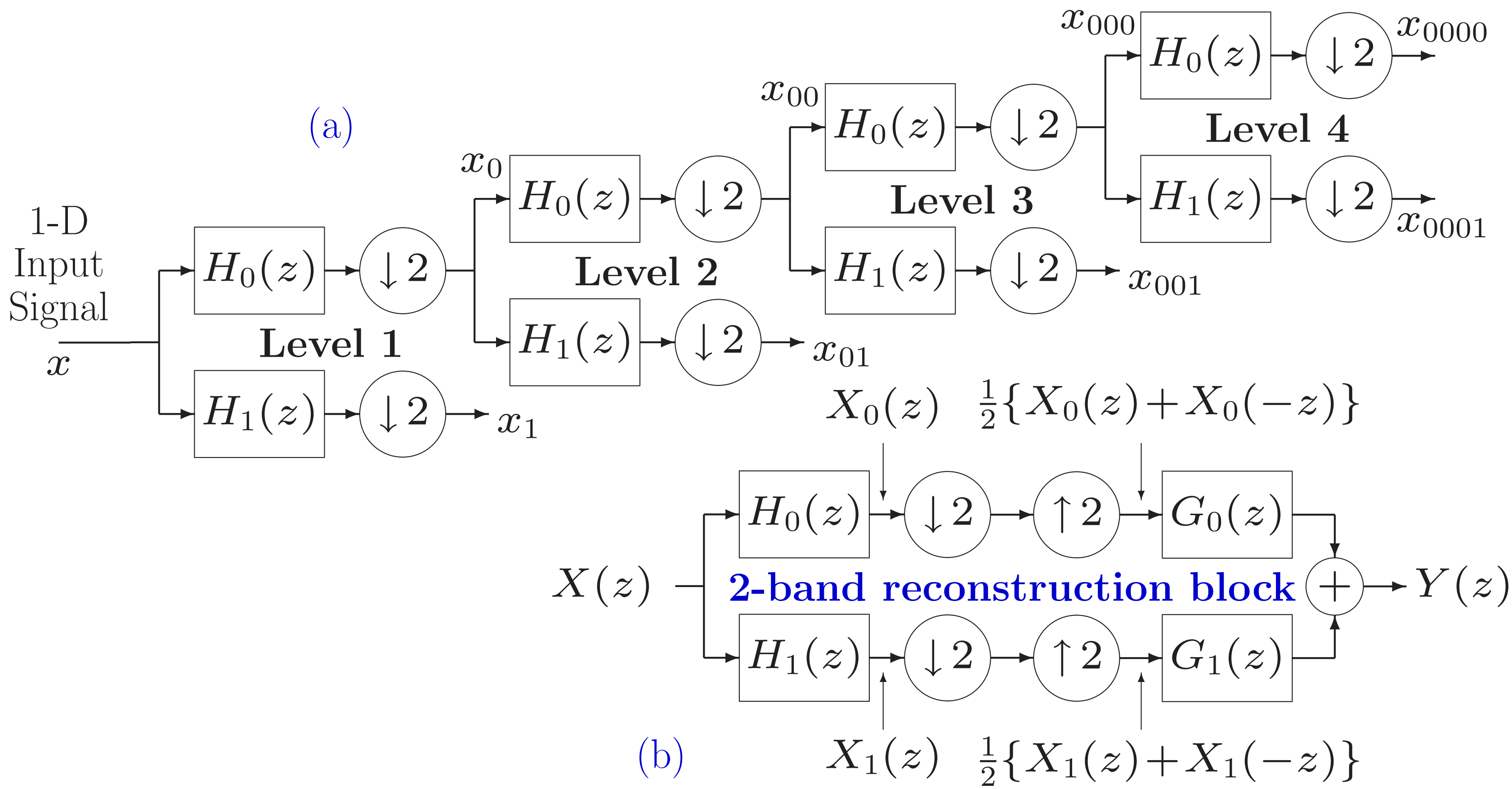


Figure 1: (a) Tree of real filters for the DWT. (b) Reconstruction filter block for 2 bands at a time, used in the inverse transform.

Features of the Dual Tree Complex Wavelet Transform (DT CWT)

- Good **shift invariance**.
- Good **directional selectivity** in 2-D, 3-D etc.
- **Perfect reconstruction** with short support filters.
- **Limited redundancy** – 2:1 in 1-D, 4:1 in 2-D etc.
- **Low computation** – much less than the undecimated (à trous) DWT.

Each tree contains purely real filters, but the two trees produce the **real and imaginary parts** respectively of each complex wavelet coefficient.

Q-shift Dual Tree Complex Wavelet Transform in 1-D

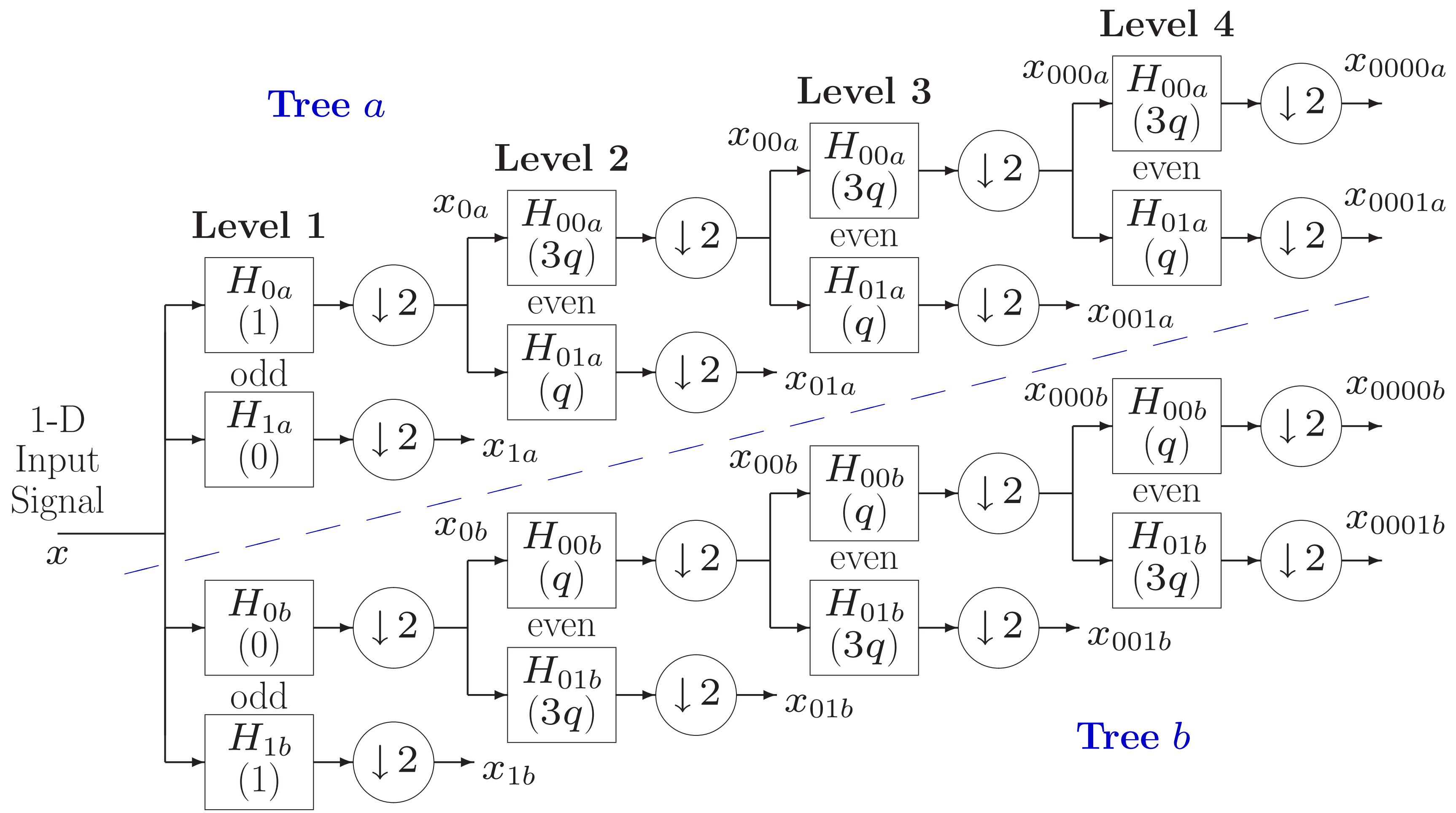


Figure 2: Dual tree of real filters for the Q-shift CWT, giving real and imaginary parts of complex coefficients from tree a and tree b respectively. Figures in brackets indicate the approximate delay for each filter, where $q = \frac{1}{4}$ sample period.

Features of the Q-shift Filters

Below level 1:

- Half-sample delay difference is obtained with filter delays of $\frac{1}{4}$ and $\frac{3}{4}$ of a sample period (instead of 0 and $\frac{1}{2}$ a sample for our original DT CWT).
- This is achieved with an **asymmetric even-length** filter $H(z)$ and its time reverse $H(z^{-1})$.
- Due to the asymmetry (like Daubechies filters), these may be designed to give an **orthonormal perfect reconstruction** wavelet transform.
- Tree **b** filters are the **reverse** of tree **a** filters, and reconstruction filters are the reverse of analysis filters, so **all filters** are from the **same orthonormal set**.
- Both trees have the **same frequency responses**.
- **Symmetric sub-sampling** – see below.

Q-shift DT CWT Basis Functions

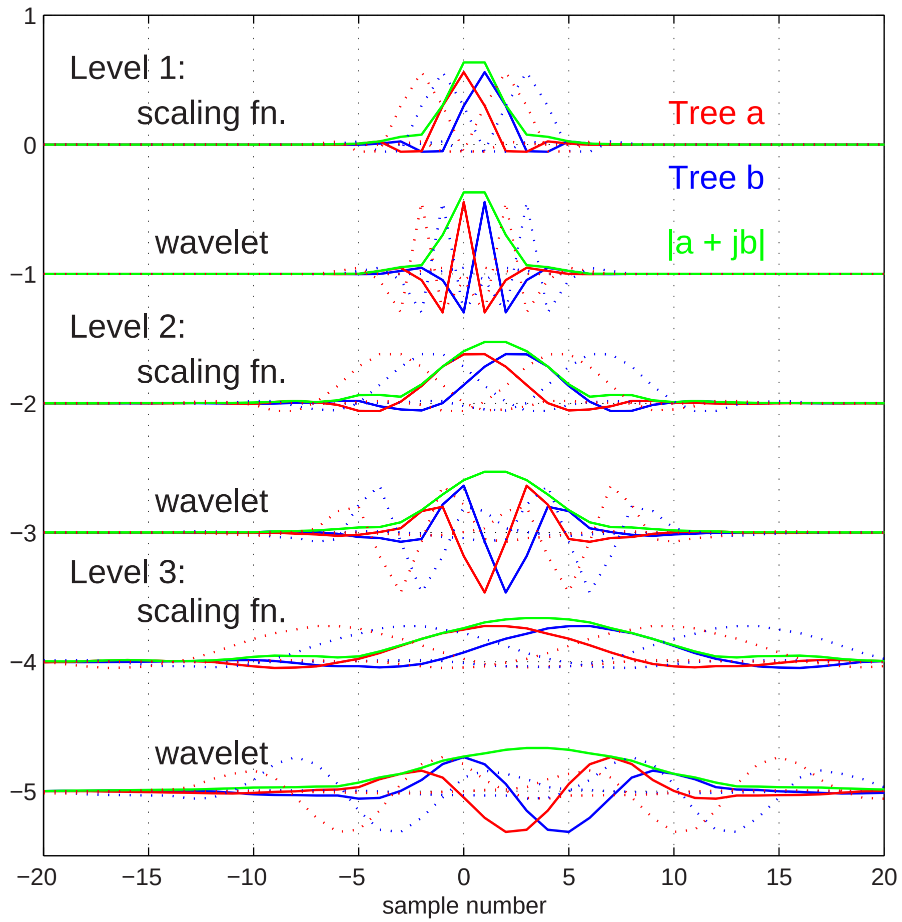


Figure 3: Basis functions (reconstruction impulse responses) of the Q-shift DT CWT filters for levels 1 to 3. Tree a bases are shown in red and tree b in blue. The magnitudes of the complex bases, formed by combining the two trees, are in green. Bases for adjacent sampling points are shown dotted.

Sampling Symmetries

In a regular multi-resolution pyramid structure each parent coefficient must lie symmetrically below the mean position of its 2 children (4 children in 2-D). Each filter should also be symmetric about its mid point.

- For the **Q-shift filters**, fig 3 shows that parents are symmetrically below their children, and that Hi and Lo filters at each level are aligned correctly.
- Since one Q-shift tree is the time-reverse of the other, the combined **complex** impulse responses are **conjugate symmetric** about their mid points, even though the separate responses are asymmetric (see fig 3, right). Hence **symmetric extension** is still an effective technique at image edges.

Q-shift DT CWT Filter Design

For the two trees we need lowpass filters with group delays which differ by **half a sample period**. This ensures low aliasing energy and hence good shift invariance.

The **Q-shift** version of the DT CWT achieves this with filters with group delays $\simeq\ \frac{1}{4}$ **and** $\frac{3}{4}$ of a sample period, and has the following additional features:

- **Tree b** filters are the time-reverse of the **Tree a** filters.
- **Reconstruction** filters are the time-reverse of the **Analysis** filters.
- Bases are **orthonormal**, yielding a **tight-frame** transform.
- The complex bases are **linear phase**, since their magnitudes are symmetric and their phases are anti-symmetric (with a 45 degree offset).

Q-shift Filter Design Requirements

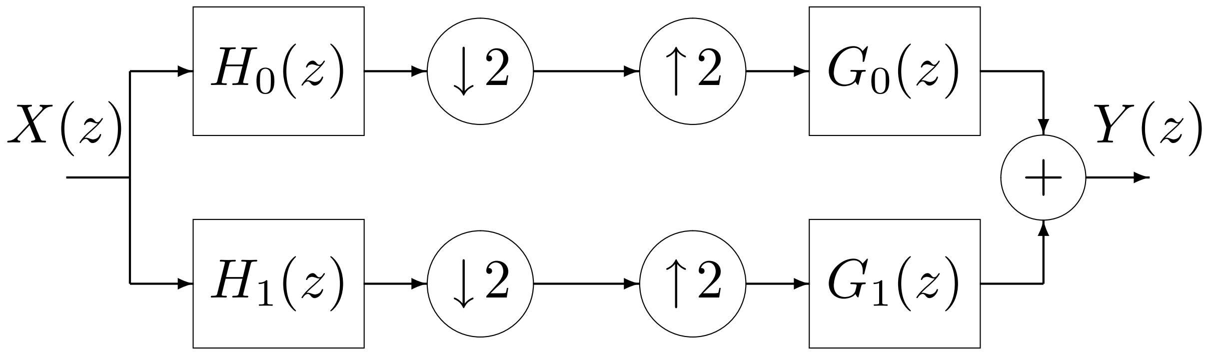


Fig. 2: 2-band analysis and reconstruction filter banks.

1. **No aliasing:** $G_1(z) = zH_0(-z)$; $H_1(z) = z^{-1}G_0(-z)$
2. **Perfect reconstruction:** $H_0(z)G_0(z) + H_0(-z)G_0(-z) = 2$
3. **Orthogonality:** $G_0(z) = H_0(z^{-1})$
4. Group delay $\simeq \frac{1}{4}$ **sample period** for H_0.
5. **Good smoothness** properties when iterated over scale.

FILTER DESIGN — DELAY

To get $2n$-tap lowpass filters, $H_0(z)$ and $G_0(z)$, with $\frac{1}{4}$ and $\frac{3}{4}$ sample delays:

- Design a **$4n$-tap** symmetric lowpass filter $H_{L2}(z)$ with half the required bandwidth and a delay of $\frac{1}{2}$ sample;
- **Subsample** $H_{L2}(z)$ by 2:1 to get $H_0(z)$ and $G_0(z)$.

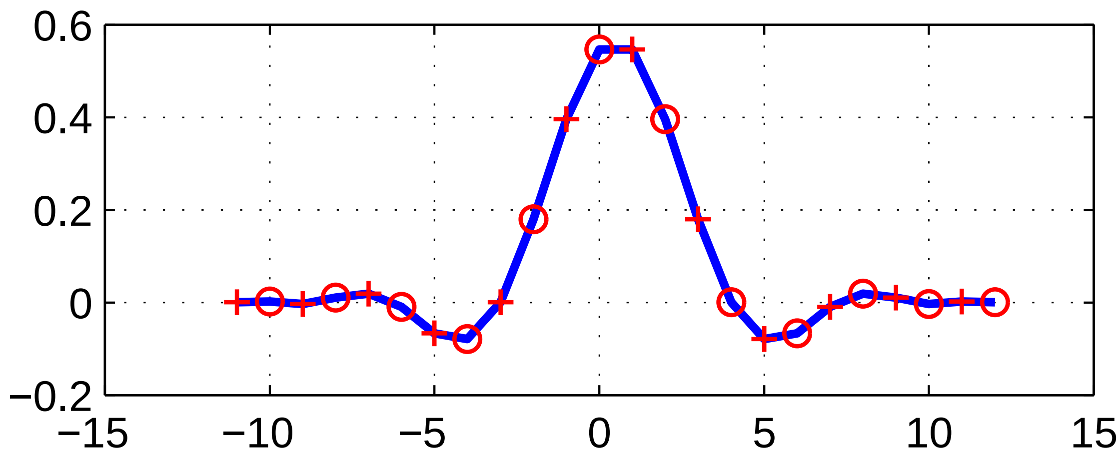


Fig. 3: Impulse response of $H_{L2}(z)$ for $n = 6$. The H_0 and G_0 filter taps are shown as circles and crosses respectively.

FILTER DESIGN – PERFECT RECONSTRUCTION (PR)

For PR and orthogonality:

$H_0(z)\,G_0(z) = H_0(z)\,H_0(z^{-1})$ must have **no terms in** z^{2k} except the term in z^0.

$$\therefore\ H_0(z^2)\,H_0(z^{-2}) \text{ must have } \textbf{no terms in } z^{4k} \text{ except the term in } z^0.$$

But

$$H_{L2}(z) = H_0(z^2) + z^{-1} H_0(z^{-2})$$

and so

$$H_{L2}(z)\,H_{L2}(z^{-1}) = 2\,H_0(z^2)\,H_0(z^{-2}) + \underbrace{z^{-1}H_0^2(z^{-2}) + zH_0^2(z^2)}_{\textbf{odd powers of } z \textbf{ only}}$$

$$\therefore\ H_{L2}(z)\,H_{L2}(z^{-1}) \text{ must have } \textbf{no terms in } z^{4k} \text{ except the term in } z^0.$$

Hence we can include PR as a **direct design constraint on** $H_{L2}(z)\,H_{L2}(z^{-1})$.

Filter Design — Smoothness

To obtain smoothness when iterated over many scales:

- **Ensure that the stopband of $H_0(z)$ suppresses energy at frequencies where unwanted passbands appear from subsampled filters operating at coarser scales.**

Consider the combined frequency response of H_0 over just two scales:

$$H_0(z)\, H_0(z^2)|_{z=e^{j\omega}} = H_0(e^{j\omega})\, H_0(e^{2j\omega})$$

If the stopband of $H_0(e^{j\omega})$ covers $\omega_s \leq \omega \leq \pi$, then the unwanted transition band and passband of $H_0(e^{2j\omega})$ will extend from $\pi - \frac{\omega_s}{2}$ to π.

For $H_0(e^{j\omega})$ to suppress the unwanted bands of $H_0(e^{2j\omega})$ (see fig. 4):

$$\omega_s \leq \pi - \frac{\omega_s}{2} \qquad\qquad \therefore\ \boldsymbol{\omega_s \leq \frac{2\pi}{3}}$$

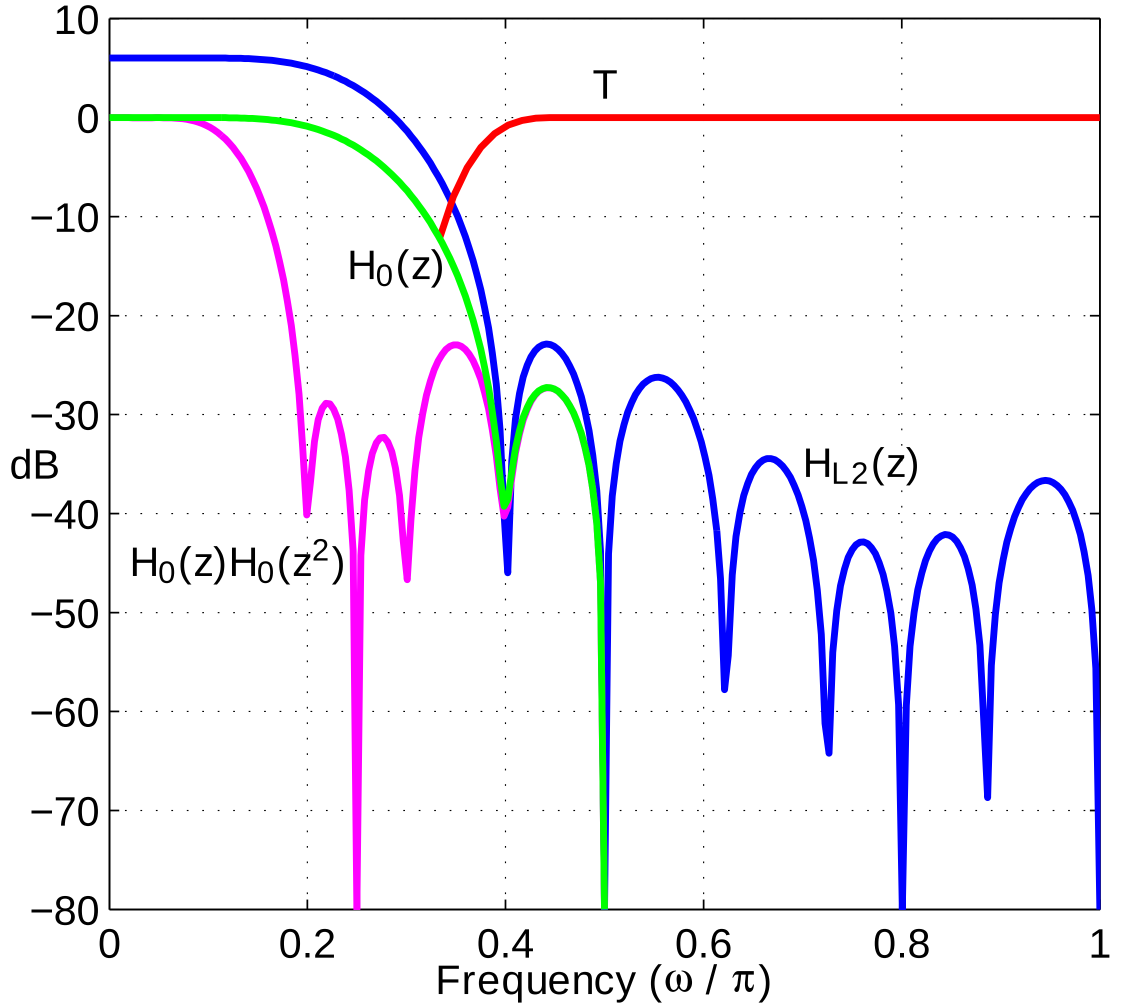


Fig. 4: Frequency responses of $H_{L2}(z)$ (blue), $H_0(z)$ (green), $H_0(z)\ H_0(z^2)$ (magenta), and the gain correction matrix $\mathbf{T}$ (red) for $n = 6$ (12 taps for H_0).

Optimization for MSE in the frequency domain

We have now reduced the ideal design conditions for the length $4n$ symmetric lowpass filter H_{L2} to be:

- Zero amplitude for all the terms of $H_{L2}(z)\, H_{L2}(z^{-1})$ in z^{4k} except the term in z^0, which must be 1 (these are **quadratic constraints** on coef vector $\mathbf{h}_{L2}$);
- Zero (or near-zero) amplitude of $H_{L2}(e^{j\omega})$ for the stopband, $\frac{\pi}{3} \leq \omega \leq \pi$ (these are **linear constraints** on $\mathbf{h}_{L2}$).

If all constraints were linear, the LMS error solution for $\mathbf{h}_{L2}$ could be found using a matrix pseudo-inverse method. $\therefore$ **we linearise the problem and iterate.**

If $\mathbf{h}_{L2}$ at iteration i is $\mathbf{h}_i = \mathbf{h}_{i-1} + \Delta\mathbf{h}_i$, then

$$\mathbf{h}_i * \mathbf{h}_i = (\mathbf{h}_{i-1} + \Delta\mathbf{h}_i) * (\mathbf{h}_{i-1} + \Delta\mathbf{h}_i) = \mathbf{h}_{i-1} * (\mathbf{h}_{i-1} + 2\Delta\mathbf{h}_i) + \Delta\mathbf{h}_i * \Delta\mathbf{h}_i$$

Since **$\Delta\mathbf{h}_i$ becomes small** as i increases, the final term can be neglected and the convolution ($*$) is expressed as a **linear function of $\Delta\mathbf{h}_i$**.

Hence we solve for $\mathbf{\Delta h}_i$ such that:

$$\begin{aligned} \mathbf{C}_{i-1}\ (\mathbf{h}_{i-1} + 2\mathbf{\Delta h}_i) &= [0 \ldots 0\ 1]^T \\ \mathbf{F}\ (\mathbf{h}_{i-1} + \mathbf{\Delta h}_i) &\simeq [0 \ldots 0]^T \end{aligned}$$

where $\mathbf{C}_{i-1}$ calculates every 4th term in the convolution with $\mathbf{h}_{i-1}$, and $\mathbf{F}$ evaluates the Fourier transform at M discrete frequencies ω from $\frac{\pi}{3}$ to π (typically $M \simeq 8n$)

Note that only one side of the symmetric convolution is needed in the rows of $\mathbf{C}_{i-1}$, and the columns of $\mathbf{C}_{i-1}$ and $\mathbf{F}$ can be combined in pairs so that only the first half of the symmetric $\mathbf{\Delta h}_i$ need be solved for.

To obtain **high accuracy solutions to the PR constraints**, we scale the equations in $\mathbf{C}_{i-1}$ up by $\beta_i = 2^i$ to get the following iterative LMS method for $\mathbf{\Delta h}_i$ and then $\mathbf{h}_i$:

$$\begin{bmatrix} 2\beta_i \mathbf{C}_{i-1} \\ \mathbf{F} \end{bmatrix} \mathbf{\Delta h}_i = \begin{bmatrix} \beta_i(\mathbf{c} - \mathbf{C}_{i-1}\mathbf{h}_{i-1}) \\ -\mathbf{F}\ \mathbf{h}_{i-1} \end{bmatrix} \quad \text{with} \quad \mathbf{h}_i = \mathbf{h}_{i-1} + \mathbf{\Delta h}_i$$

where $\mathbf{c} = [0 \ldots 0\ 1]^T$.

Two Final Refinements

- To include **transition band** effects, we scale rows of $\mathbf{F}$ by diagonal matrix $\mathbf{T}_i$, the gain (at iteration i) of $H_0(z^2)/H_0(1)$ at frequencies corresponding to $\frac{\pi}{3} \leq \omega \leq \frac{\pi}{2}$ in the frequency domain of H_{L2} ($\mathbf{T}_i$ is the red curve in fig. 4).

- To insert **predefined zeros** in $H_0(z)$ or $H_{L2}(z)$, we first note that a zero at $z = e^{j\pi}$ in H_0 will be produced by a pair of zeros at $z = e^{\pm j\pi/2}$ in H_{L2}. We can force zeros in H_{L2} by forming a convolution matrix $\mathbf{H}_f$ such that $\mathbf{H}_f \; \mathbf{h}'_i = \mathbf{h}_i$, where $\mathbf{h}'_i$ is the coef vector of the filter which represents all the zeros of H_{L2} that are **not** predefined, and $\mathbf{H}_f$ produces convolution with the predefined zeros.

Hence we now solve for $\Delta\mathbf{h}'_i$ and then $\mathbf{h}_i$ using

$$\begin{bmatrix} 2\beta_i \mathbf{C}_{i-1} \\ \mathbf{T}_{i-1}\mathbf{F} \end{bmatrix} \mathbf{H}_f \; \Delta\mathbf{h}'_i = \begin{bmatrix} \beta_i(\mathbf{c} - \mathbf{C}_{i-1}\mathbf{h}_{i-1}) \\ -\mathbf{T}_{i-1}\mathbf{F} \; \mathbf{h}_{i-1} \end{bmatrix} \quad \text{with} \quad \mathbf{h}_i = \mathbf{h}_{i-1} + \mathbf{H}_f \; \Delta\mathbf{h}'_i$$

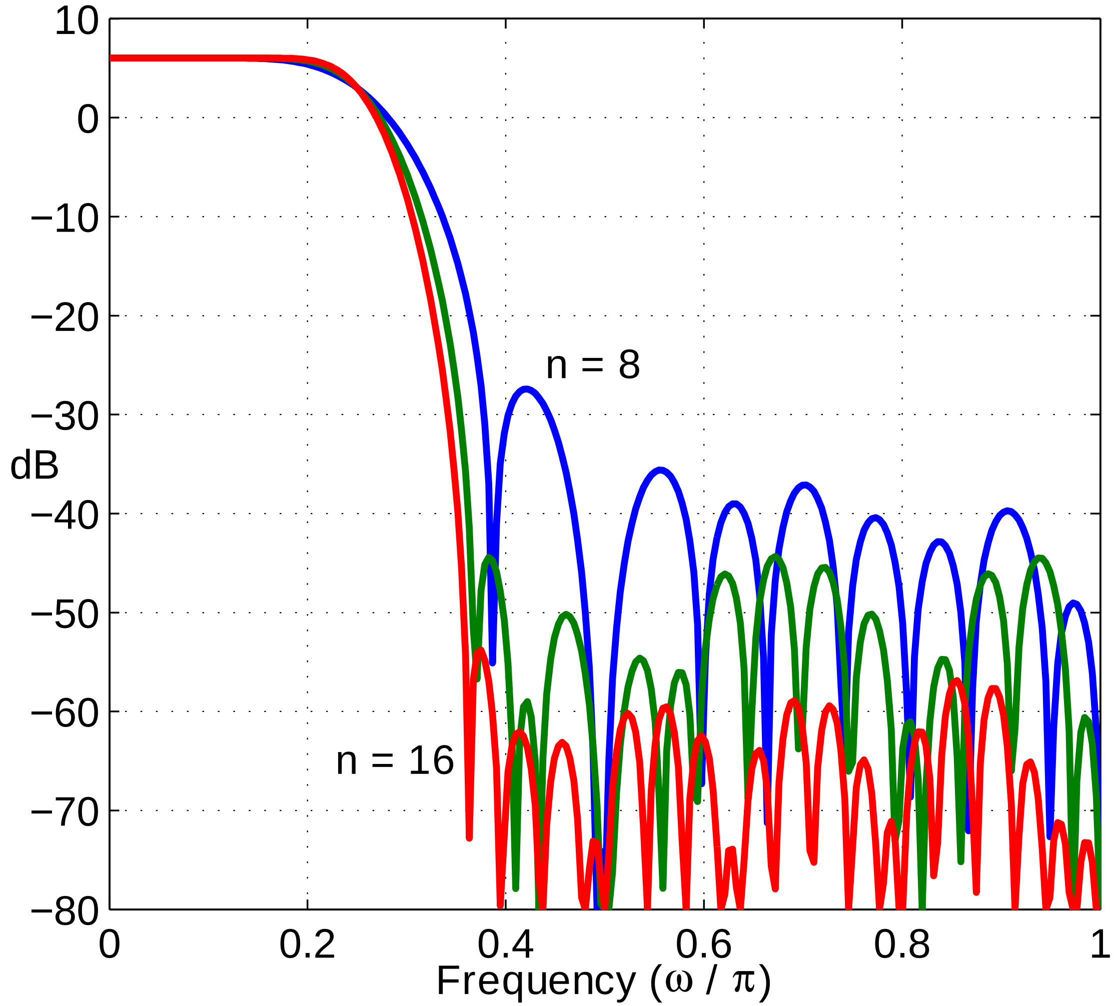


Fig. 5: Frequency responses of $H_{L2}(z)$ for $n = 8$ (blue), $n = 12$ (green) and $n = 16$ (red). Each filter has one predefined zero at $\omega = \frac{\pi}{2}$ and one at $\omega = \pi$.

Initialisation

To initialise the iterative algorithm when $i = 1$, we must define $\mathbf{h}_0$ and hence $\mathbf{C}_0$ and $\mathbf{T}_0$.

This is not critical and can be achieved by a simple inverse FFT of an 'ideal' lowpass frequency response for $H_{L2}(e^{j\omega})$ with a root-raised-cosine transition band covering the range

$$\tfrac{\pi}{6} < \omega < \tfrac{\pi}{3}$$

The impulse response is truncated symmetrically to length $4n$ to obtain $\mathbf{h}_0$.

$\mathbf{C}_0$ and $\mathbf{T}_0$ may then be calculated from $\mathbf{h}_0$.

Convergence

For some larger values of n, convergence can be slow. We have found this can be improved by using

$$\mathbf{h}_i = \mathbf{h}_{i-1} + \alpha \mathbf{H}_f \, \Delta\mathbf{h}'_i \qquad \text{where} \quad 0 < \alpha < 1 \quad (\text{e.g. } \alpha \sim 0.8)$$

Results

- Figs. 4 and 5 show the frequency responses of $H_{L2}(z)$ for the cases $n = 6$, 8, 12 and 16, when there is one predefined zero at $\omega = \frac{\pi}{2}$ and one at $\omega = \pi$.

- Figs. 6 to 15 show, for a range of values of n, the impulse response of $H_{L2}(z)$, the level-4 DT CWT scaling functions and wavelets, the frequency responses of $H_0(z)$ and of $H_0(z)\,H_0(z^2)$, and the group delay of $H_0(z)$.

- Figs. 6 to 11 show these responses for the cases $n = 5$, 6 and 7, with either 0 or 1 predefined zero in $H_0(z)$ at $\omega = \pi$.

- Figs. 12 to 15 show these responses for the cases $n = 8$, 12 and 16, with 1 predefined zero in $H_0(z)$ at $\omega = \pi$.

Note how the responses improve with increasing n. The effect of predefining a zero in H_0 is in general quite small. More predefined zeros tend to degrade performance.

$n = 7$ gives a good tradeoff between complexity and performance.

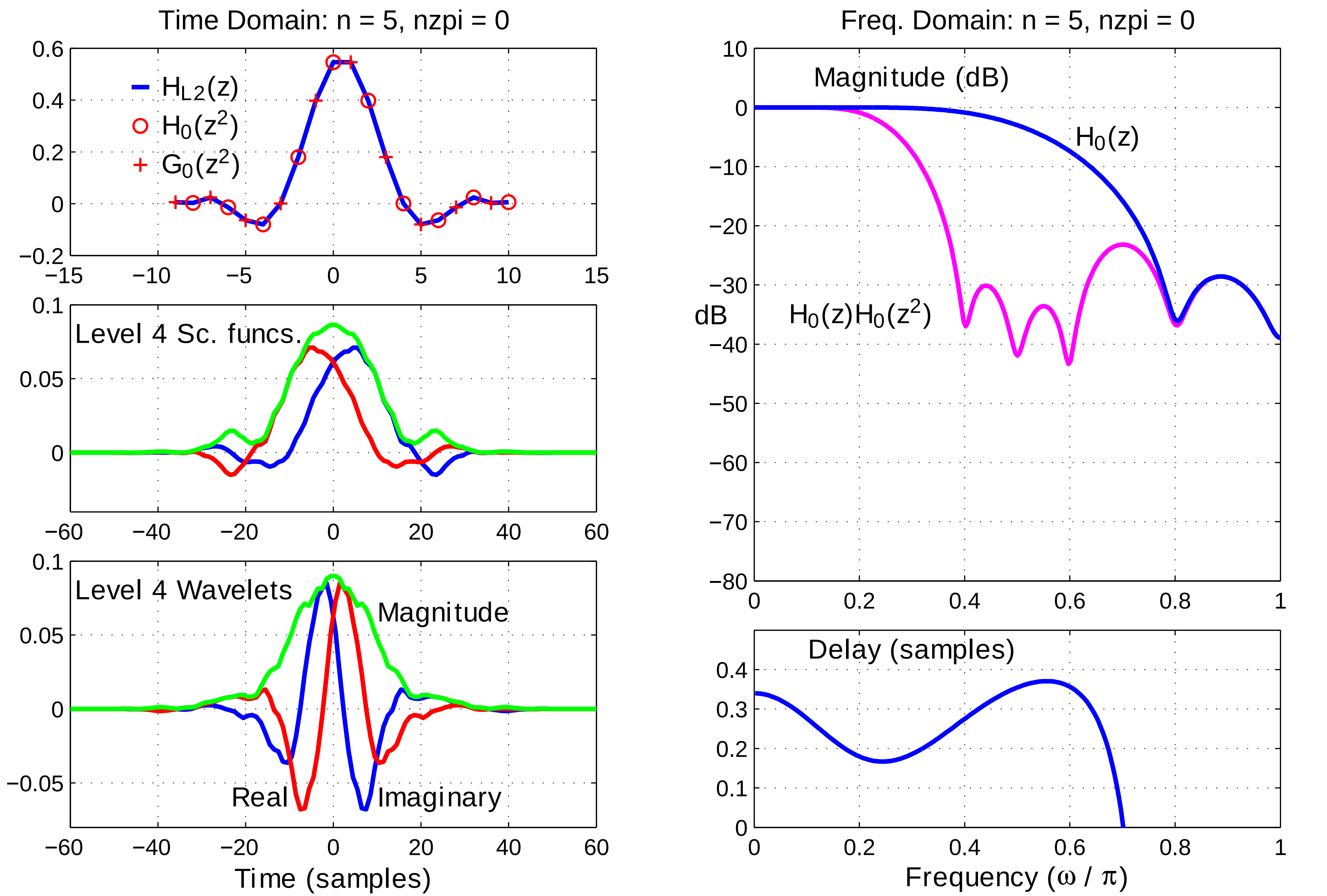


Fig. 6: Q-shift filters for $n = 5$ (10 filter taps) and no predefined zeros.

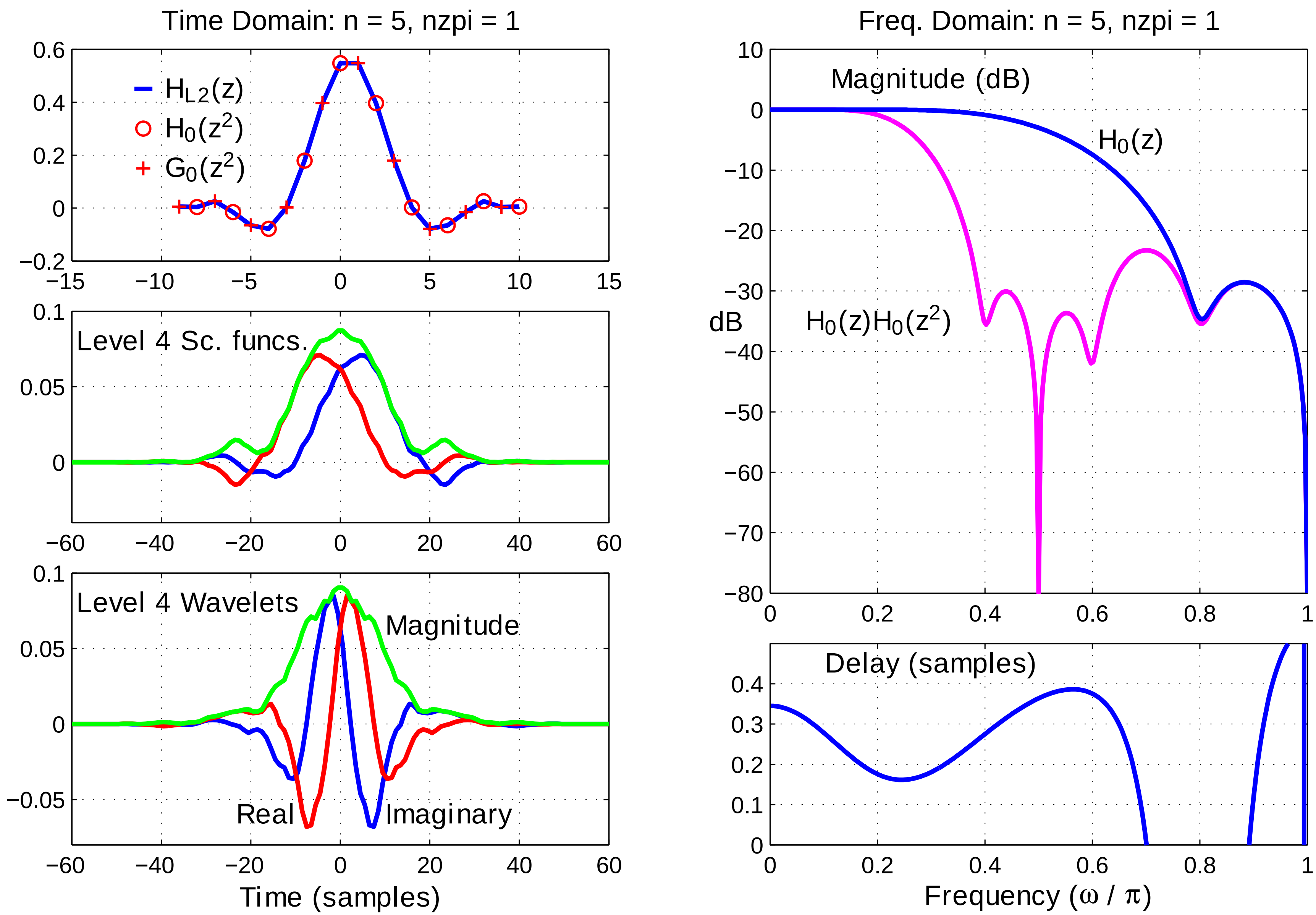


Fig. 7: Q-shift filters for $n = 5$ (10 filter taps) and 1 predefined zero at $\omega = \pi$.

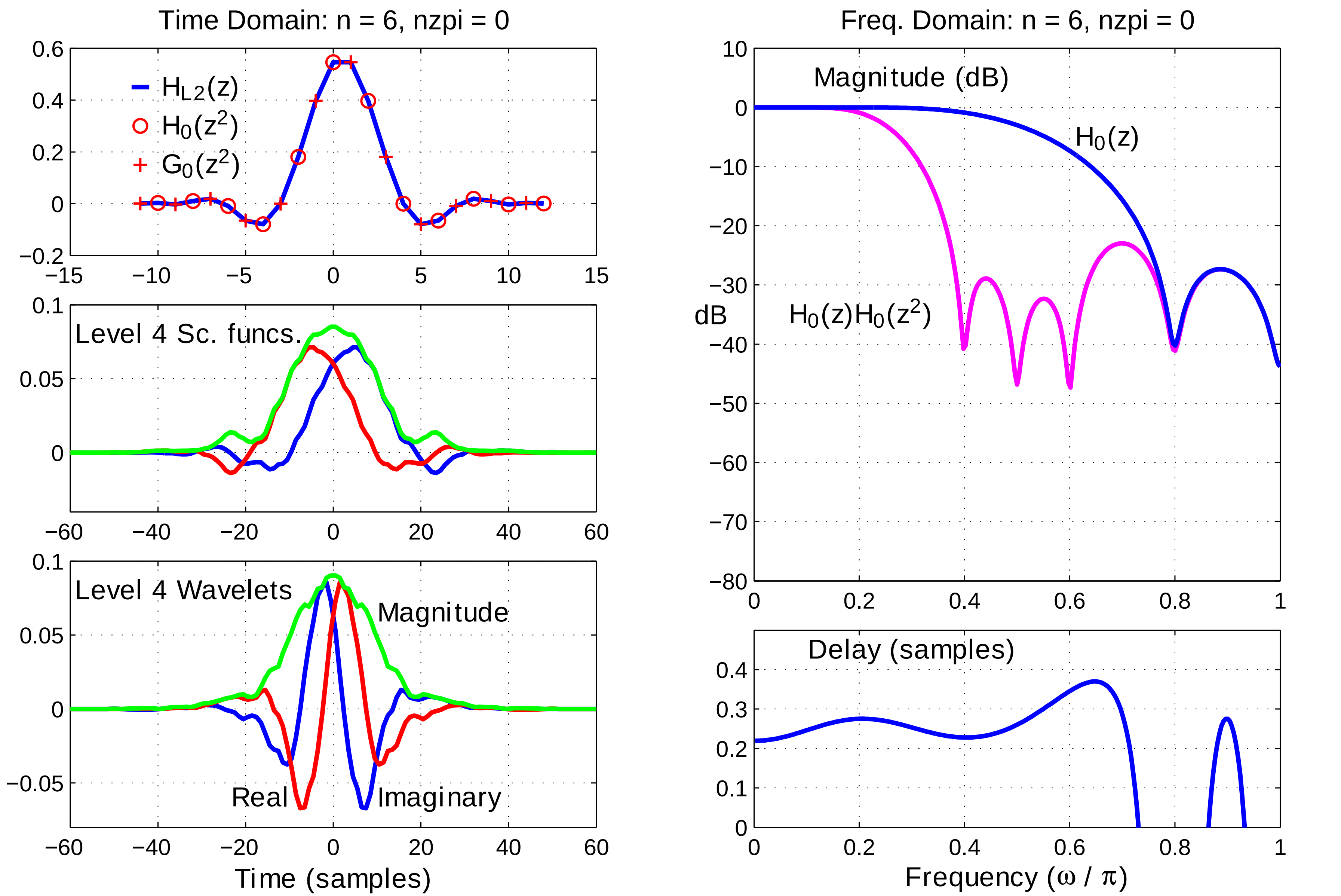


Fig. 8: Q-shift filters for $n = 6$ (12 filter taps) and no predefined zeros.

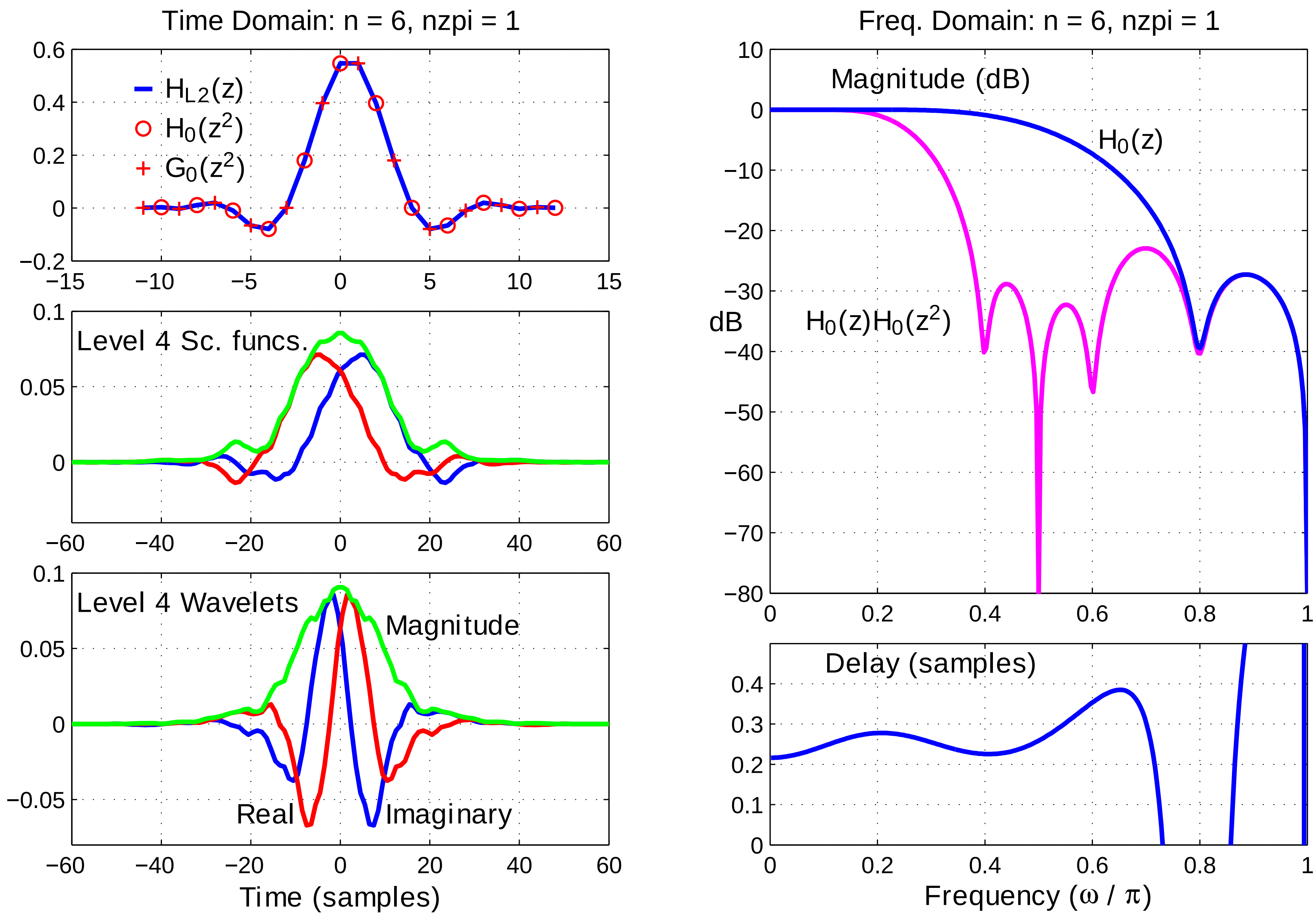


Fig. 9: Q-shift filters for $n = 6$ (12 filter taps) and 1 predefined zero at $\omega = \pi$.

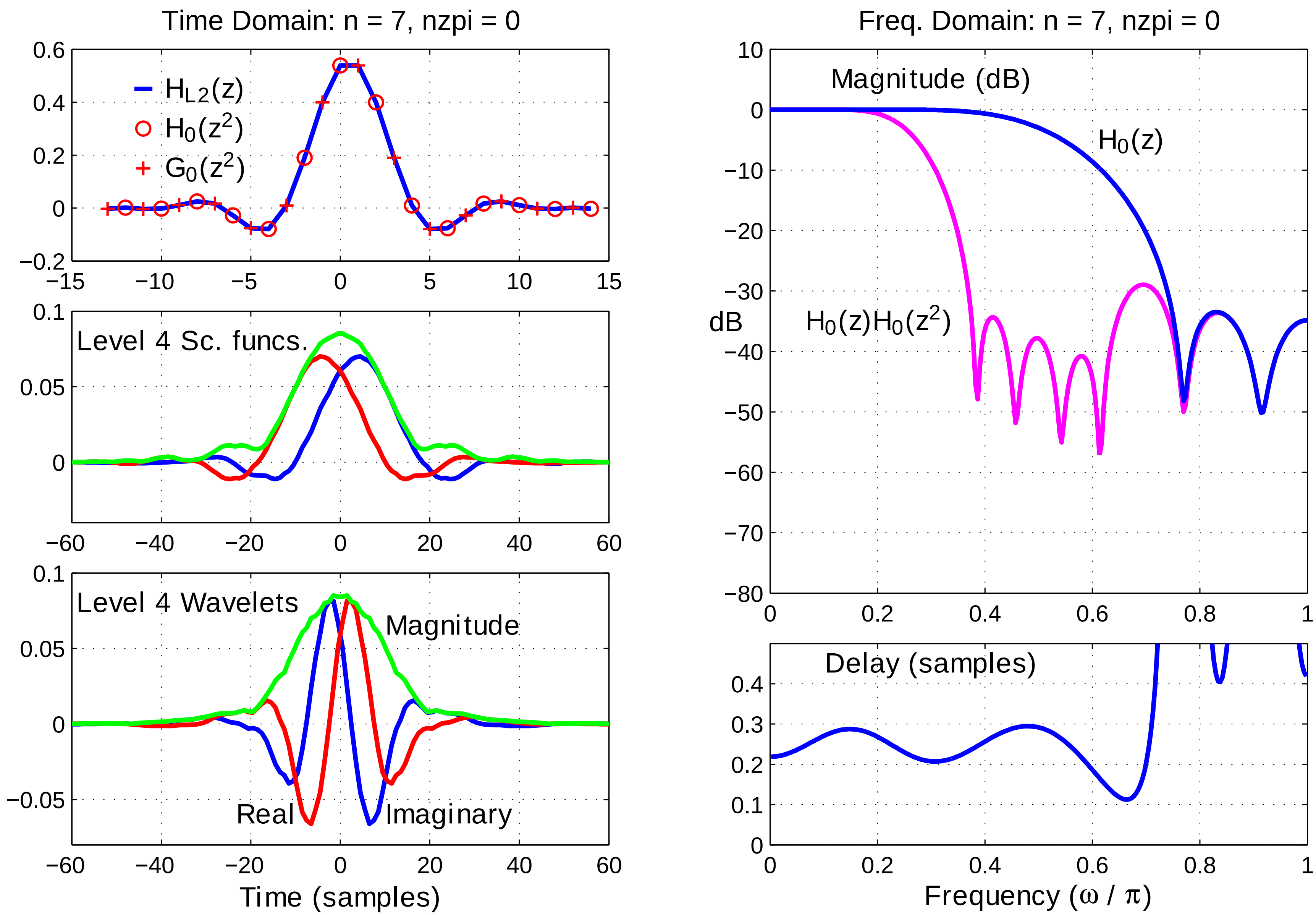


Fig. 10: Q-shift filters for $n = 7$ (14 filter taps) and no predefined zeros.

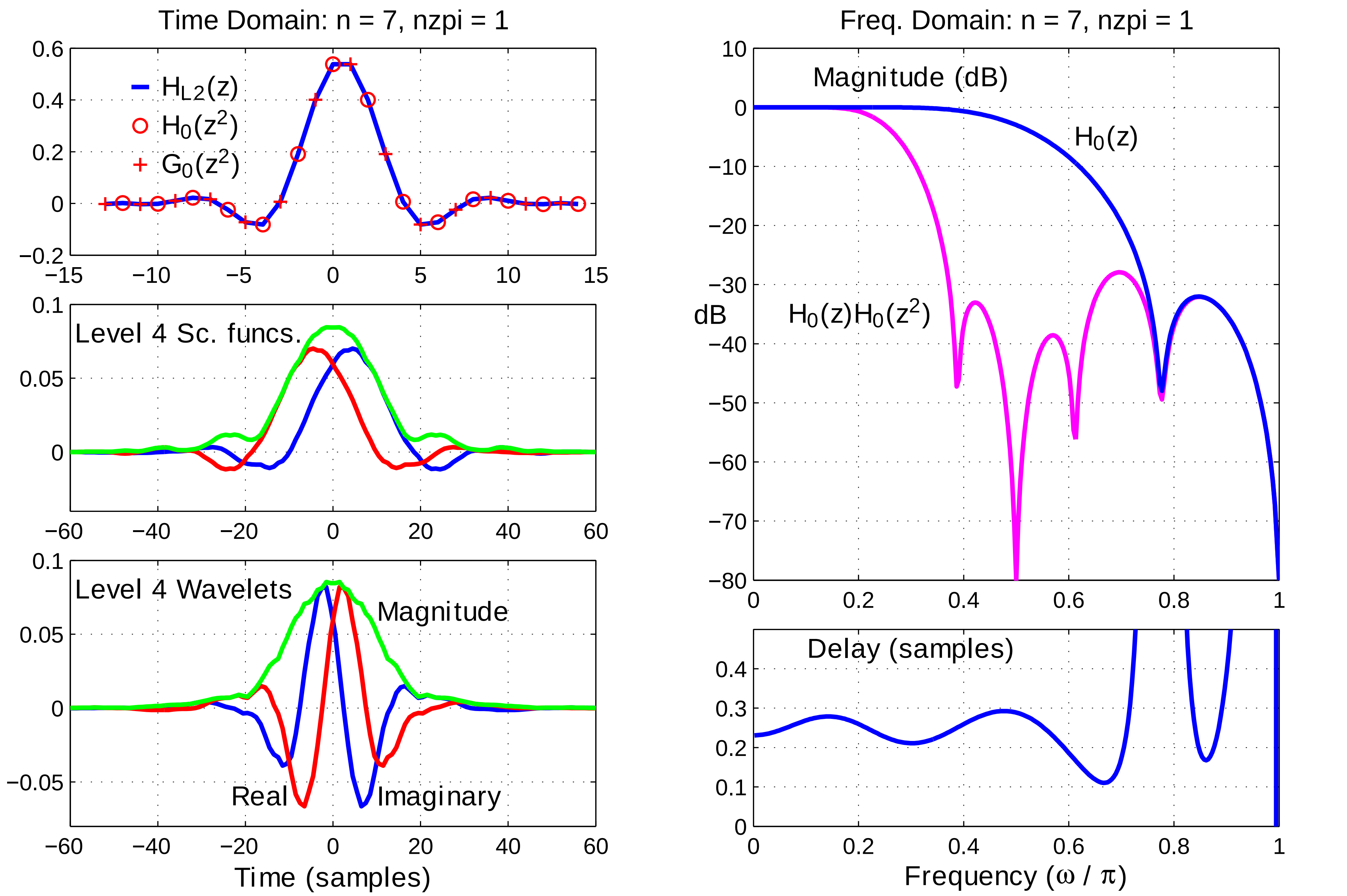


Fig. 11: Q-shift filters for $n = 7$ (14 filter taps) and 1 predefined zero at $\omega = \pi$.

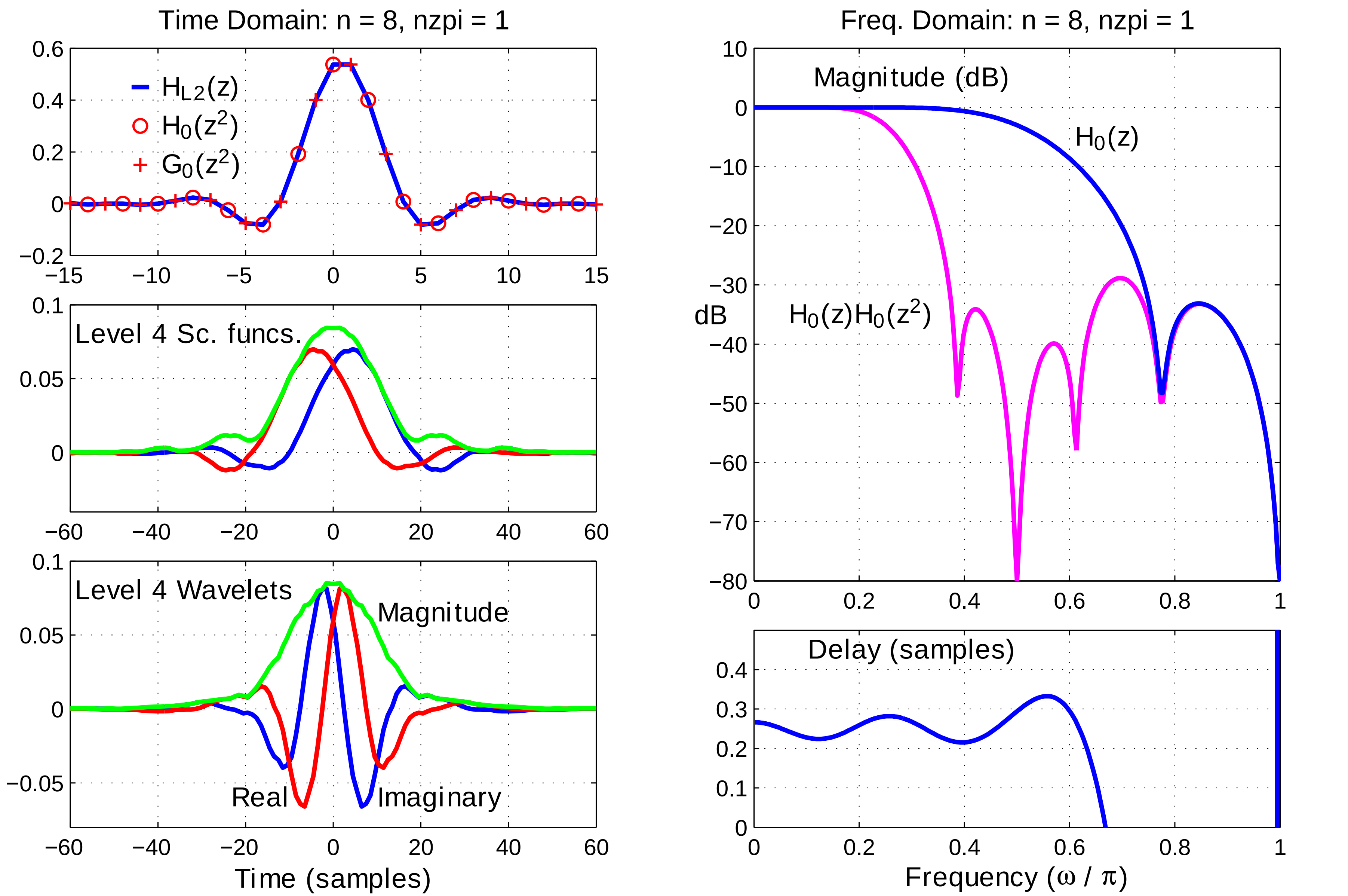


Fig. 12: Q-shift filters for $n = 8$ (16 filter taps) and 1 predefined zero at $\omega = \pi$.

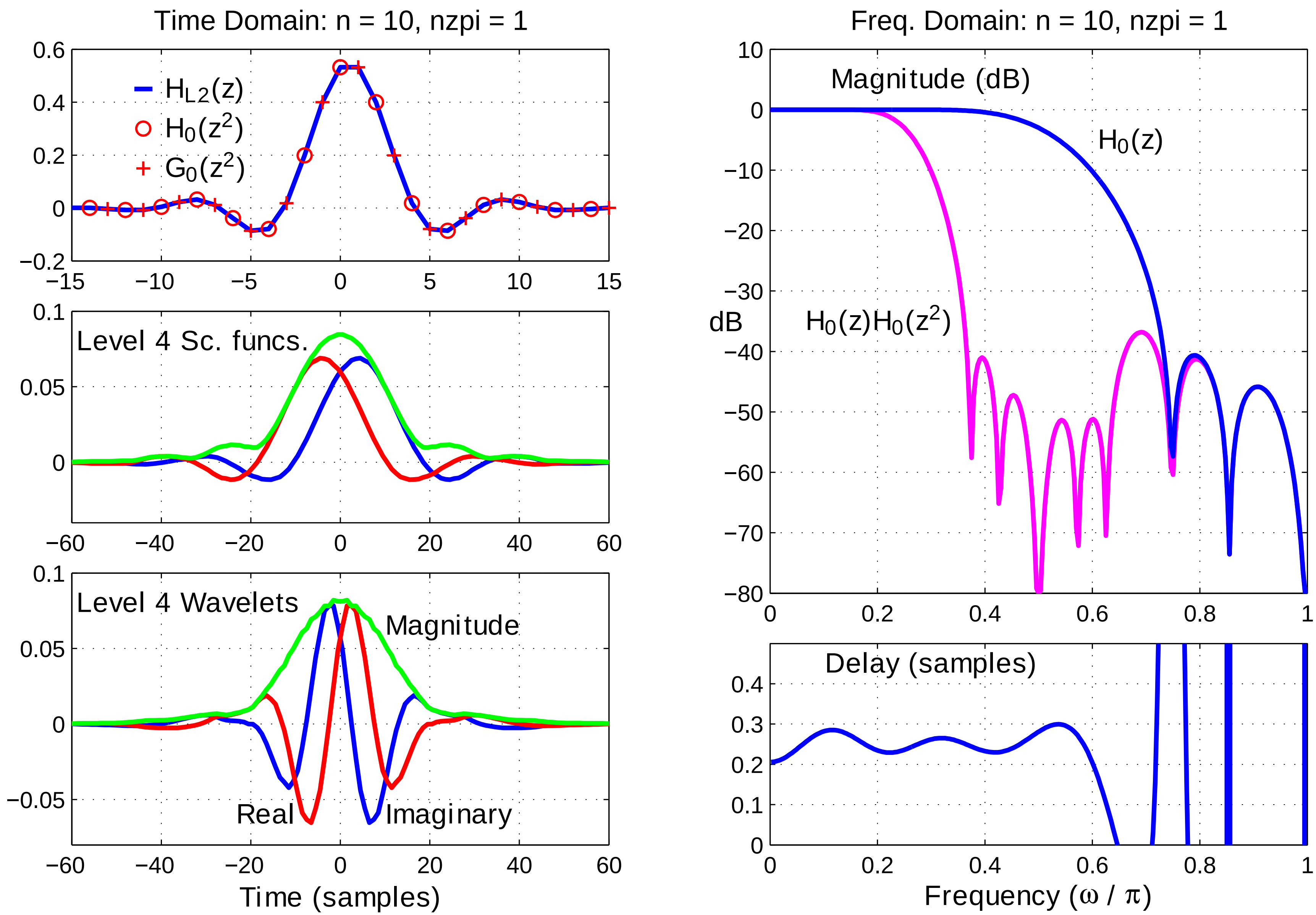


Fig. 13: Q-shift filters for $n = 10$ (20 filter taps) and 1 predefined zero at $\omega = \pi$.

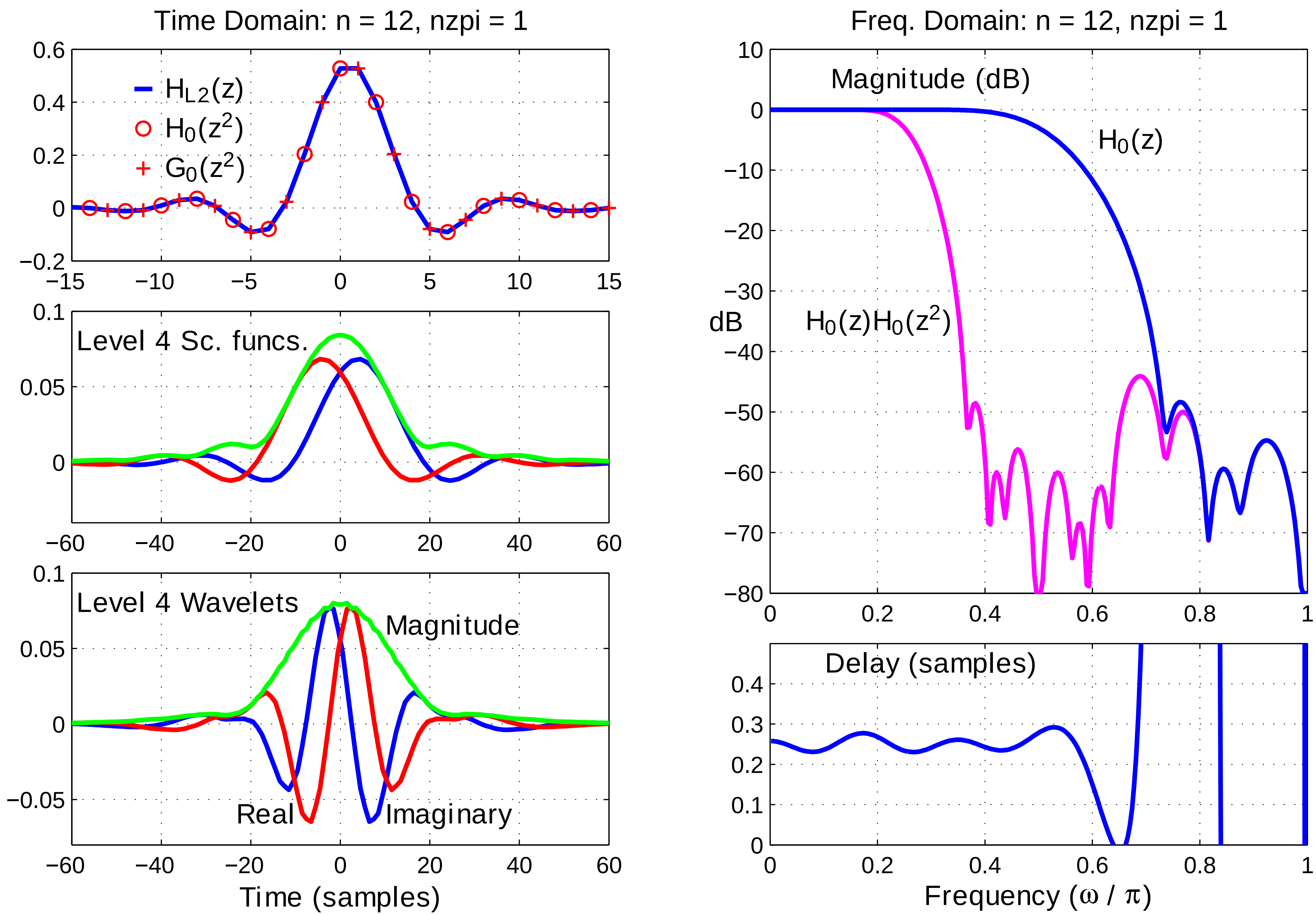


Fig. 14: Q-shift filters for $n = 12$ (24 filter taps) and 1 predefined zero at $\omega = \pi$.

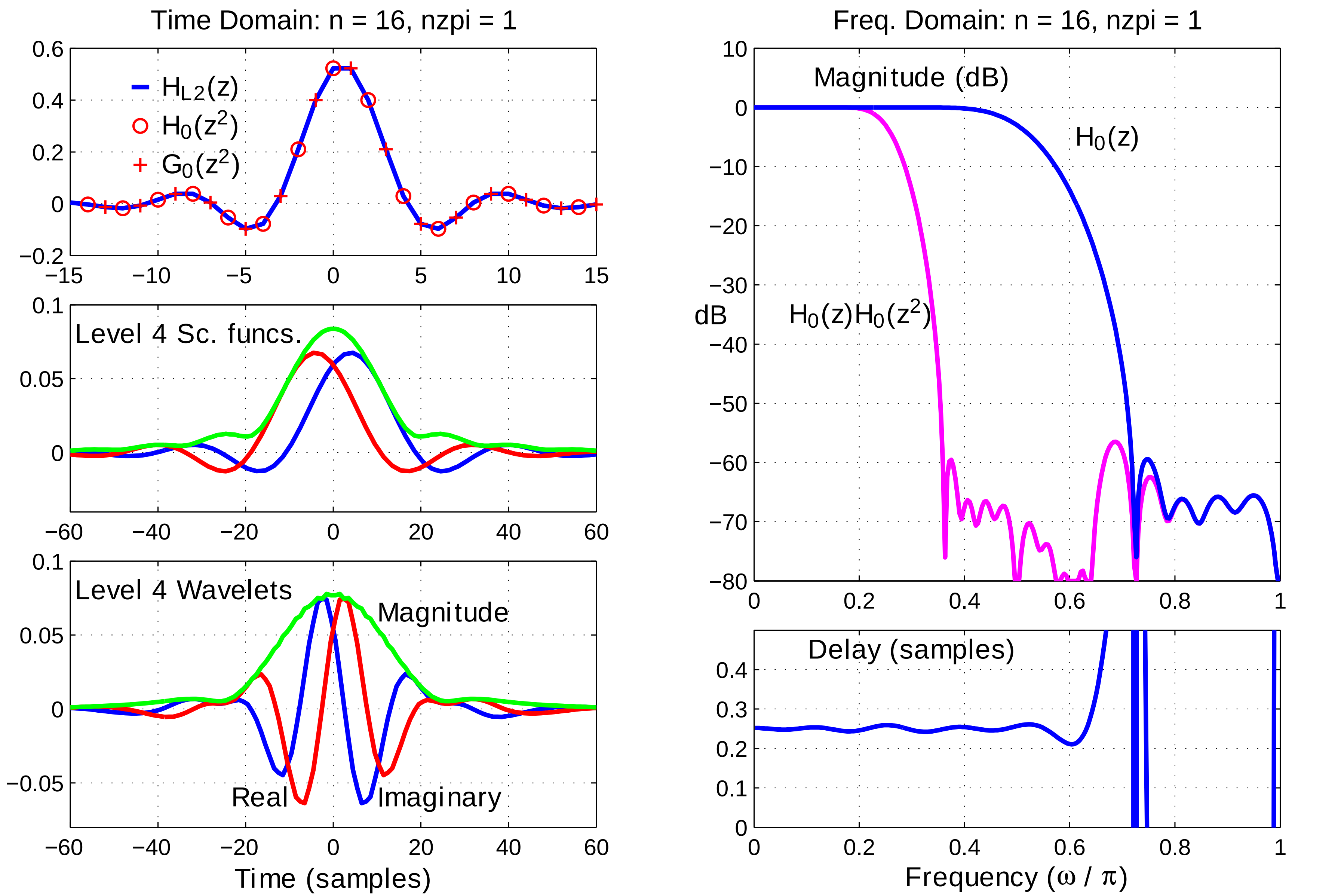


Fig. 15: Q-shift filters for $n = 16$ (32 filter taps) and 1 predefined zero at $\omega = \pi$.

Filter Design – Conclusions

- The proposed algorithm gives a fast and effective way of designing Q-shift filters for the DT CWT.
- All filters produce perfect reconstruction, tight frames and linear-phase complex wavelets.
- As the length of the filters $(2n)$ increases, the design method gives improvements in stopband attenuation, constancy of group delay, and smoothness in the resulting wavelet bases. Hence we get increasing accuracy of shift-invariance.
- The algorithm works well for filter lengths from 10 to over 50 taps.
- Matlab code for the algorithm and papers on the DT CWT can be downloaded from the author's website, **http://www-sigproc.eng.cam.ac.uk/~ngk/**.
- Matlab code to implement the DT CWT is free for researchers and available by emailing the author at **ngk@eng.cam.ac.uk** .

Visualising Shift Invariance

- Apply a standard input (e.g. unit step) to the transform for a **range of shift positions**.
- Select the transform coefficients from **just one wavelet level** at a time.
- Inverse transform each set of selected coefficients.
- Plot the component of the reconstructed output for each shift position at each wavelet level.
- Check for **shift invariance** (similarity of waveforms).

Fig 4 shows that the DT CWT has near-perfect shift invariance, whereas the maximally-decimated real discrete wavelet transform (DWT) has substantial shift dependence.

Shift Invariance of DT CWT vs DWT

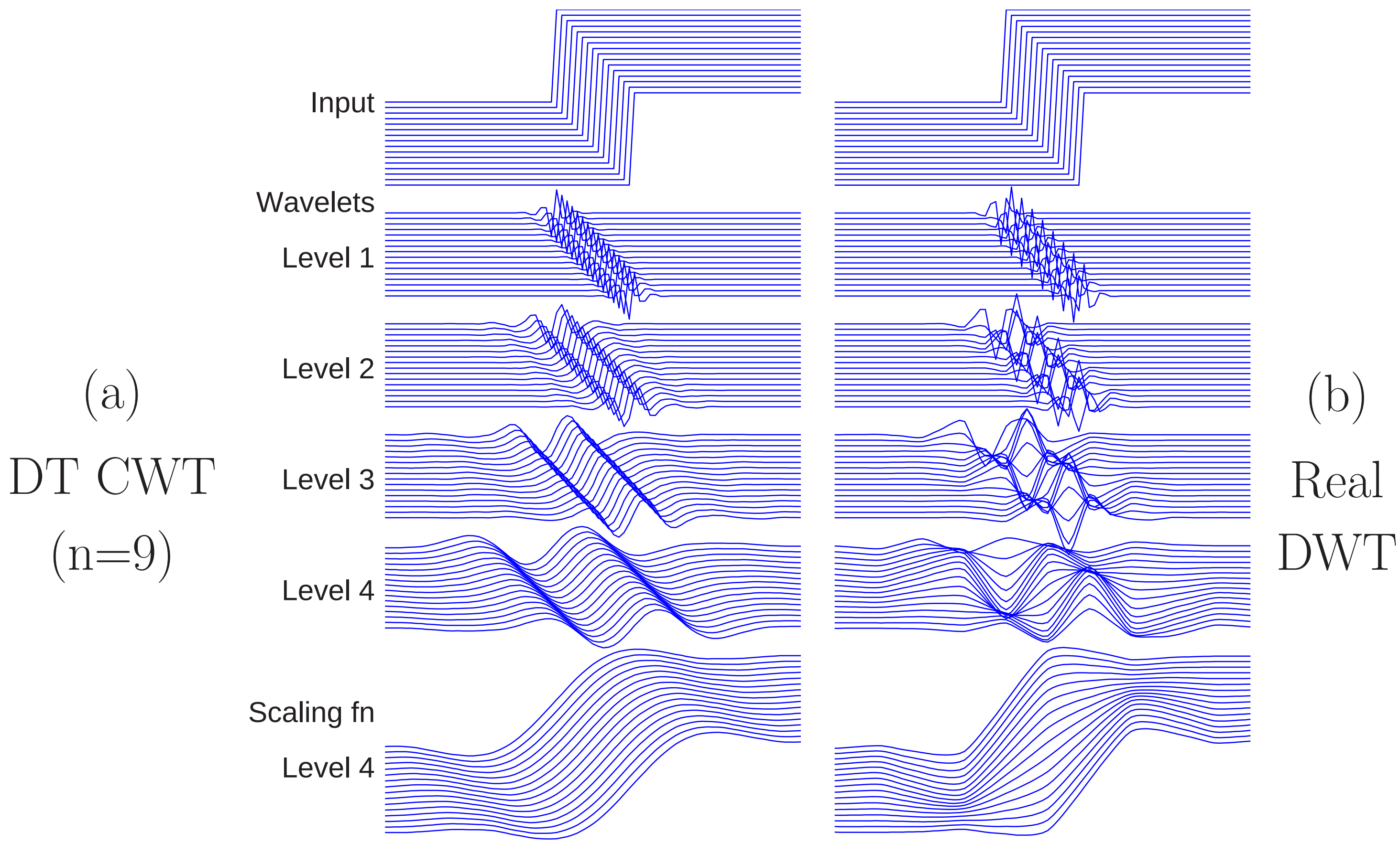


Figure 4: Wavelet and scaling function components at levels 1 to 4 of 16 shifted step responses of the DT CWT (a) and real DWT (b). If there is good shift invariance, all components at a given level should be similar in shape, as in (a).

Shift Invariance of simpler DT CWTs

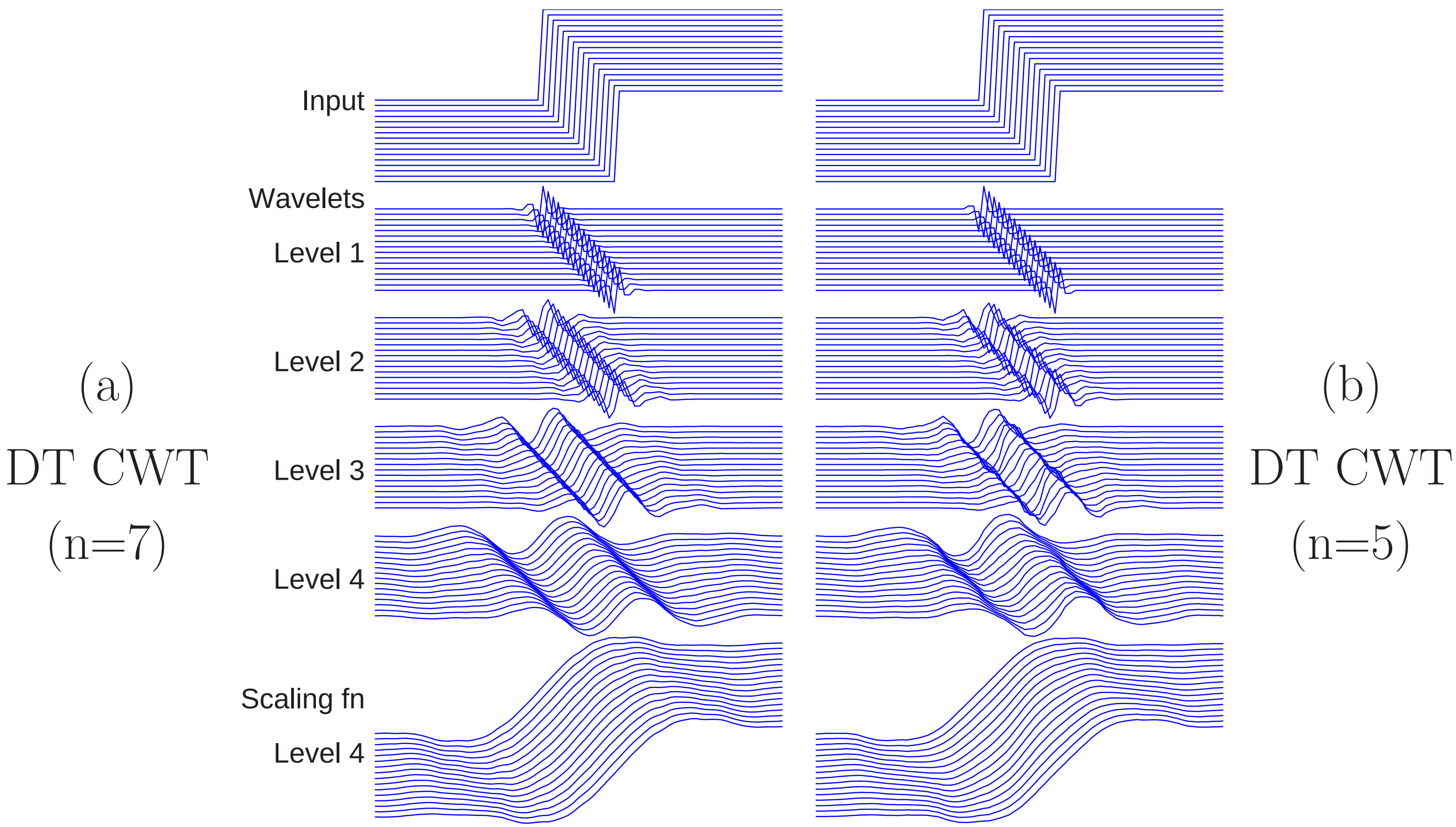
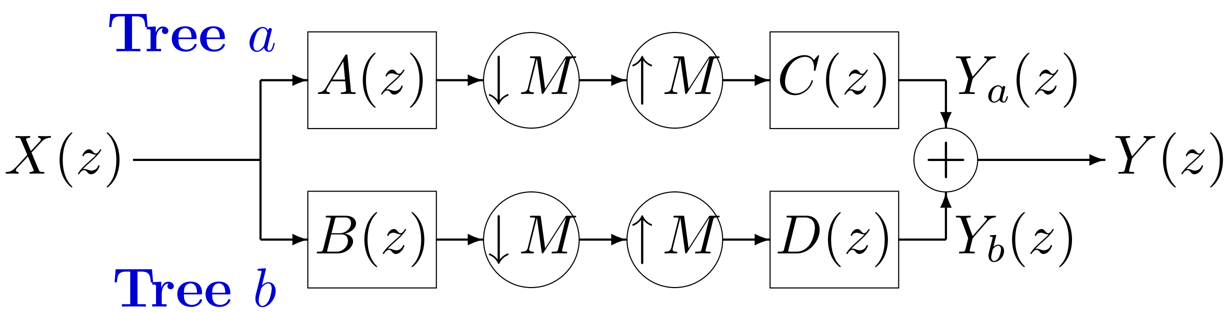


Figure 5: Wavelet and scaling function components at levels 1 to 4 of 16 shifted step responses of simpler forms of the DT CWT, using (a) 14-tap and (b) 6-tap Q-shift filters with n = 7 and 5 respectively.

Shift Invariance – Quantitative measurement



Basic configuration of the dual tree if either wavelet or scaling-function coefficients from just level m are retained ($M = 2^m$).

Letting $W = e^{j2\pi/M}$, **multi-rate analysis** gives:

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X(W^k z)[A(W^k z)\, C(z) + B(W^k z)\, D(z)]$$

For shift invariance, **aliasing terms ($k \neq 0$) must be negligible.** So we design $B(W^k z)\, D(z)$ to cancel $A(W^k z)\, C(z)$ for all non-zero k that give overlap of the passbands of filters $C(z)$ or $D(z)$ with those of shifted filters $A(W^k z)$ or $B(W^k z)$.

A Measure of Shift Invariance

Since

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X(W^k z)[A(W^k z)\, C(z) + B(W^k z)\, D(z)]$$

we quantify the shift dependence of a transform by calculating the ratio of the total energy of the **unwanted aliasing transfer functions** (the terms with $k \neq 0$) to the energy of the **wanted transfer function** (when $k = 0$):

$$R_a = \frac{\sum_{k=1}^{M-1} \mathcal{E}\{A(W^k z)\, C(z) + B(W^k z)\, D(z)\}}{\mathcal{E}\{A(z)\, C(z) + B(z)\, D(z)\}}$$

where $\mathcal{E}\{U(z)\}$ calculates the energy, $\sum_r |u_r|^2$, of the impulse response of a z-transfer function, $U(z) = \sum_r u_r z^{-r}$.

$\mathcal{E}\{U(z)\}$ may also be interpreted in the **frequency domain** as the integral of the squared magnitude of the frequency response, $\frac{1}{2\pi} \int_{-\pi}^{\pi} |U(e^{j\theta})|^2 \, d\theta$ from Parseval's theorem.

Types of DT CWT filters

We show results for the following combinations of filters:

A **(13,19)-tap** and **(12,16)-tap** near-orthogonal odd/even filter sets.

B **(13,19)-tap** near-orthogonal filters at level 1, **18-tap** Q-shift filters at levels ≥ 2.

C **(13,19)-tap** near-orthogonal filters at level 1, **14-tap** Q-shift filters at levels ≥ 2.

D **(9,7)-tap** bi-orthogonal filters at level 1, **18-tap** Q-shift filters at levels ≥ 2.

E **(9,7)-tap** bi-orthogonal filters at level 1, **14-tap** Q-shift filters at levels ≥ 2.

F **(9,7)-tap** bi-orthogonal filters at level 1, **6-tap** Q-shift filters at levels ≥ 2.

G **(5,3)-tap** bi-orthogonal filters at level 1, **6-tap** Q-shift filters at levels ≥ 2.

Aliasing Energy Ratios,

Values of R_a **in dB**, for filter types A to G over levels 1 to 5.

Filters: Complexity:	A 2.0	B 2.3	C 2.0	D 1.9	E 1.6	F 1.0	G 0.7	DWT 1.0
Wavelet								
Level 1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	-9.40
Level 2	-28.25	-31.40	-29.06	-22.96	-21.81	-18.49	-14.11	-3.54
Level 3	-23.62	-27.93	-25.10	-20.32	-18.96	-14.60	-11.00	-3.53
Level 4	-22.96	-31.13	-24.67	-32.08	-24.85	-16.78	-15.80	-3.52
Level 5	-22.81	-31.70	-24.15	-31.88	-24.15	-18.94	-18.77	-3.52
Scaling fn.								
Level 1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	-9.40
Level 2	-29.37	-32.50	-30.17	-24.32	-23.19	-19.88	-15.93	-9.38
Level 3	-28.17	-35.88	-29.21	-36.94	-29.33	-21.75	-20.63	-9.37
Level 4	-27.88	-37.14	-28.57	-37.37	-28.56	-24.37	-24.15	-9.37
Level 5	-27.75	-36.00	-28.57	-36.01	-28.57	-24.67	-24.65	-9.37

The DT CWT in 2-D

When the DT CWT is applied to 2-D signals (images), it has the following features:

- It is performed separably, with 2 trees used for the rows of the image and 2 trees for the columns – yielding a **Quad-Tree** structure (4:1 redundancy).

- The 4 quad-tree components of each coefficient are combined by simple sum and difference operations to yield a **pair of complex coefficients**. These are part of two separate subbands in adjacent quadrants of the 2-D spectrum.

- This produces **6 directionally selective subbands** at each level of the 2-D DT CWT. Fig 6 shows the basis functions of these subbands at level 4, and compares them with the 3 subbands of a 2-D DWT.

- The DT CWT is directionally selective (see fig 9) because the complex filters can **separate positive and negative frequency components** in 1-D, and hence **separate adjacent quadrants** of the 2-D spectrum. Real separable filters cannot do this!

2-D Basis Functions at level 4

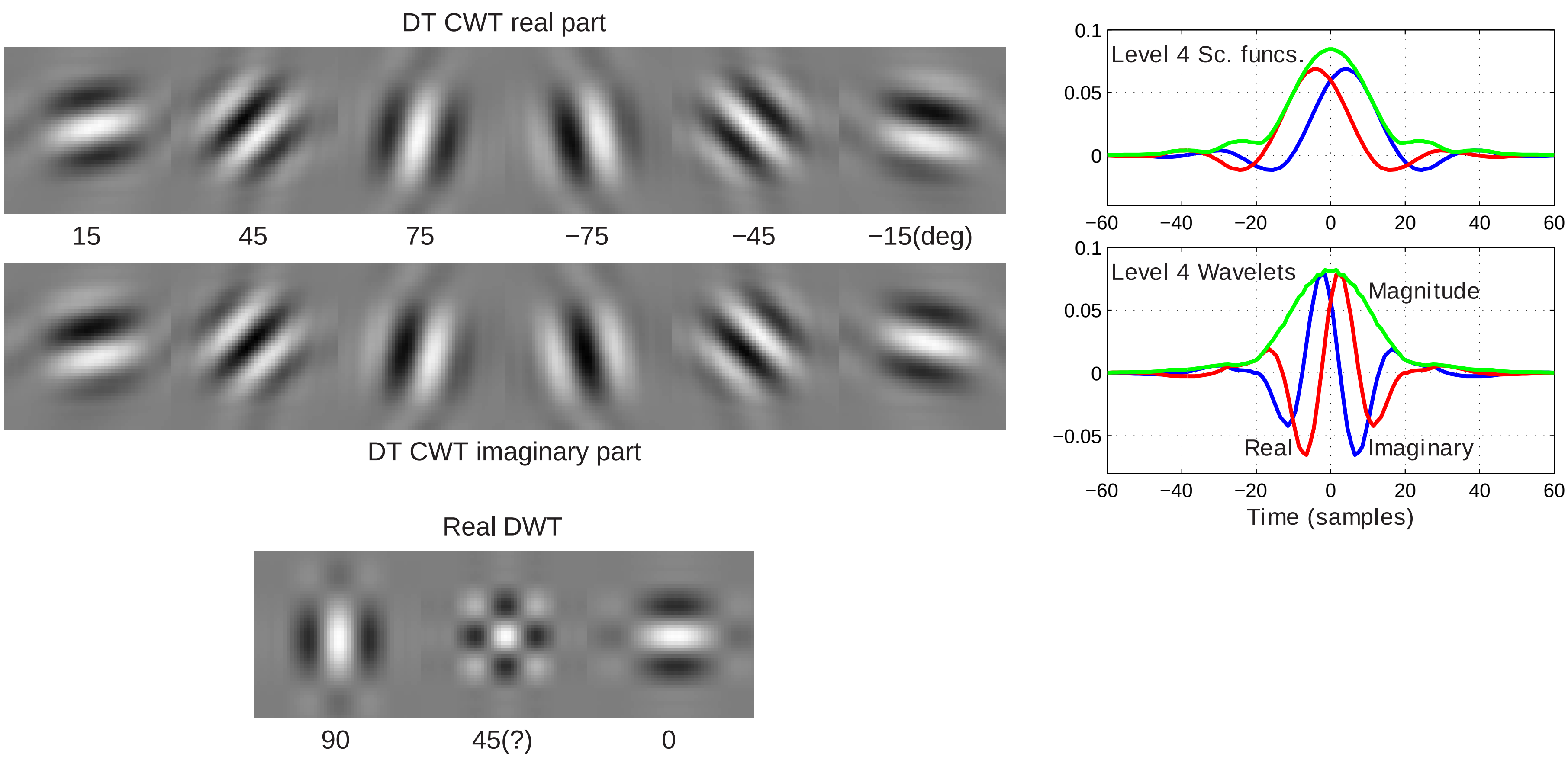


Figure 6: Basis functions of 2-D Q-shift complex wavelets (top), and of 2-D real wavelet filters (bottom), all illustrated at level 4 of the transforms. The complex wavelets provide 6 directionally selective filters, while real wavelets provide 3 filters, only two of which have a dominant direction. The 1-D bases, from which the 2-D complex bases are derived, are shown to the right.

2 levels of DT CWT in 2-D

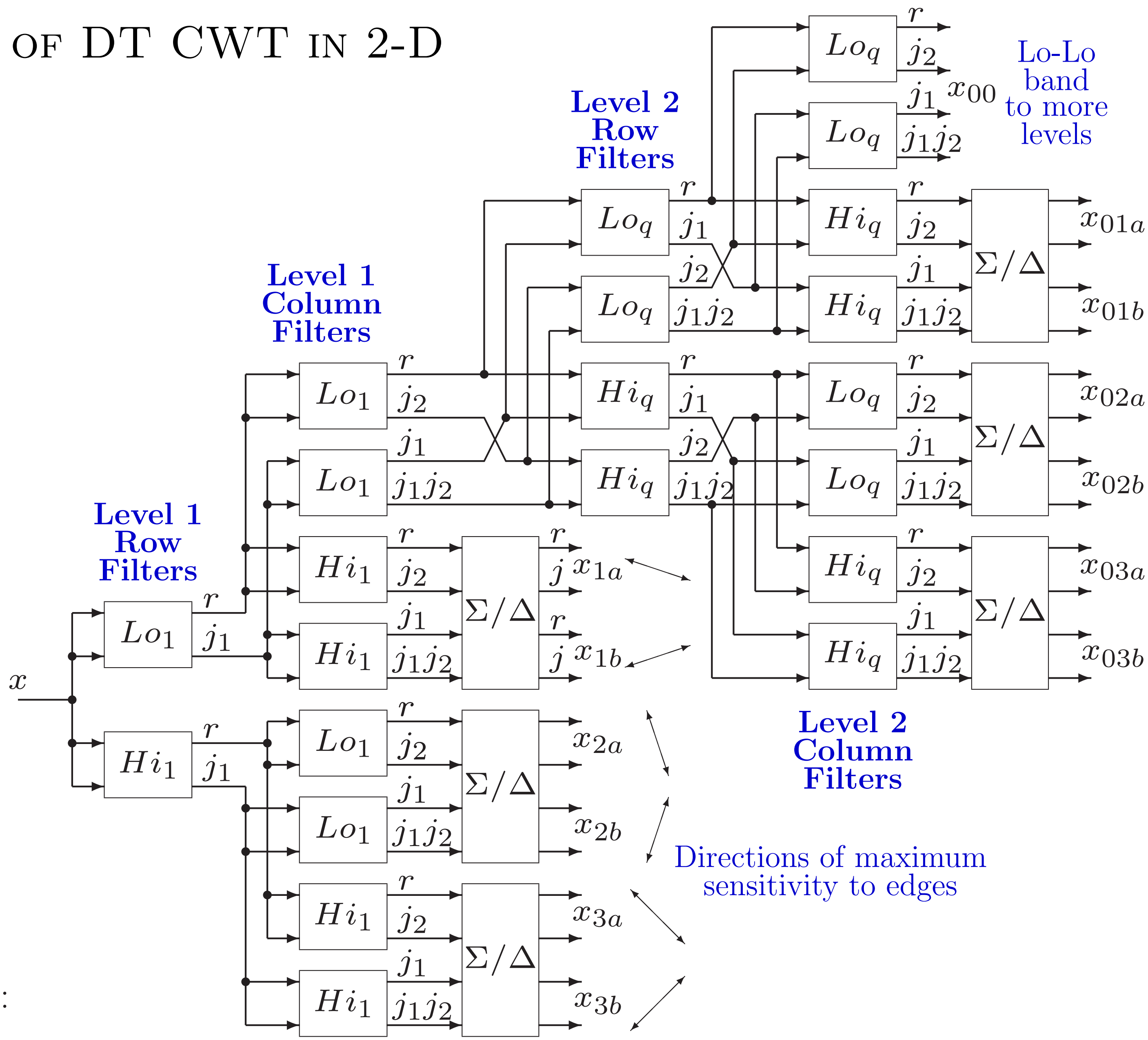
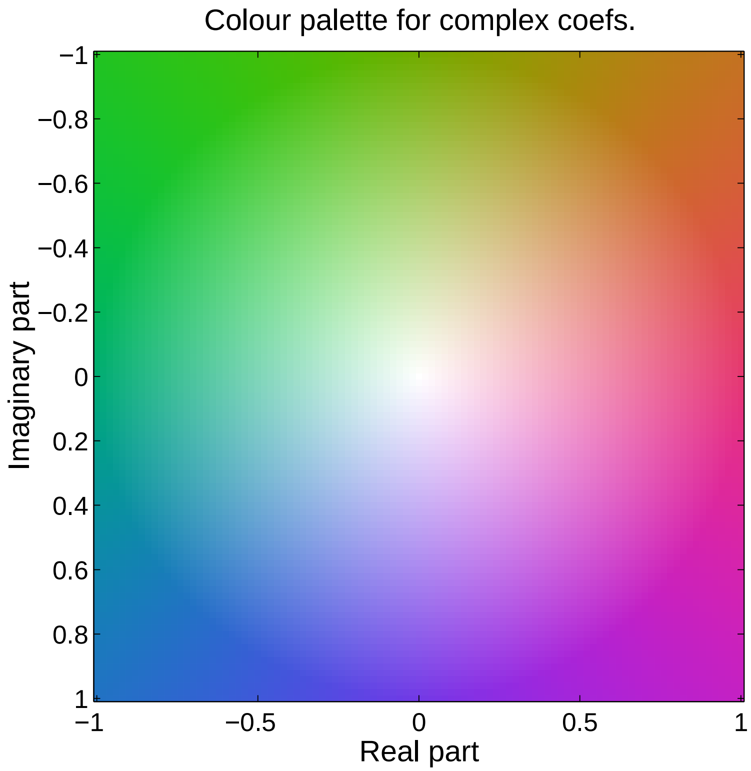


Figure 7:

Test Image and Colour Palette for Complex Coefficients



2-D DT-CWT Decomposition into Subbands

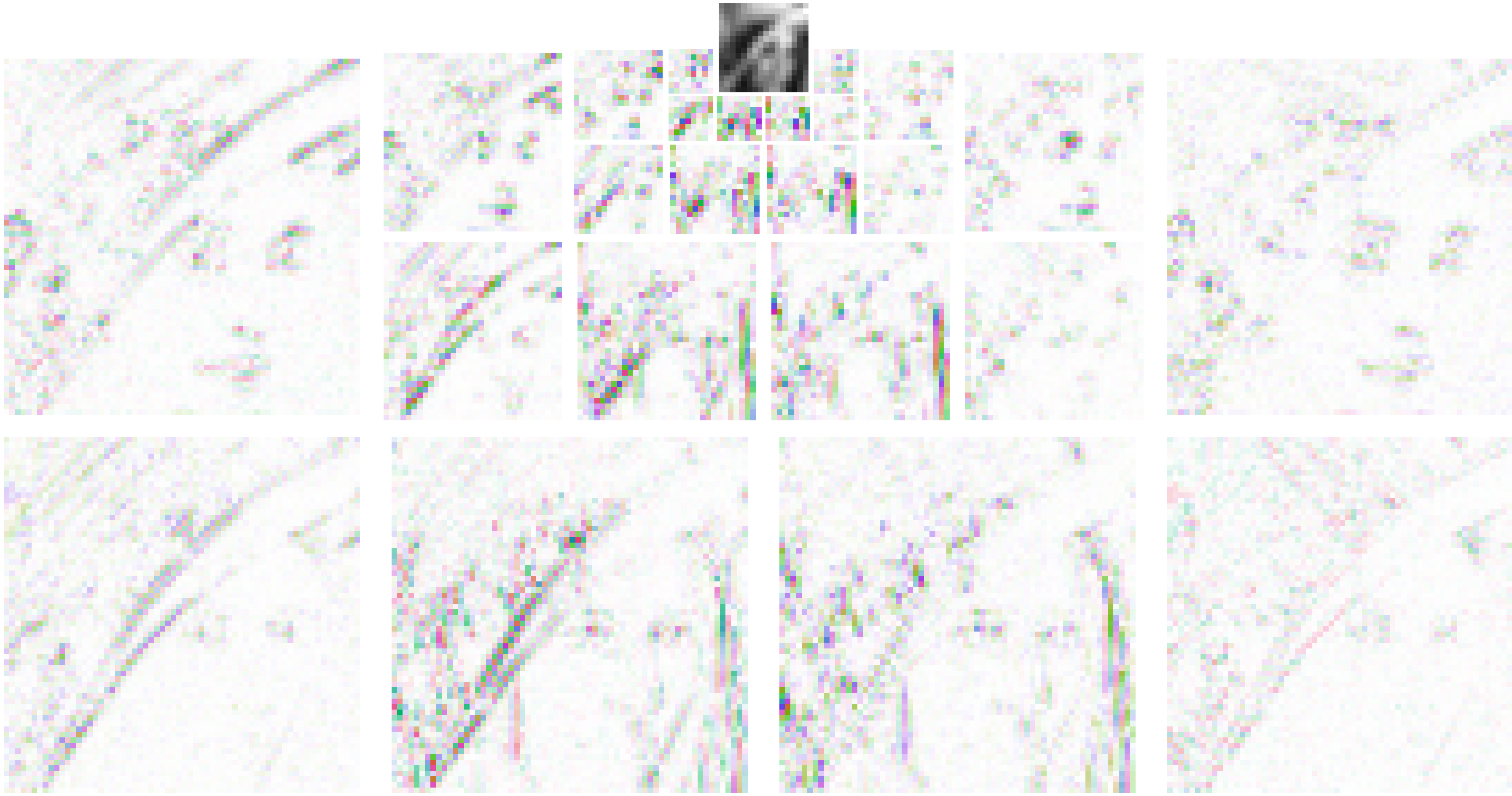


Figure 8: Four-level DT-CWT decomposition of *Lenna* into 6 subbands per level (only the central 128×128 portion of the image is shown for clarity). A colour-wheel palette is used to display the complex wavelet coefficients.

2-D DT-CWT Reconstruction Components from Each Subband

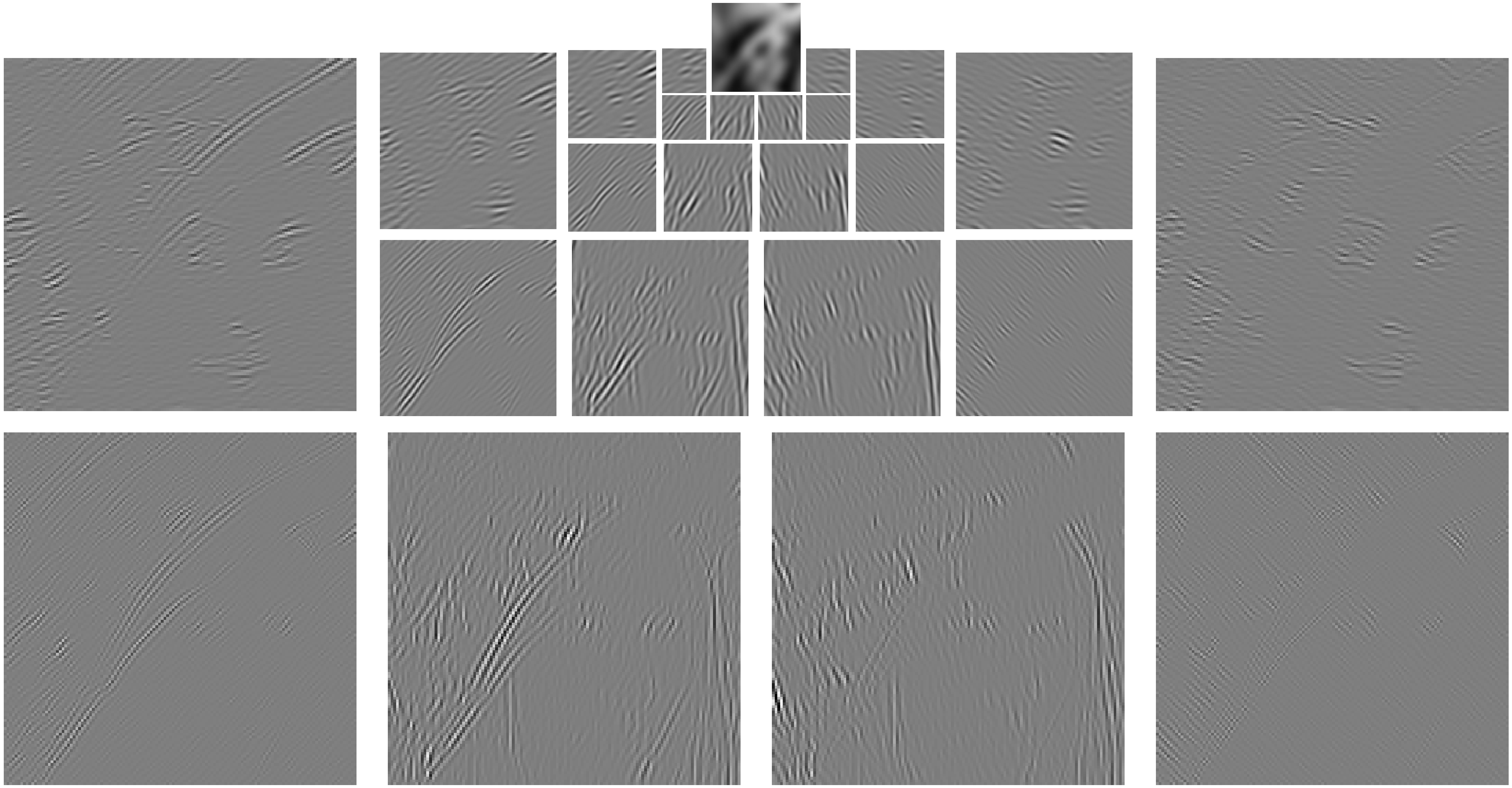


Figure 9: Components from each subband of the reconstructed output image for a 4-level DT-CWT decomposition of *Lenna* (central 128×128 portion only).

2-D Shift Invariance of DT CWT vs DWT

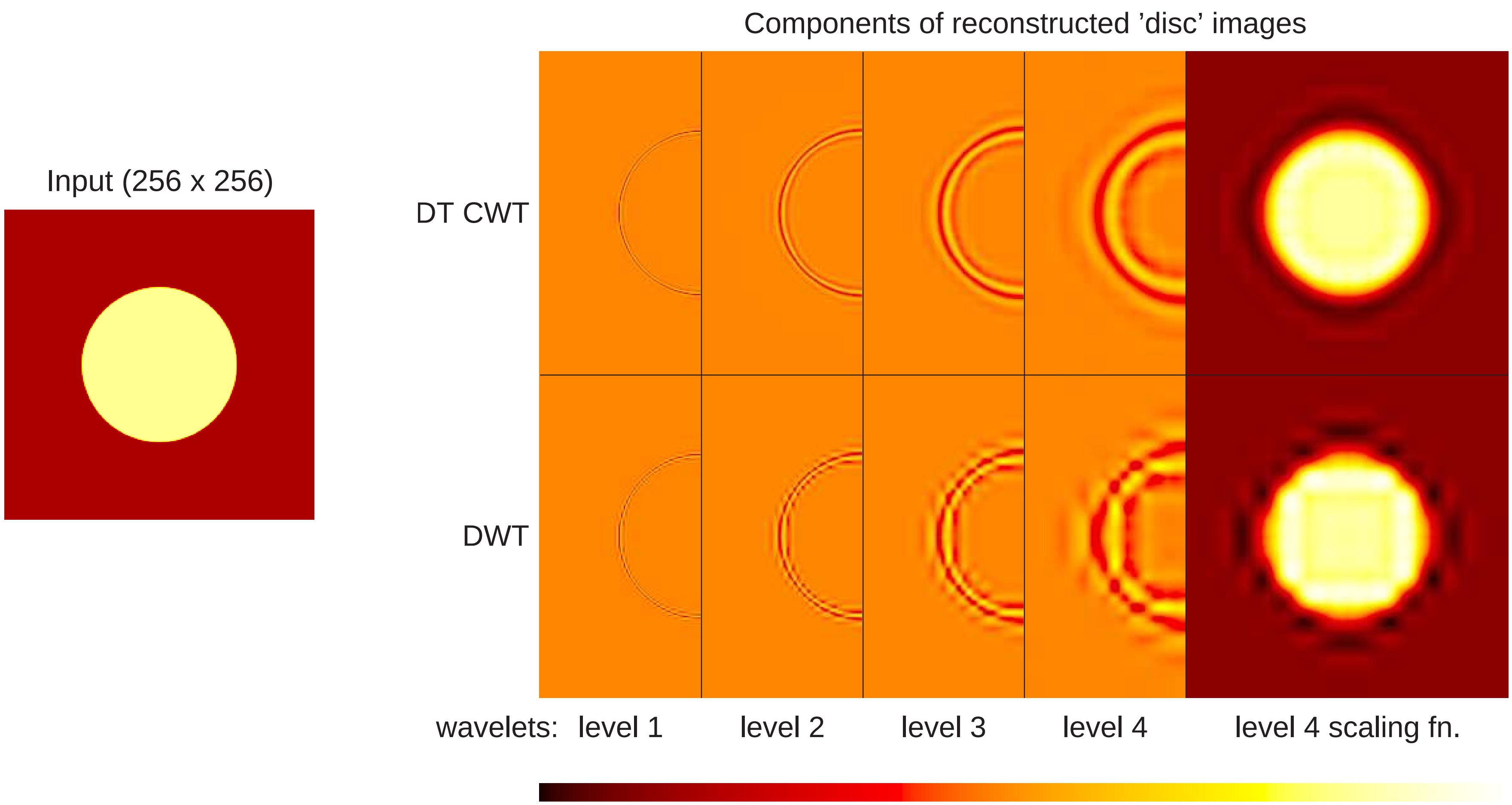


Figure 10: Wavelet and scaling function components at levels 1 to 4 of an image of a light circular disc on a dark background, using the 2-D DT CWT (upper row) and 2-D DWT (lower row). Only half of each wavelet image is shown in order to save space.

Applications

The Q-shift DT CWT provides a valuable analysis and reconstruction tool for a variety of application areas:

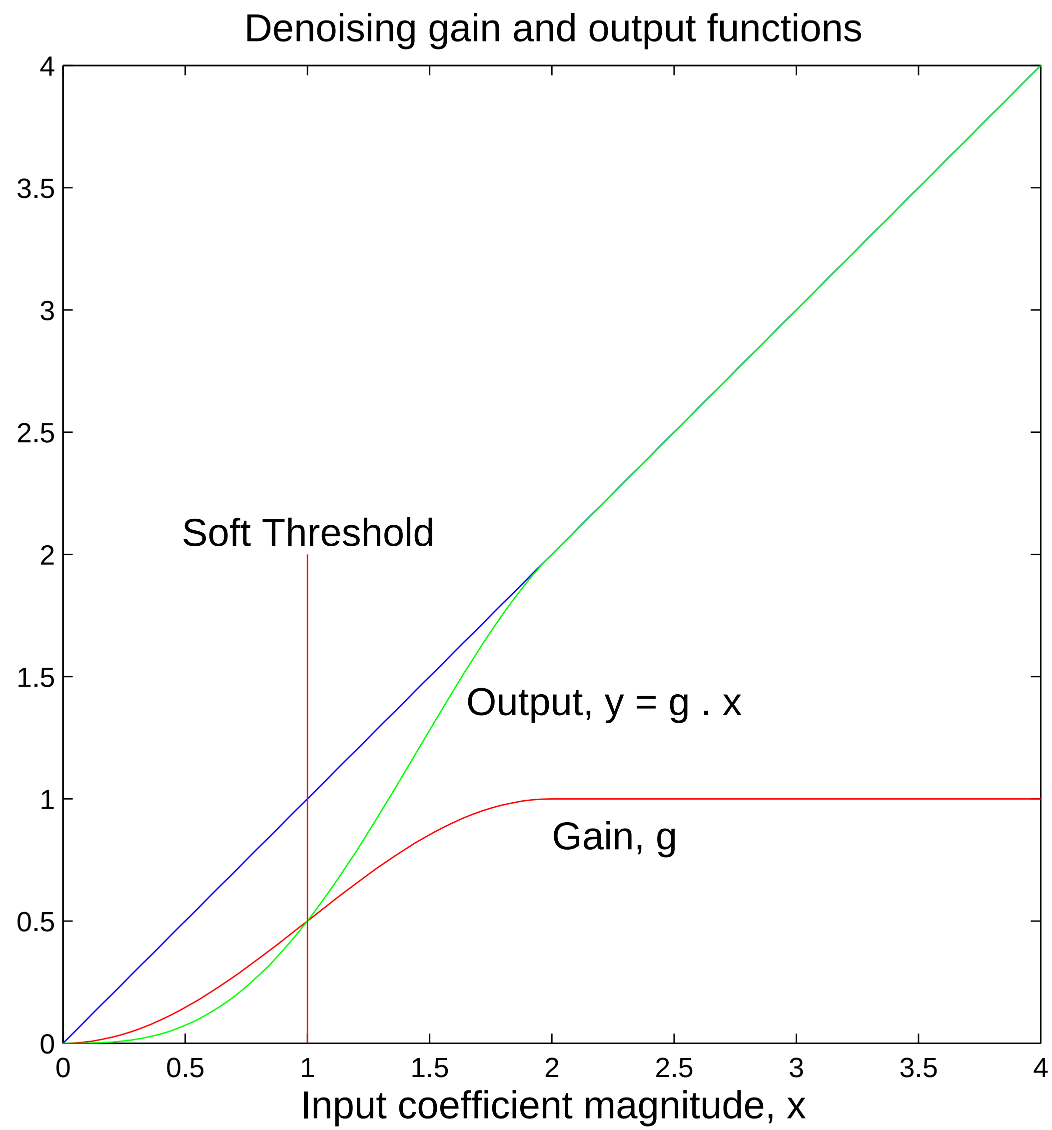
- **Motion estimation** [Magarey 98] and **compensation**
- **Registration** [Kingsbury 02]
- **Denoising** [Choi 00] and **deconvolution** [Jalobeanu 00, De Rivaz 01]
- **Texture analysis** [Hatipoglu 99] and **synthesis** [De Rivaz 00]
- **Segmentation** [De Rivaz 00, Shaffrey 02]
- **Classification** [Romberg 00] and **image retrieval** [Kam & Ng 00, Shaffrey 03]
- **Watermarking of images** [Loo 00] and **video** [Earl 03]
- **Compression / Coding** [Reeves 03]
- **Seismic analysis** [van Spaendonck & Fernandes 02]
- **Diffusion Tensor MRI visualisation** [Zymnis 04]

De-Noising – Method:

- Transform the noisy input image to **compress the image energy** into as few coefs as possible, leaving the noise well distributed.
- Suppress lower energy coefs (mainly noise).
- Inverse transform to recover de-noised image.

What is the Optimum Transform ?

- **DWT** is better than **DCT** or **DFT** for compressing image energy.
- But DWT is **shift dependent** – Is a coef small because there is no signal energy at that scale and location, **or** because it is sampled near a zero-crossing in the wavelet response?
- The **undecimated DWT** can solve this problem but at **significant cost** – redundancy (and computation) is increased by $3M:1$, where M is no. of DWT levels.
- The **DT CWT** has only $4:1$ redundancy, is directionally selective, and works well.


Denoising gain and output functions
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
Soft Threshold
Output, y = g . x
Gain, g
Input coefficient magnitude, x

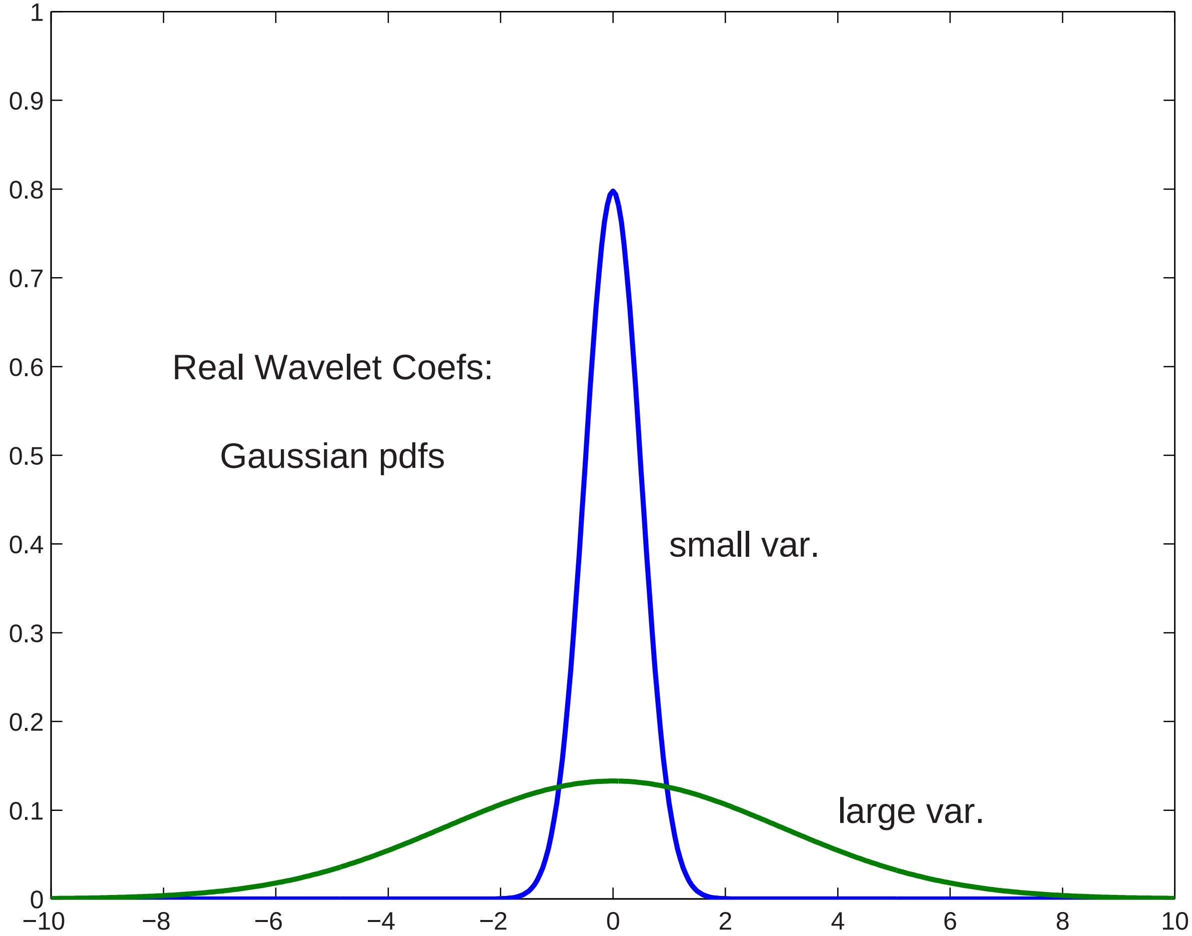


Figure 11: Probability density functions (pdfs) of small and large variance Gaussian distributions, typical for modelling **real and imaginary parts** of complex wavelet coefficients.

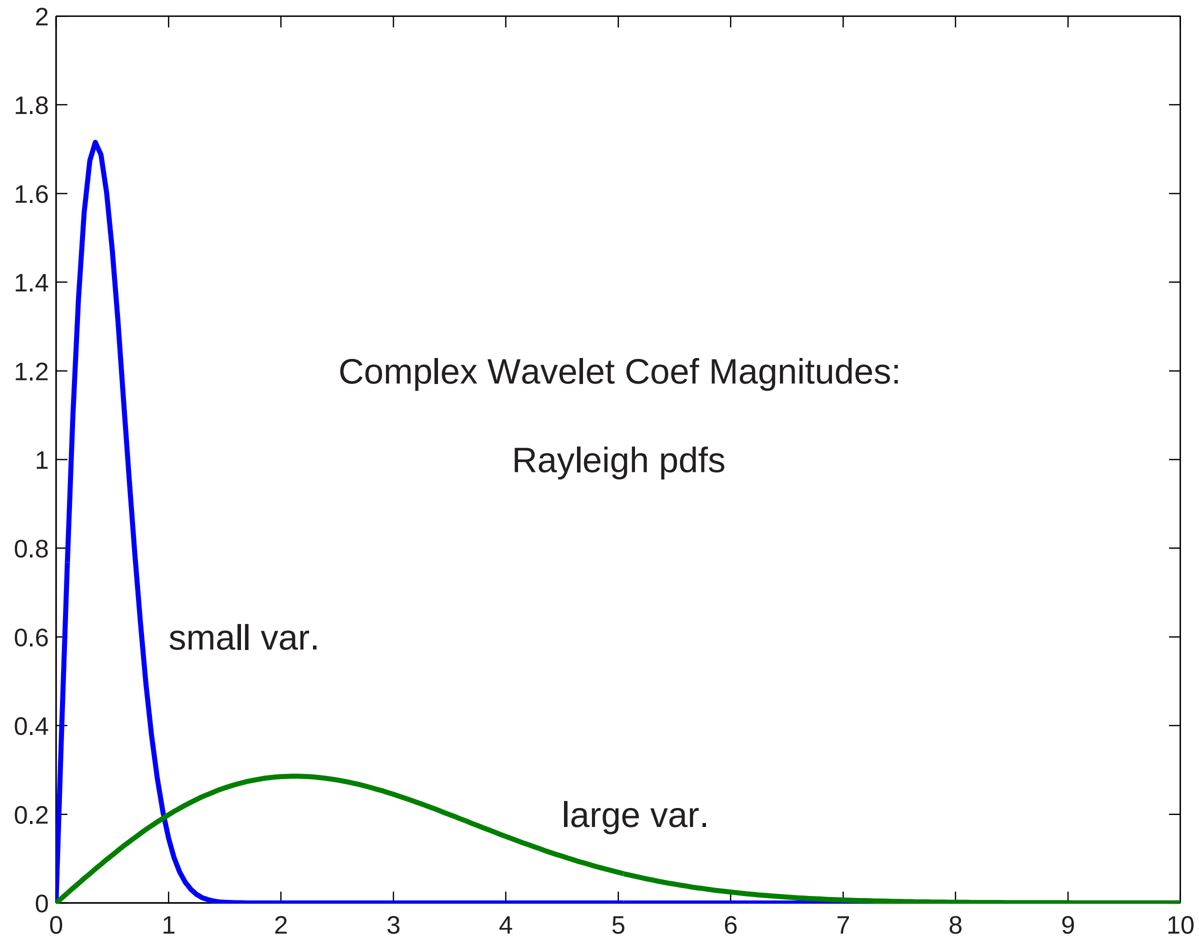


Figure 12: Probability density functions (pdfs) of small and large variance Rayleigh distributions, typical for modelling **magnitudes** of complex wavelet coefficients.

Image Denoising with different Wavelet Transforms - Lenna

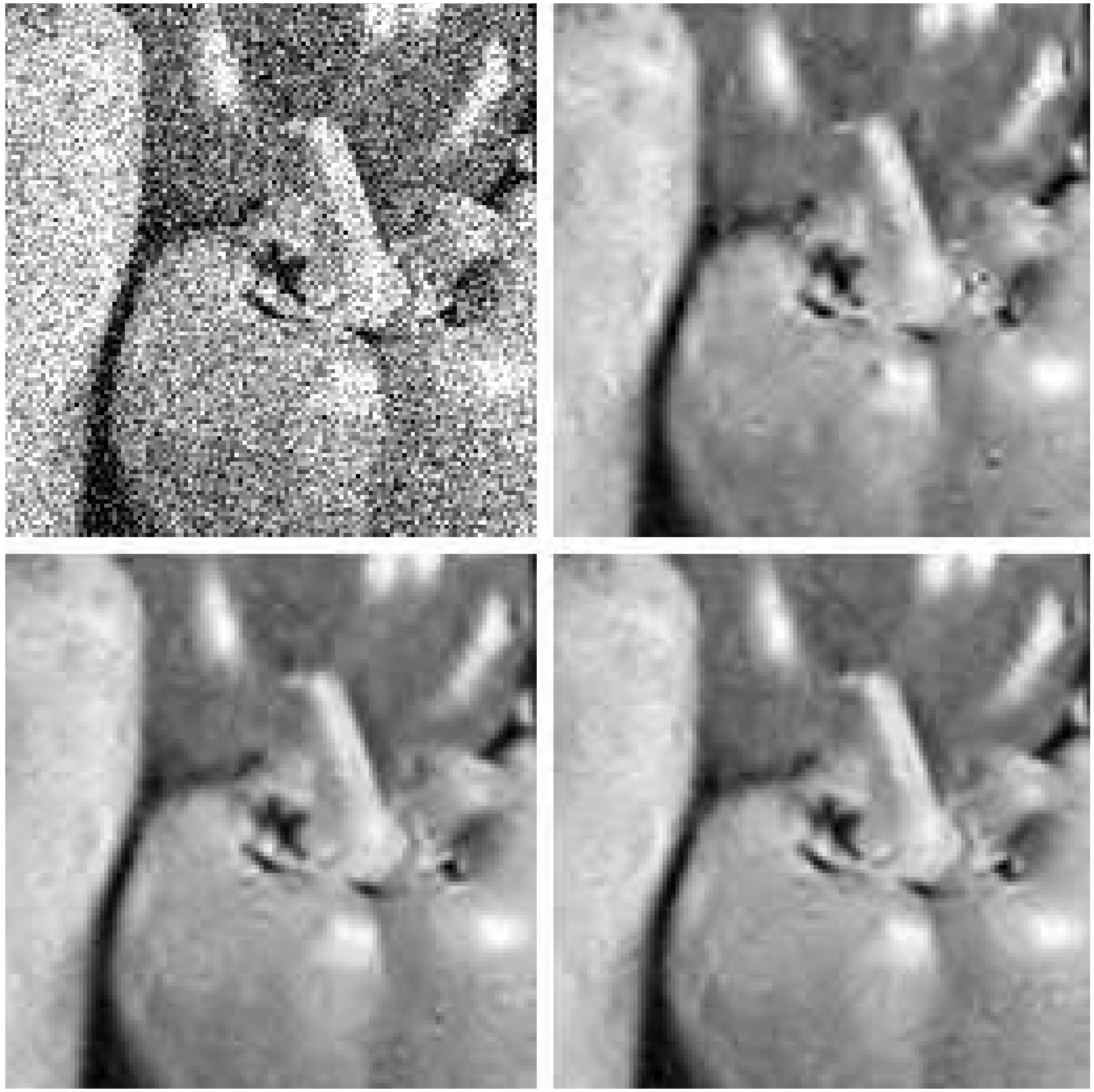
AWGN
SNR =
3.0 dB

Real DWT
SNR =
11.67 dB

Undec. WT
SNR =
12.82 dB

DT CWT
SNR =
12.99 dB

Image Denoising with different Wavelet Transforms - Peppers



AWGN
SNR =
3.0 dB

Real DWT
SNR =
12.24 dB

Undec. WT
SNR =
13.45 dB

DT CWT
SNR =
13.51 dB

Heirarchical Denoising with Gaussian Scale Mixtures (GSMs)

Non-heir.
DT CWT
SNR =
12.99 dB

Heirarchical
DT CWT
SNR =
13.51 dB

Non-heir.
DT CWT
SNR =
13.51 dB

Heirarchical
DT CWT
SNR =
13.85 dB

Coding with the DT CWT

- DT CWT is $4:1$ **redundant** – Why use it for compression?

Because:

- Overcomplete dictionaries of basis functions are known to provide the **potential for better coding** (e.g. Matching Pursuits).

- The 4 reconstruction trees **average** the quantisation noise.

- Reconstruction is a **projection** from $4N$-space to N-space. Noise components, which are not in the N-dimensional range space of the transform, are in the $3N$-dimensional null space and **do not affect the decoded image**.

- Complex wavelet coefficients can define edge locations more accurately than real coefficients.

How to achieve sparsity ?

Basic Algorithm – motivated by Matching Pursuits:

1. Set $i = 1$ and take the DT CWT of the input image.

2. Set to zero all wavelet coefs with magnitude smaller than a threshold θ_i.

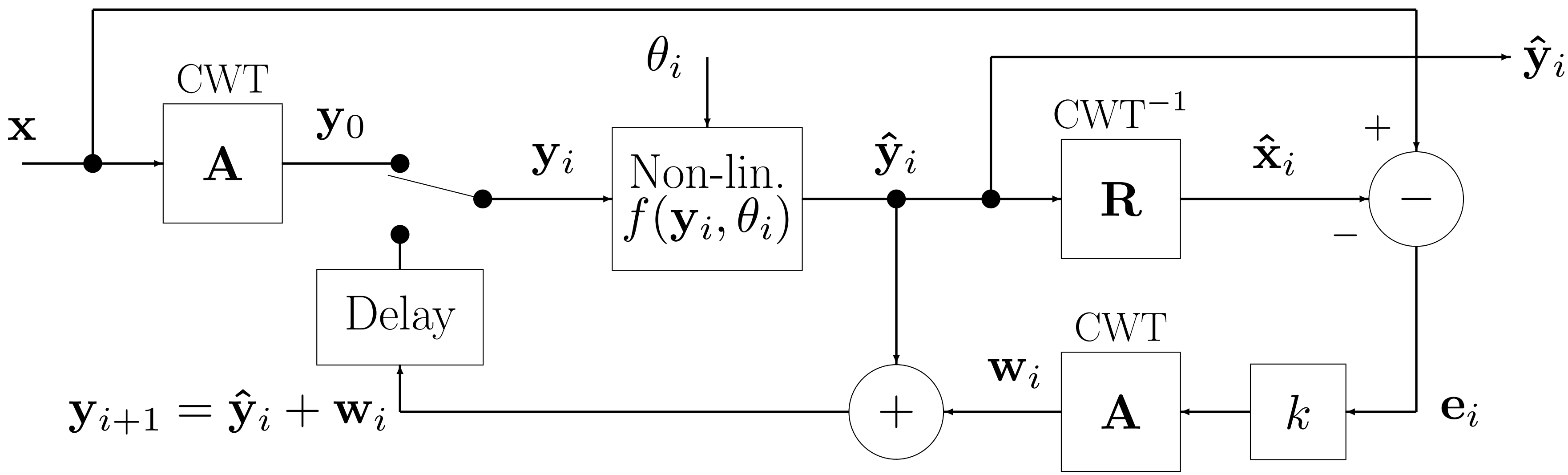
3. Take DT CWT^{-1} and measure the error due to loss of smaller coefs.

4. Take DT CWT of the error image and adjust the non-zero wavelet coefs from step **2** to reduce the error.

5. Increment i, reduce θ_i a little (to include a few more non-zero coefs) and repeat steps **2** to **4**.

6. When there are sufficient non-zero coefs to give the required rate-distortion tradeoff, keep θ_i constant and iterate a few more times until converged.

Iterative Projection



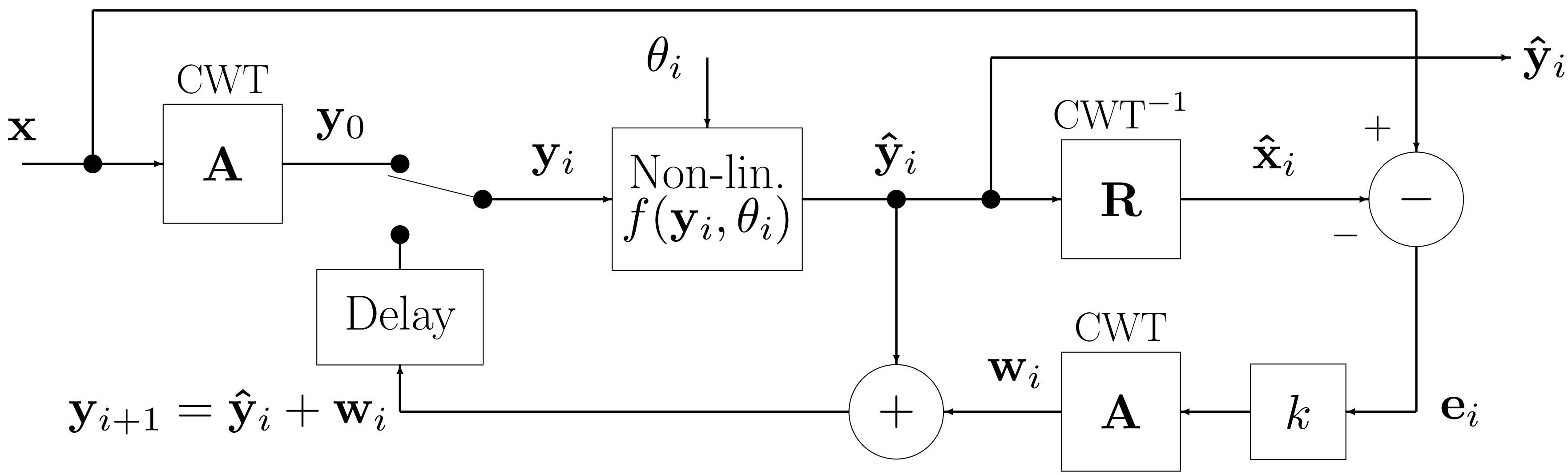
If $\mathcal{S}$ is the range space of the DT CWT, projection onto $\mathcal{S}$ is $\mathbf{P}^{\mathcal{S}} = \mathbf{AR}$, and onto the null space is $\mathbf{P}^{\perp} = \mathbf{I} - \mathbf{P}^{\mathcal{S}}$.

On iteration i: $\quad \mathbf{w}_i = k\mathbf{A}(\mathbf{x} - \mathbf{R}\hat{\mathbf{y}}_i) = k\mathbf{y}_0 - k\mathbf{P}^{\mathcal{S}}\hat{\mathbf{y}}_i$

$$\therefore\ \mathbf{y}_{i+1} = \hat{\mathbf{y}}_i + \mathbf{w}_i = k\mathbf{y}_0 + (\mathbf{I} - k\mathbf{P}^{\mathcal{S}})\hat{\mathbf{y}}_i = \mathbf{y}_0 + \mathbf{P}^{\perp}\hat{\mathbf{y}}_i \ \text{ if } k = 1$$

Thus on each iteration the range-space component of $\mathbf{y}_{i+1}$ remains at $\mathbf{y}_0$ (so its inverse transform is always $\mathbf{x}$) while its null-space component varies and attempts to minimise $||\mathbf{e}_i||$. Note that **$\mathbf{y}_{i+1}$ is a projection of $\hat{\mathbf{y}}_i$**.

Convergence

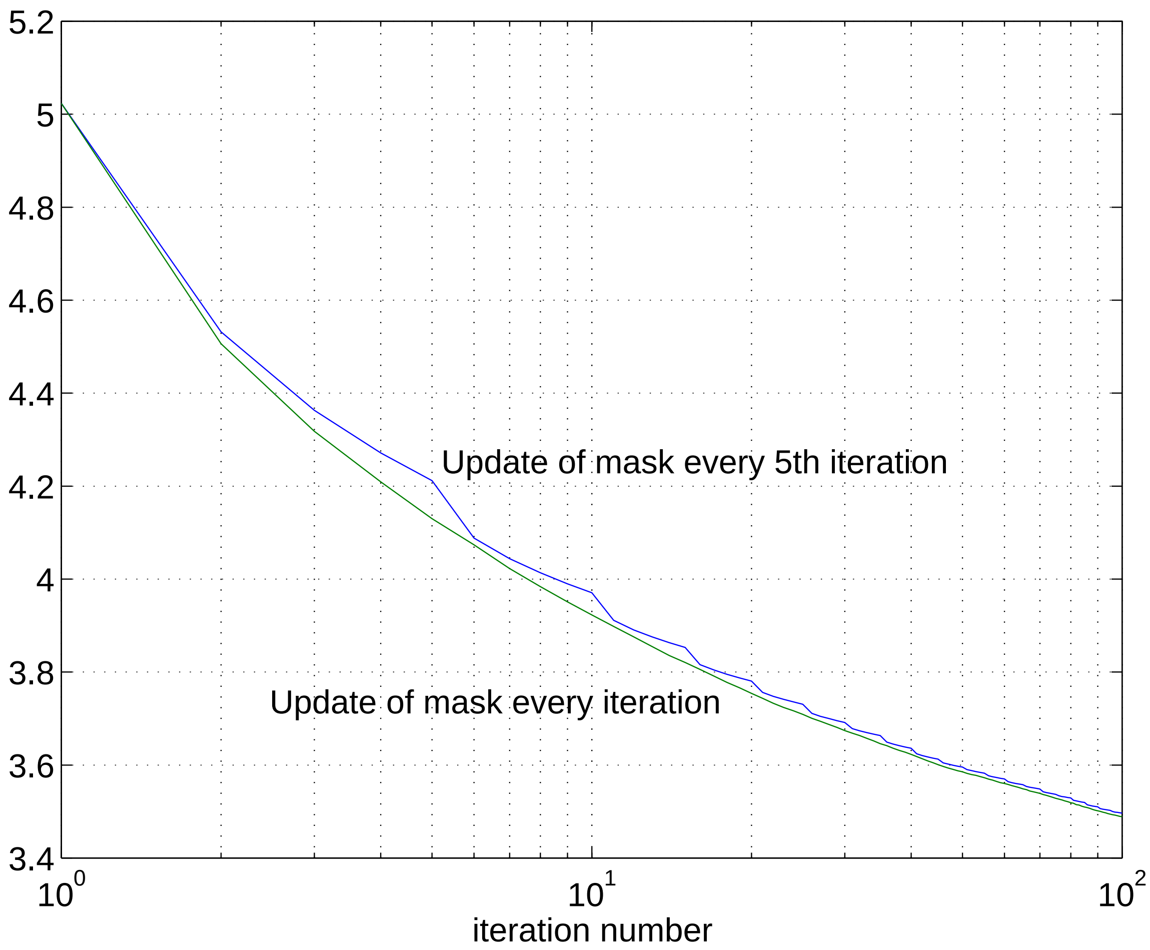


With a centre-clipping non-linearity and $k = 1$, convergence to a **local minimum** can be proved by **Projection onto Convex Sets** (POCS).

Substantial improvements in the converged result can be achieved by:

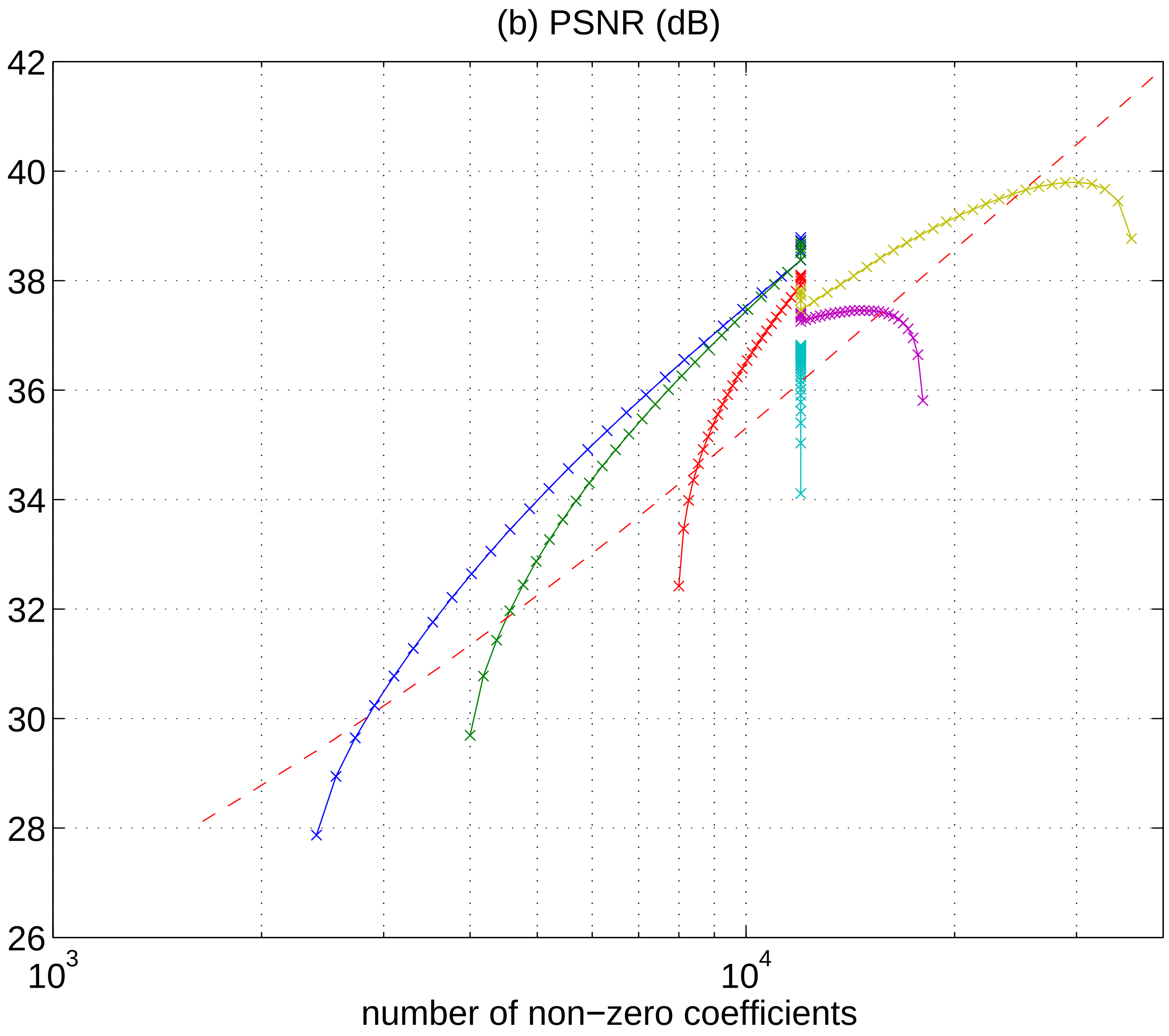
- Gradual **reductions in clip threshold** θ_i with i.
- Use of a **soft non-linearity**, such as Wiener function, for early iterations.
- **Increasing** k (must be kept < 2 for stability). $k \approx 1.8$ is good.

Convergence of loop RMS error for Centre-Clipper



The centre-clipper first selects a mask of coefs to clip, and then multiplies by the mask (a projection operation - hence can use POCS).

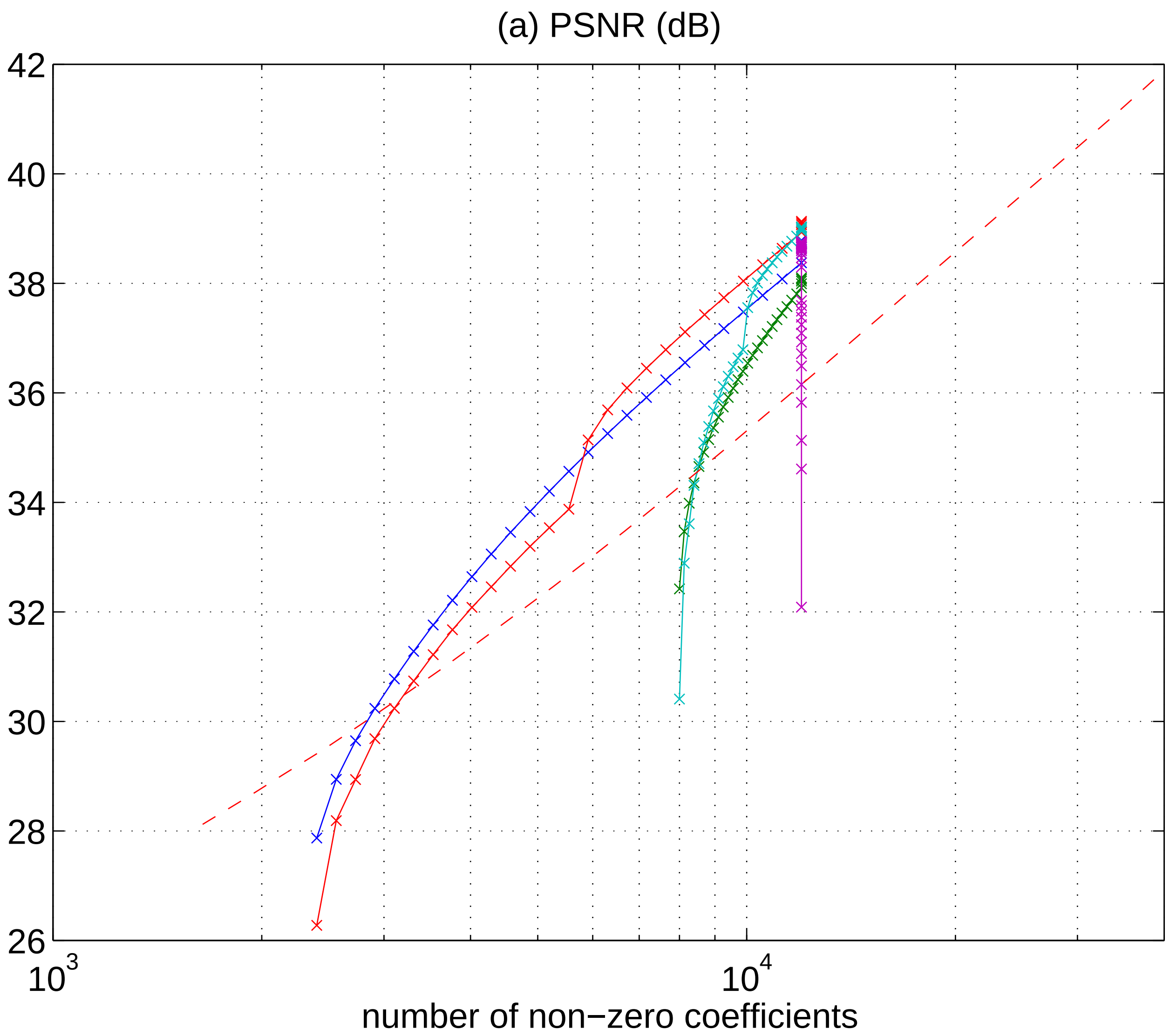
THRESHOLD MODIFICATION EXPERIMENTS FOR DT CWT ($k = 1$)



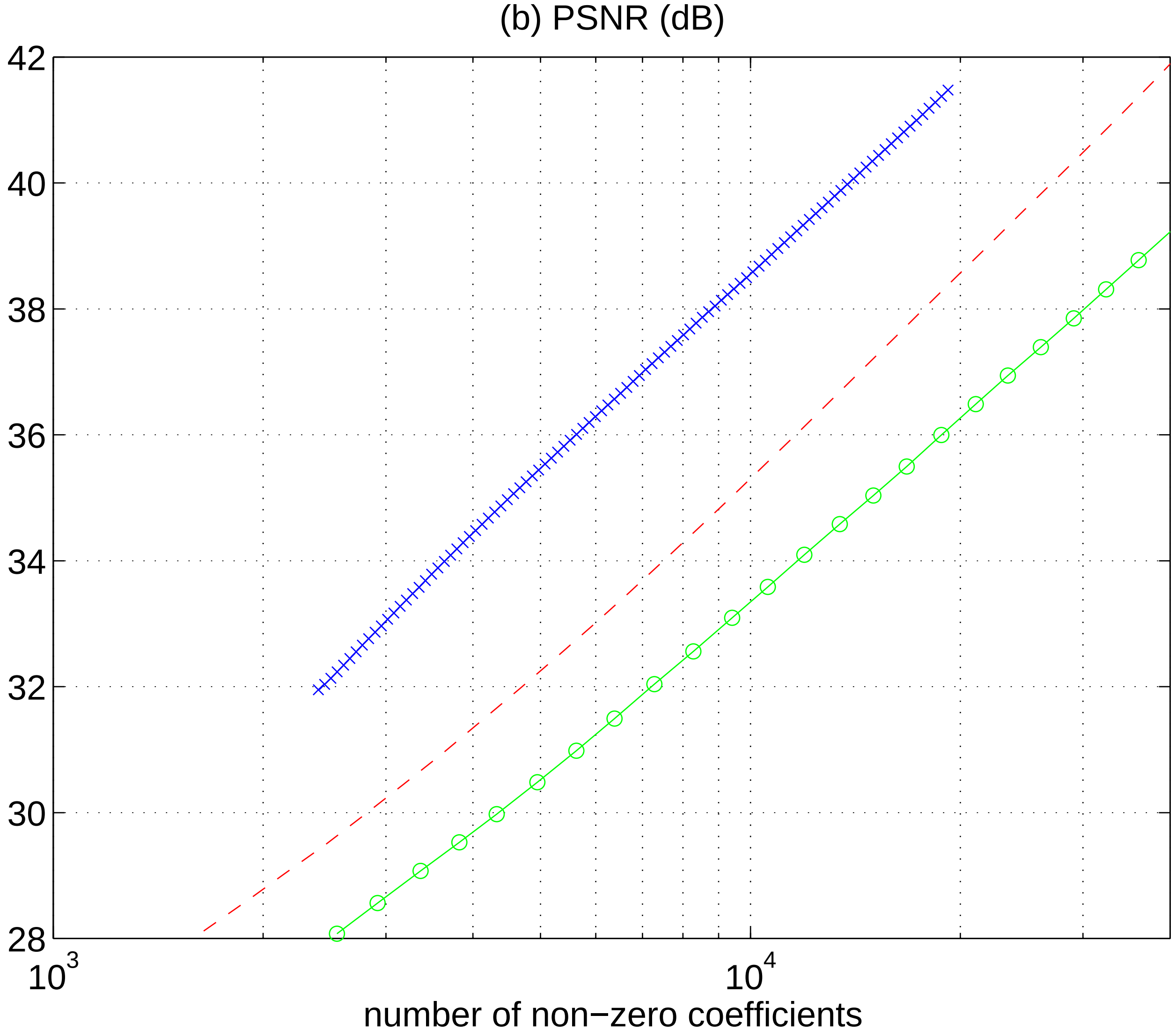
- - - shows non-redundant DWT for reference.

Threshold Modification Experiments:

$k = 1.8$ and Wiener non-linearity for first 15 iterations (better by $\mathbf{0.34}$ dB).

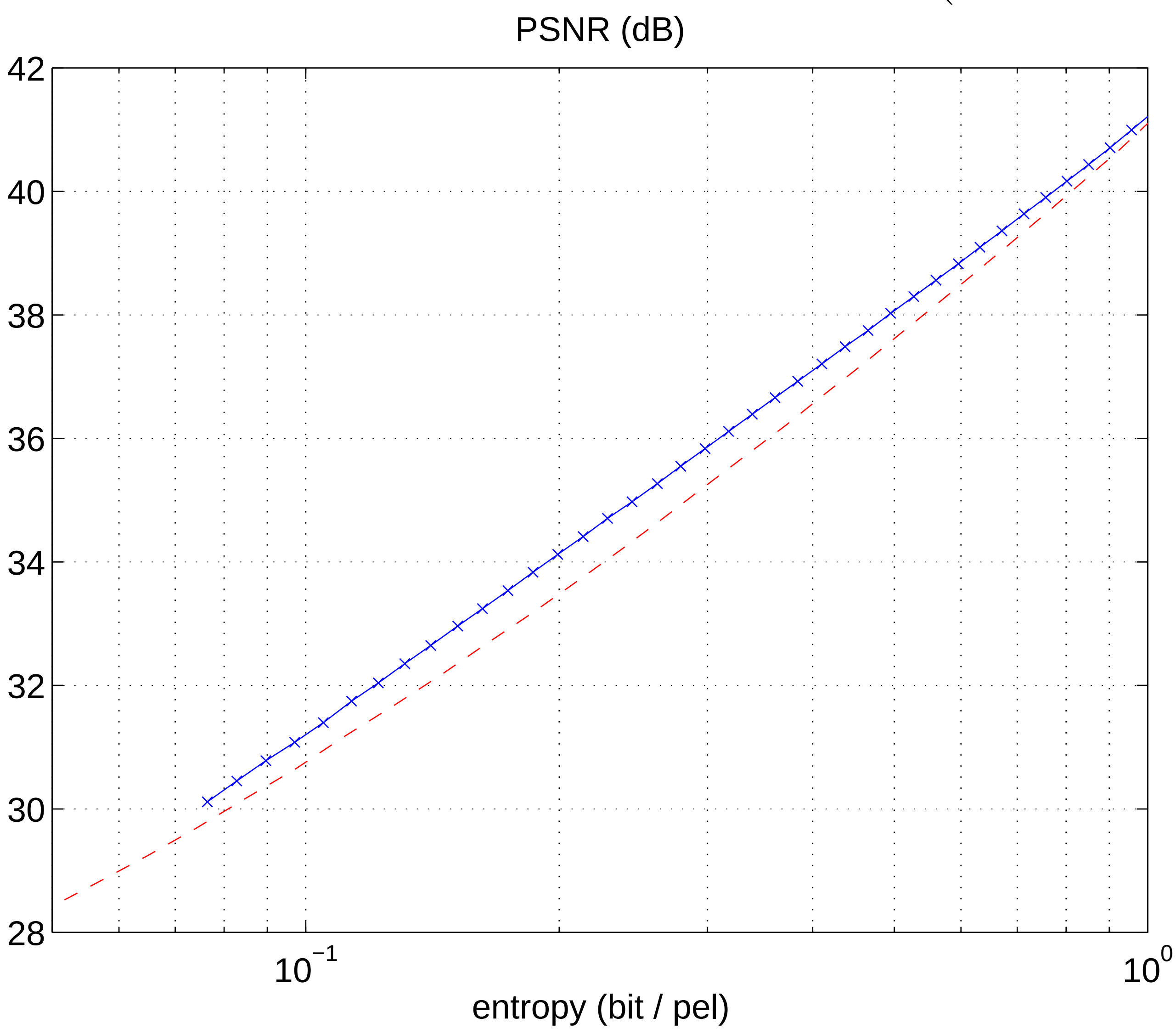


Comparison of DT CWT and DWT (centre-clipping only)



xxx Iterated DT CWT - - - DWT -o-o- non-iterated DT CWT

Compression results for 512×512 'Lena' image (fully quantised)



xxx Iterated DT CWT - - - DWT

Non-redundant DWT
0.0975 bit/pel (30.66 dB PSNR)

4:1 Overcomplete DT CWT
0.0970 bit/pel (31.08 dB PSNR)

Non-redundant DWT
0.1994 bit/pel (33.47 dB)

4:1 Overcomplete DT CWT
0.1992 bit/pel (34.12 dB)

Non-redundant DWT
0.3876 bit/pel (36.41 dB)

4:1 Overcomplete DT CWT
0.3839 bit/pel (36.93 dB)

Non-redundant DWT
0.8024 bit/pel (39.94 dB)

4:1 Overcomplete DT CWT
0.8018 bit/pel (40.17 dB)

Iterative Projection – Conclusions

- Reducing the centre-clipping threshold θ_i from an initial value that is at least twice the final value, as iterations proceed, improves performance.

- Setting $k = 1.8$ and using a soft non-linearity for early iterations improves performance and convergence rate.

- Despite a redundancy of $4:1$, the DT CWT can achieve coding performance that is competitive with the non-redundant DWT (PSNR 0.65 dB better).

- Visibility of some coding artifacts can be reduced with the DT CWT.

- With a suitably optimised convergence strategy, computation rate should be significantly less than for matching pursuits.

Key Features of Robust Registration Algorithms

- Edge-based methods are more robust than point-based ones.
- Must be automatic (no human picking of correspondence points) in order to achieve sub-pixel accuracy in noise.
- Bandlimited multiscale (wavelet) methods will allow spatially adaptive denoising.
- Phase-based bandpass methods can give rapid convergence and immunity to illumination changes between images.
- Displacement field should be smooth, so use of a wide-area parametric (affine) model is preferable to local translation-only models.

SELECTED METHOD

- Dual-tree Complex Wavelet Transform (DT CWT):
  - provides complex coefficients whose phase shift depends approximately linearly with displacement;
  - allows each subband of coefficients to be interpolated independently of other subbands (because of shift invariance).

- Parametric model of displacement field, whose solution is based on local edge-based motion constraints (Hemmendorf et al., IEEE Trans Medical Imaging, submitted 2002):
  - derives straight-line contraints from directional subbands of DT CWT;
  - solves for model parameters which minimise constraint error energy over multiple directions and scales.

PARAMETRIC MODEL: CONSTRAINT EQUATIONS

Let the displacement vector at the i^{th} location $\mathbf{x}_i$ be $\mathbf{v}(\mathbf{x}_i)$; and let $\tilde{\mathbf{v}}_i = \begin{bmatrix} \mathbf{v}(\mathbf{x}_i) \\ 1 \end{bmatrix}$.

A straight-line constraint on $\mathbf{v}(\mathbf{x}_i)$ can be written

$$\mathbf{c}_i^T \, \tilde{\mathbf{v}}_i = 0 \quad \text{or} \quad c_{1,i}v_{1,i} + c_{2,i}v_{2,i} + c_{3,i} = 0$$

For a phase-based system in which wavelet coefficients at $\mathbf{x}_i$ in images A and B have phases θ_A and θ_B, approximate phase linearity means that

$$\mathbf{c}_i = C_i \begin{bmatrix} \nabla_{\mathbf{x}} \, \theta(\mathbf{x}_i) \\ \theta_B(\mathbf{x}_i) - \theta_A(\mathbf{x}_i) \end{bmatrix}$$

In practise we compute this by averaging finite differences at the centre of a $2 \times 2 \times 2$ block of coefficients from images A and B.

C_i is a constant which does not affect the line defined by the constraint, but which is important later.

Parameters of the Model

We can define an affine parametric model for $\mathbf{v}$ such that

$$\mathbf{v}(\mathbf{x}) = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} a_3 & a_5 \\ a_4 & a_6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

or in a more useful form

$$\mathbf{v}(\mathbf{x}) = \begin{bmatrix} 1 & 0 & x_1 & 0 & x_2 & 0 \\ 0 & 1 & 0 & x_1 & 0 & x_2 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ \vdots \\ a_6 \end{bmatrix} = \mathbf{K}(\mathbf{x}) \ . \ \mathbf{a}$$

Affine models can synthesise translation, rotation, constant zoom, and shear.

A quadratic model, which allows for linearly changing zoom (approx perspective), requires up to 6 additional parameters and columns in $\mathbf{K}$ of the form

$$\begin{bmatrix} \dots & x_1 x_2 & 0 & x_1^2 & 0 & x_2^2 & 0 \\ \dots & 0 & x_1 x_2 & 0 & x_1^2 & 0 & x_2^2 \end{bmatrix}$$

Solving for the Model Parameters

Let $\tilde{\mathbf{K}}_i = \begin{bmatrix} \mathbf{K}(\mathbf{x}_i) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$ and $\tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix}$ so that $\tilde{\mathbf{v}}_i = \tilde{\mathbf{K}}_i \; \tilde{\mathbf{a}}$.

Ideally for a given image locality $\mathcal{X}$, we wish to find the parametric vector $\tilde{\mathbf{a}}$ such that

$$\mathbf{c}_i^T \; \tilde{\mathbf{v}}_i = 0 \quad \text{when} \quad \tilde{\mathbf{v}}_i = \tilde{\mathbf{K}}_i \; \tilde{\mathbf{a}} \quad \text{for all } i \text{ such that } \mathbf{x}_i \in \mathcal{X}.$$

In practise this is an overdetermined set of equations, so we find the LMS solution, the value of $\mathbf{a}$ which minimises the squared error

$$\mathcal{E}_{\mathcal{X}} = \sum_{i \in \mathcal{X}} ||\mathbf{c}_i^T \; \tilde{\mathbf{v}}_i||^2 = \sum_{i \in \mathcal{X}} ||\mathbf{c}_i^T \; \tilde{\mathbf{K}}_i \; \tilde{\mathbf{a}}||^2 = \tilde{\mathbf{a}}^T \; \tilde{\mathbf{Q}}_{\mathcal{X}} \; \tilde{\mathbf{a}}$$

where $\tilde{\mathbf{Q}}_{\mathcal{X}} = \sum_{i \in \mathcal{X}} (\tilde{\mathbf{K}}_i^T \; \mathbf{c}_i \; \mathbf{c}_i^T \; \tilde{\mathbf{K}}_i)$.

Solving for the Model Parameters (cont.)

Since $\tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix}$ and $\tilde{\mathbf{Q}}_{\mathcal{X}}$ is symmetric, we define $\tilde{\mathbf{Q}}_{\mathcal{X}} = \begin{bmatrix} \mathbf{Q} & \mathbf{q} \\ \mathbf{q}^T & q_0 \end{bmatrix}_{\mathcal{X}}$ so that

$$\mathcal{E}_{\mathcal{X}} = \tilde{\mathbf{a}}^T \, \tilde{\mathbf{Q}}_{\mathcal{X}} \, \tilde{\mathbf{a}} = \mathbf{a}^T \, \mathbf{Q} \, \mathbf{a} + 2 \, \mathbf{a}^T \mathbf{q} + q_0$$

$\mathcal{E}_{\mathcal{X}}$ is minimised when $\nabla_{\mathbf{a}} \, \mathcal{E}_{\mathcal{X}} = 2 \, \mathbf{Q} \, \mathbf{a} + 2 \, \mathbf{q} = \mathbf{0}$, so $\mathbf{a}_{\mathcal{X},\mathrm{min}} = - \, \mathbf{Q}^{-1} \, \mathbf{q}$.

The choice of locality $\mathcal{X}$ will depend on application:

- If it is expected that the affine (or quadratic) model will apply accurately to the whole image, then $\mathcal{X}$ can be the whole image and maximum robustness will be achieved.

- If not, then $\mathcal{X}$ should be a smaller region, chosen to optimise the tradeoff between robustness and model accuracy. A good way to produce a smooth field is to make $\mathcal{X}$ fairly small (e.g. a 32×32 pel region) and then to apply a smoothing filter across all the $\tilde{\mathbf{Q}}_{\mathcal{X}}$ matrices, element by element, before solving for $\mathbf{a}_{\mathcal{X},\mathrm{min}}$ in each region.

Constraint Weighting Factors

Returning to the equation for the constraint vectors, $\mathbf{c}_i = C_i \begin{bmatrix} \nabla_{\mathbf{x}}\, \theta(\mathbf{x}_i) \\ \theta_B(\mathbf{x}_i) - \theta_A(\mathbf{x}_i) \end{bmatrix}$,

the constant gain parameter C_i will determine how much weight is given to each constraint in $\tilde{\mathbf{Q}}_{\mathcal{X}} = \sum_{i \in \mathcal{X}} (\tilde{\mathbf{K}}_i^T\, \mathbf{c}_i\, \mathbf{c}_i^T\, \tilde{\mathbf{K}}_i)$.

Hemmendorf proposes some quite complicated heuristics for computing C_i, but for the DT CWT, we find the following works well:

$$C_i = \frac{|d_{AB}|^2}{\sum_{k=1}^{4} |u_k|^3 + |v_k|^3} \quad \text{where} \quad d_{AB} = \sum_{k=1}^{4} u_k^*\, v_k$$

and $\begin{bmatrix} u_1 & u_2 \\ u_3 & u_4 \end{bmatrix}$ and $\begin{bmatrix} v_1 & v_2 \\ v_3 & v_4 \end{bmatrix}$ are 2×2 blocks of wavelet coefficients centred on $\mathbf{x}_i$ in images A and B respectively.

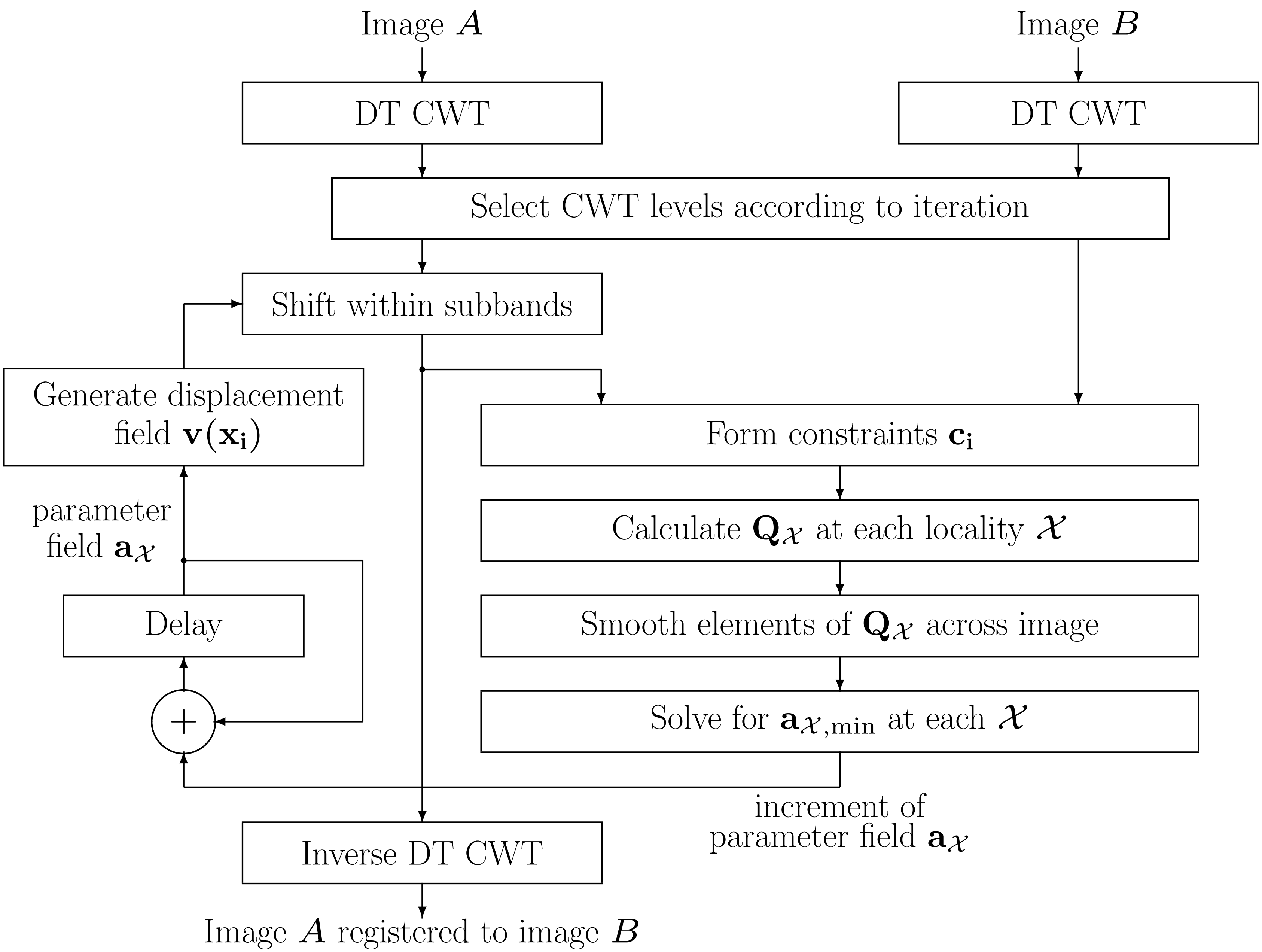


Image A
Image B
DT CWT
DT CWT
Select CWT levels according to iteration
Shift within subbands
Generate displacement field $\mathbf{v}(\mathbf{x_i})$
Form constraints $\mathbf{c_i}$
parameter field $\mathbf{a}_{\mathcal{X}}$
Calculate $\mathbf{Q}_{\mathcal{X}}$ at each locality $\mathcal{X}$
Delay
Smooth elements of $\mathbf{Q}_{\mathcal{X}}$ across image
$+$
Solve for $\mathbf{a}_{\mathcal{X},\min}$ at each $\mathcal{X}$
increment of parameter field $\mathbf{a}_{\mathcal{X}}$
Inverse DT CWT
Image A registered to image B

Conclusions

The **Dual-Tree Complex Wavelet Transform** provides:

- Approximate **shift invariance**
- **Directionally selective** filtering in 2 or more dimensions
- **Low redundancy** – only $2^m : 1$ for m-D signals
- **Perfect reconstruction**
- **Orthonormal filters** below level 1, but still giving **linear phase** (conjugate symmetric) complex wavelets
- **Low computation** – order-N; less than 2^m times that of the fully decimated DWT (~ 3.3 times in 2-D, ~ 5.1 times in 3-D)

Conclusions (cont.)

- A **general purpose multi-resolution front-end** for many image analysis and reconstruction tasks:
  - Enhancement (deconvolution)
  - Denoising
  - Motion / displacement estimation and compensation
  - Texture analysis / synthesis
  - Segmentation and classification
  - Watermarking
  - 3D data enhancement and visualisation
  - Coding (?)

Papers on complex wavelets are available at:

http://www.eng.cam.ac.uk/~ngk/

A Matlab DTCWT toolbox is available on request from:

ngk@eng.cam.ac.uk