green radiation yielding a 15% conversion efficiency, probably limited mainly by the mismatch of the 20 μm period with that required for exact QPM at 1.085 μm.

![Graphs showing pulse peak power and width as function of coupled pump power, repetition rate and applied voltage.](image)

**Fig. 2** Pulse peak power and width as function of coupled pump power, repetition rate and applied voltage.

- a. Coupled pump power $P_c (f = 1.2 kHz, \tau = 35 \mu s)$
- b. Repetition rate $f (P_c = 14 mW, f = 35 \mu s)$
- c. Applied voltage $V (P_c = 14 mW, f = 1.2 kHz)$

- peak power
- pulse width

In summary, we have reported a short pulse and high power monolithic Q-switched Nd: MgO: LiNbO$_3$ waveguide laser. This pulsed source requires only 15 mW of coupled pump power so inexpensive commercial diode lasers could be used. Although further improvements in the laser performance could be achieved they are already suitable for use in ranging, OTDR and time-multiplexed fibre sensor systems applications. In addition, the integration of periodic domain-reversal sections in the laser cavity should enable efficient frequency conversion through second-harmonic and parametric processes.

29th October 1992

E. Lallier, D. Papillon, J. P. Pocholle and M. Papuchon (Thomson-CSF, Laboratoire Central de Recherches, Domaine de Corbeville, 91404 Orsay, France)

M. De Micheli and D. B. Ostrowsky (Laboratoire de Physique de la Matière Condensée, Université de Nice, 06034 Nice, France)

References


2. SIEGMAN, A. E.: 'Lasers' (University Science Books, CA, USA)


**SUPPRESSION OF NONSTATIONARY SINUSOIDAL INTERFERENCE USING TRANSFORM DOMAIN MEDIAN FILTERING**

T. Kasperis

**Indexing terms:** Filters, Digital signal processing, Communications

A novel technique for the suppression of sinusoidal interference by using transform domain median filtering is proposed. The most important advantage of this new technique is that estimation of the interference frequency is not needed. Other important advantages are also discussed. Residual errors are examined and compared with the use of notch filtering. Experimental results with random binary sequences are presented.

**Introduction:** Traditionally, suppression of narrowband interference from wideband signals is accomplished in the time domain with a variety of notch filters. With recent technology, processing can also be performed directly in the frequency domain by using real-time Fourier transformation as an approximation to sectioning and possibly windowing to reduce spectral leakage. It is usually required in the transformation process. References 1-4 provide more details. In the transform domain interference appears as large and narrow frequency components that have been suppressed. Either by notching (assuming known frequencies) [2, 3], or by dedicated hardware that can reduce nonstationary interference as well [4]. With the latter approach, difficulties may arise when multiple interference is present, and signal distortion is possible when interference is not present.

**Proposed approach:** The problem of suppressing large interference impulses from a signal transform where smaller impulsive components may also be present, is similar to the problem of suppressing impulsive noise from an image while preserving the finer detail. This fact inspired the idea of using filters traditionally used in image processing, for transform domain interference suppression. The median filter (MF) has been a popular filter because it has the ability to suppress narrow impulses while preserving smoother signals [5]. However, MFs suppress impulses of any amplitude and they usually introduce additional undesirable smoothing. A conditional median filter (CMF) that we have proposed suppresses only sufficiently large impulses and has been very successful in removing impulse noise from images while preserving the finer detail [6]. We propose in this Letter the use of a CMF in the frequency domain to suppress large interference impulses. A CMF is defined by the following equation:

$$y_i = \begin{cases} x_i & \text{if } |m_i - x_i| < C \\ m_i & \text{otherwise} \end{cases}$$

(1)

where $x_i, y_i$ are the input and output data sequences, $m_i$ is the median of an odd sized sliding window of length $N$, and $C$ is a threshold. Applied on either the real or imaginary part of the magnitude of the transform components, the CMF essentially performs median filtering only on signal values that satisfy the threshold condition in eqn. 1. Because MFs favour smooth signals, better results can be obtained by filtering the smoothest transform components. For binary signals that we have considered, the magnitude-phase was a better choice, but because the phase did not always present interference impulses, we processed the magnitude only. Large impulses will be suppressed if they are also narrower than $(N - 1)/2$, otherwise they will be preserved [5]. With the two filter parameters $N$ and $C$ we can control the minimum amplitude and maximum bandwidth of impulses to be suppressed. The position of an impulse in the signal is irrelevant; the filter will suppress it anywhere it is encountered. Fig. 1 presents a CMF filtering example where Fig. 1a displays the frequency magnitude of a binary sequence, and also the same magnitude corrupted by four sinusoidal interferers of unknown frequencies. Fig. 1b presents the result when the corrupted magnitude is filtered with a CMF$_{13}$ (window size 13) and $C = 50$. The
result obtained when \( C = 0 \), is also displayed in Fig 1b, and the disadvantage is obvious. We are currently developing a technique to adapt \( C \) when the signal power varies, and the results will be reported soon.

![Graph](image)

Fig 1 CMF filtering example

- **a** Original and corrupted signals
  - Original
  - Corrupted
- **b** Outputs of CMF, with \( C = 50 \) and \( C = 0 \)
  - \( C = 50 \)
  - \( C = 0 \)

This new suppression technique offers several advantages. The centre frequency and bandwidth of each interferer do not have to be determined. Because there is no adaptation time, nonstationary interference can be tracked more quickly and accurately. Multiple interference is handled without any modifications, and no filtering is performed when no interference is present. Another advantage is simplicity, and because VISI components are available for both MF and Fourier transforms, simple real-time implementations are feasible.

**Residual errors:** A general error analysis is complicated by the nonlinear nature of the CMF. To obtain some performance results we will make the following simplifying assumptions: a single large sinusoidal interferer is present at \( \omega = \omega_0 \), and the CMF threshold is properly adjusted so that actual filtering takes place only at frequency \( \omega_0 \). Let \( x(t), x_t(t), x(t) \) be the original, corrupted, and recovered signals, and \( (G_0(l), \phi_0(l)), (G_0(l), \phi_0(l)), (G_0(l), \phi_0(l)) \) be their Fourier magnitude-phase transforms, respectively, where \( G_0(l) \) is the output of a CMF, applied on \( x(l) \). We define:

\[
\epsilon(l) = x_t(l) - \phi_0(l) \quad \text{phase error due to interference} \quad (2)
\]

\[
\tilde{\epsilon}(l) = x_t(l) - \tilde{x}_t(l) \quad \text{time-domain residual error} \quad (3)
\]

Based on the two assumptions we see that \( \epsilon(l) \) will be a sinusoid of frequency \( \omega_0 \). The average power of \( \epsilon(l) \) can be computed from eqns. 2 and 3 and the result is found to be given by

\[
P_{\epsilon}(\omega_0) = G_0^2(\omega_0) + G_0^2(\omega_0) - 2G_0(\omega_0)G_0(\omega_0) \cos \left( \angle \omega_0 \right) \quad (4)
\]

We notice from eqn. 4 that \( P_{\epsilon} \) depends also on the phase error introduced by the interference. For a comparison, a similar calculation for a notch filter of \( M \) discrete frequencies yields

\[
P_{\epsilon, \text{notch}} = \sum_{n=1}^{M} G_0^2(\omega_n) \quad (5)
\]

where \( \omega_n \) is one of the \( M \) frequencies \( \omega_n \) to be notched.

**Application:** One primary application of the proposed technique that we are considering is antijam processing in direct sequence spread-spectrum (DS-SS) communication systems. Some preliminary simulation results have been reported in Reference 7. We will examine here the residual error power of pseudorandom binary sequences (widely known as PN sequences) used in these systems. For these signals the error power \( P_{\epsilon}(\omega_0) \) given by eqn. 4 will be a random variable. The mean value of \( P_{\epsilon}(\omega_0) \) can be computed from the statistics of each of the terms in eqn. 4, but such an analysis will be presented elsewhere. Instead, only some results will now be presented. Returning to eqn. 4, two results that can also be experimentally verified are

\[
G_0(\omega_0) G_0(\omega_0) \cos \left( \angle \omega_0 \right) = 0 \quad (6)
\]

\[
G_0(\omega_0) \approx G_0(\omega_0) \quad \text{for} \ N \geq 5 \quad (7)
\]

where \( N \) is the CMF window size and the overbar indicates the mean values. By using eqns. 6 and 7 in eqn. 4 the mean value of \( P_{\epsilon} \) is

\[
P_{\epsilon}(\omega_0) \approx 2G_0^2(\omega_0) \quad (8)
\]

The mean error power is proportional to \( G_0^2(\omega_0) \), i.e. the power spectral density of the signal at the frequency of the interference. For a narrow notch filter we find from eqn. 5 by approximating \( G(l) \approx G_0(\omega_0) \), that

\[
P_{\epsilon, \text{notch}}(\omega_0) \approx M G_0^2(\omega_0) \quad \text{for small} \ M \quad (9)
\]

By comparing eqns. 8 and 9 we conclude that without processing the phase component of the transform, the performance of the proposed technique on average residual error power is similar to that of the \( M = 2 \) notch filter. In an upcoming communication we will relate the average error power with the bit error rate of a DS-SS system when noise is also present.

**Experimental results:** We recovered 32-chip PN sequences from a randomly phased sinusoidal interferer of power 9 dB above the signal. Transformation was implemented using a 128-point FFT, and some spectral leakage was allowed. For the suppression we used a CMF, with \( C = 50 \), and also notch filters of variable notch-width (assuming known interference frequency). The residual error power averaged over 10,000 different sequences is plotted in Figure 2 against the interference frequency. As expected, the performance of the CMF technique is similar to that of the notch-2 filter, whereas wider notches are well outperformed.

![Graph](image)

**Fig. 2** Average residual error powers against interference frequency for 32 chip PN sequences

**Future work:** As indicated by eqn. 4, the performance of the technique can be further enhanced by filtering that minimises the phase error. We are currently pursuing this goal.

25th November 1992
T. Kasparis (Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816, USA)

**References**

LATTICE QUANTISATION FOR IMAGE SEQUENCES CODING AT 16 DIMENSION

D.G. Sampson and M. Ghanbari

Indexing terms: Image processing, Vector quantisation

The performance of a 16-dimensional lattice quantisation scheme is investigated. A simple \( E_4 \) type lattice which includes all the necessary shape patterns for reliable image reconstruction in a codebook of manageable size, is devised to replace the most dense \( \Lambda_{16} \). Simulation results demonstrate that \( E_4 \)-based LQ achieves better signal-to-noise ratio performance than \( \Lambda_{16} \), still at only 1/16 of \( \Lambda_{16} \) encoding complexity.

Introduction: Vector quantisation (VQ) due to the decoder simplicity is a popular video compression technique. However, the main drawback in implementing a VQ coding system is the high encoding complexity. Lattice quantisation (LQ) is a simple and fast alternative to vector quantisation. The main advantage of LQ is the simplified encoding process, attained by exploiting the well defined geometric structure of the root lattices [1]. We have proposed a practical method which incorporates the basic ideas of gain-shape vector quantisation with \( E_4 \) lattice code [2]. Simulation results demonstrate that \( E_4 \) LQ can achieve similar performance with VQ of 8-dimensional (8-D) only at a very small fraction of complexity [3]. Typically VQ is carried out at 16-D blocks, because theory anticipates that larger compression ratios can be achieved with higher dimensions. In this Letter we investigate the performance of LQ at \( k = 16 \)-D. The most dense root lattice in 16-D is the Barnes-Wall lattice \( \Lambda_{16} \) [4]. However, \( \Lambda_{16} \) is not realisable for practical image coding applications, because it requires a very highly populated codebook that cannot be entropy coded. Instead we propose the use of a simpler lattice which belongs to the family of \( E_4 \) lattices. Simulation results are provided to demonstrate that \( E_4 \)-based LQ achieves better signal-to-noise ratio performance than \( \Lambda_{16} \), still at only 1/16 of \( \Lambda_{16} \) encoding complexity.

Lattice quantisation at 16-D: The most dense root lattice at 16-D is the Barnes-Wall lattice \( \Lambda_{16} \). \( \Lambda_{16} \) can be constructed as the union of 32 translated versions of the simpler root lattice \( 2D_{16}, \) which is the set of even co-ordinates vectors of \( Z_{16}, \) such that the sum of the co-ordinates is a multiple of 4. Thus, \( \Lambda_{16} \) is defined as

\[
\Lambda_{16} = \bigcup_{i=1}^{32} (2D_{16} + e_i)
\]

(1)

where \( e_i \) denotes the rows of the Hadamard matrix \( H_{16} \) and its complementary \( H_{16} \) after changing the 1s to 0s and the -1s to 1s. Similar to any regular root lattice, the points of \( \Lambda_{16} \) are distributed on the surface of successive, concentric 16-D hyperspheres, so-called shells, centred at the origin. \( \Lambda_{16} \) consists of clusters of points at distances of \( \sqrt{m} \) from the origin, where \( m \) is the shell number. There are 0, 42, 6144 lattice points in shells 1, 2, and 3, respectively, and the population grows astronomically with the shell size, e.g., shell 4 contains 52222720 points. To use \( \Lambda_{16} \) as the shape codebook in a gain-shape lattice quantisation scheme, only the innermost three shells with the total population of 65760 points are realisable, because the design of a practical entropy coder (Huffman) is the bottleneck for the shape size. However, the performance of a \( \Lambda_{16} \)-based shape codebook with these limited shells is poor as important image patterns, such as \( [ \pm 1 ]^n, \) are not included in the codebook. To incorporate these patterns, at least shell 4 should be considered, resulting in a prohibitively large codebook population. On the other hand if only the most important points from various shells are included in the codebook, the advantage of a fast algorithm for selecting the nearest neighbouring point is diminished [7]. Instead of the Barnes-Wall \( \Lambda_{16} \), we define an \( E_4 \)-type lattice which is not as dense as \( \Lambda_{16} \). However, with manageable shell sizes it contains all the necessary shape patterns for image coding applications. We derive the \( E_4 \) lattice \( E_4 \) at 8-D from \( D_{16} \), in a similar way to which \( E_4 \) is related to \( D_4 \) [1]:

\[
E_4 = \{ D_{16} + [0^n]\} \cup \{ D_{16} + [0^4 + 0^8]\}
\]

(2)

\( E_4 \) can also be related to \( \Lambda_{16} \). By definition, \( \Lambda_{16} \) can be decomposed into 16 subsets of points based on the Hadamard matrix rows:

\[
\Lambda_{16} = \bigcup_{i=1}^{16} \{ D_{16} + [0110010001100100]\}
\]

(3)

That is, \( 2E_4 \) is a subset of \( \Lambda_{16} \), as the first set of eqn. 3 shows. In \( 2E_4 \), there are 480 and 61920 lattice points at distances of \( \sqrt{3} \) and 2, respectively, from the origin. Although \( E_4 \) is not the most dense regular root lattice in 16-D, its first and second shells have 6400 points, they can roughly represent all the main directions and they can easily be variable length coded with a modified Huffman code. Any shape patterns which are not present in the truncated \( E_4 \) codebook can be coded in a two-stage LQ, as the error vector has a different orientation from the first stage input vector [3]. Moreover, the encoding algorithm which selects the best-matched shape code vector based on the geometrical structure of \( E_4 \) is 16 times simpler compared with \( \Lambda_{16} \). Table 1 gives the exact number of mathematical operations performed by the \( E_4 \) and \( \Lambda_{16} \)-based LQ encoders, along with the computational complexity of a conventional gain-shape VQ with 8 bit codebook.

Simulation results: The performance of LQ is assessed using a simple interframe DPCM. Experiments are carried out using a CIF standard test image sequence of 360 x 288 pixel

| Table 1 COMPARISON BETWEEN ENCODING COMPLEXITY OF \( E_{16} \), \( \Lambda_{16} \), AND 8 bit VQ |
|-----------------|-----------------|-----------------|
| LQ-E16 | LQ-\( \Lambda_{16} \) | VQ-256 |
| Multiplications | 48 | 768 | 4096 |
| Additions | 139 | 2224 | 3840 |
| Comparisons | 94 | 1504 | 318 |
| Total | 281 | 4496 | 8254 |
| \( E_{16} \) complexity ratio | 1:16 | 1:29 |

ELECTRONICS LETTERS 21st January 1993 Vol. 29 No. 2