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.

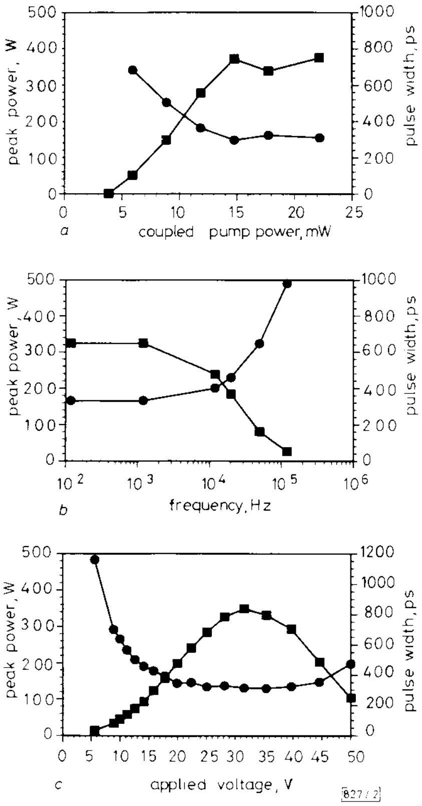

Fig. 2 *Pulse peak power and width as function of coupled pump power, repetition rate and applied voltage*

a Coupled pump power P_p (f = 1·2 kHz, v = 35 V)
b Repetition rate f (P_p = 14 mW, v = 35 V)
c Applied voltage v (P_p = 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 performances 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.

20th 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

1 LALLIER, E., POCHOLLE, J. P., PAPUCHON, M., HE, Q., DE MICHELI, M., and OSTROWSKY, D. B.: 'Integrated Q-switched Nd : MgO : LiNbO$_3$ waveguide laser', *Electron. Lett.*, 1992, **28**, (15), pp. 1428–1429

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

3 ARMANI, F., DELACOURT, D., LALLIER, E., PAPUCHON, M., HE, Q., DE MICHELI, M., and OSTROWSKY, D. B.: 'First order quasiphase matching in LiNbO$_3$', *Electron. Lett.*, 1992, **28**, (2), pp. 139–140

SUPPRESSION OF NONSTATIONARY SINUSOIDAL INTERFERENCE USING TRANSFORM DOMAIN MEDIAN FILTERING

T. Kasparis

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 that 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 transforms. Some signal preprocessing such as sectioning and possibly windowing to reduce spectral leakage 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 be also 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:

$$y_i = \begin{cases} x_i & \text{if } |m_i - x_i| < C \\ m_i & \text{otherwise} \end{cases} \qquad (1)$$

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-imaginary or the magnitude-phase 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 smoother transform components. For the 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 supressed 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. 1*a* displays the frequency magnitude of a binary sequence, and also the same magnitude corrupted by four sinusoidal interferers of unknown frequencies. Fig. 1*b* 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. 1*b*, 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.

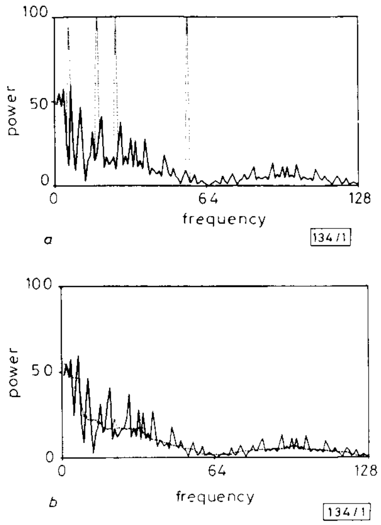

Fig. 1 *CMF filtering example*
a Original and corrupted signals
——— original
······· corrupted
b Outputs of CMF_{13} with $C = 50$ and $C = 0$
——— $C = 50$
······· $C = 0$

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 VLSI 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 ω_0. Let $x_0(t)$, $x_n(t)$, $\hat{x}_0(t)$ be the original, corrupted and recovered signals, and $\{G_0(\omega), \phi_0(\omega)\}$, $\{G_n(\omega), \phi_n(\omega)\}$, $\{\hat{G}_0(\omega), \phi_n(\omega)\}$ be their Fourier magnitude-phase transforms, respectively, where $\hat{G}_0(\omega)$ is the output of a CMF_N applied on $G_n(\omega)$. We define:

$$\varepsilon(\omega) = \phi_n(\omega) - \phi_0(\omega) \qquad \text{phase error due to interference} \quad (2)$$

$$e(t) = x_0(t) - \hat{x}_0(t) \qquad \text{time-domain residual error} \quad (3)$$

Based on the two assumptions we see that $e(t)$ will be a sinusoid of frequency ω_0. The average power of $e(t)$ can be computed from eqns. 2 and 3 and the result is found to be given by

$$P_e(\omega_0) = G_0^2(\omega_0) + \hat{G}_0^2(\omega_0) - 2G_0(\omega_0)\hat{G}_0(\omega_0)\cos[\varepsilon(\omega_0)] \quad (4)$$

We notice from eqn. 4 that P_e 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

$$P_{e,notch} = \sum_{i=1}^{M} G_0^2(\omega_i) \quad (5)$$

where ω_0 is one of the M frequencies ω_i 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 $P_e(\omega_0)$ given by eqn. 4 will be a random variable. The mean value of $P_e(\omega_0)$ 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

$$\overline{G_0(\omega_0)\hat{G}_0(\omega_0)\cos[\varepsilon(\omega_0)]} \simeq 0 \quad (6)$$

$$\overline{\hat{G}_0^2(\omega_0)} \simeq \overline{G_0^2(\omega_0)} \qquad \text{for } N \geq 5 \quad (7)$$

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_e is

$$\bar{P}_e(\omega_0) \simeq 2\overline{G_0^2(\omega_0)} \quad (8)$$

The mean error power is proportional to $\overline{G_0^2(\omega_0)}$, 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 $\overline{G_0^2(\omega_i)} \simeq \overline{G_0^2(\omega_0)}$, that

$$\bar{P}_{e,notch}(\omega_0) \simeq M\overline{G_0^2(\omega_0)} \qquad \text{for small } M \quad (9)$$

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_{13} with $C = 50$, and also notch filters of variable notch-width (assuming known interference frequency). The residual error power averaged over 10000 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.

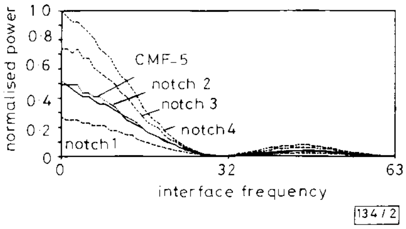

Fig. 2 *Average residual error powers against interference frequency for 32 chip PN sequences*

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

1 SHYNK, J.: 'Frequency domain and multirate adaptive filtering', *IEEE Signal Processing Magazine*, January 1992, pp. 14–37

2 MILSTEIN, L. B., and DAS, P. K.: 'An analysis of a real-time transform domain filtering digital communication system—part I: narrowband interference rejection', *IEEE Trans.*, 1980, **COM-28**, (6), pp. 816–824

3 DAVIDOVICI, S., and KANTERAKIS, E.: 'Narrow-band interference rejection using real-time Fourier transforms', *IEEE Trans.*, 1989, **COM-37**, (7), pp. 713–722

4 GEVARGIZ, J., DAS, P. K., and MILSTEIN, L. B.: 'Adaptive narrow-band interference rejection in a DS spread-spectrum intercept receiver using transform domain signal processing techniques', *IEEE Trans.*, 1989, **COM-37**, (12), pp. 1359–1366

5 PITAS, I., and VENETSANOPOULOS, A. N.: 'Non-linear digital filters' (Kluwer Academic Publishers, 1990)

6 KASPARIS, T., TZANNES, N., and CHEN, Q.: 'Conditional median filters for selective impulse suppression'. Proc. Third Int. Conf. on Communications and Control (COMCON-3), October 1991, pp. 229–238

7 KASPARIS, T., GEORGIOPOULOS, M., and PAYNE, E.: 'Non-linear filtering techniques for narrow-band interference suppression in direct sequence spread-spectrum systems'. Proc. Int. Conf. on Military communications (MILCOM), November 1991, pp. 360–364

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_k-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 Λ_{16}. Simulation results demonstrate that E_{16}-based LQ achieves better signal-to-noise ratio performance than Λ_{16}, still at only 1 : 16 of Λ_{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_8 lattice code [2]. Simulation results demonstrate that E_8 LQ can achieve similar performance with VQ of 8-dimension (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 Λ_{16} [4]. However, Λ_{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_k lattices. Simulation results are provided to demonstrate that E_{16}-based LQ achieves better signal-to-noise ratio ratio performance than Λ_{16}, still at only 1 : 16 of Λ_{16} encoding complexity.

Lattice quantisation at 16-D: The most dense root lattice at 16-D is the Barnes–Wall lattice Λ_{16} [4]. Λ_{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, Λ_{16} is defined as

$$\Lambda_{16} = \bigcup_{i=1}^{32} \{2D_{16} + c_i\} \qquad i = 1, 2, \ldots, 32 \quad (1)$$

where c_i denotes the rows of the Hadamard matrix H_{16}, and its complementary $\bar{H}_{16}$ after changing the 1s to 0s and the -1s to 1s. Similar to any regular root lattice, the points of Λ_{16} are distributed on the surface of successive, concentric 16-D hyperspheres, so-called shells, centred at the origin. Λ_{16} consists of clusters of points at distances of $2\sqrt{(m)}$ from the origin, where m is the shell number. There are 0, 4320, 61440 lattice points in shells 1, 2 and 3, respectively, and the population grows astronomically with the shell size, e.g. shell 4 contains 5 222 720 points. To use Λ_{16} as the shape codebook in a gain-shape lattice quantisation scheme, only the innermost three shells with the total population of 65 760 points are realisable, because the design of a practical entropy coder (Huffman) is the bottleneck for the shell size. However, the performance of a Λ_{16}-based shape codebook with these limited shells is poor as important image patterns, such as $[\pm 1^{16}]$, are not included in the codebook. To incorporate those 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 [3].

Instead of the Barnes–Wall Λ_{16} we define an E_k-type lattice which is not as dense as Λ_{16}. However, with manageable shell sizes it contains all the necessary shape patterns for image coding applications. We derive the Gosset lattice E_k at 16-D from D_{16}, in a similar way to which E_8 is related to D_8 [1]:

$$E_{16} = \{D_{16} + [0^{16}]\} \bigcup \{D_{16} + [0{\cdot}5^{16}]\} \qquad (2)$$

E_{16} can also be related to Λ_{16}. By definition, Λ_{16} can be decomposed into 16 subsets of points based on the Hadamard matrix rows:

$$\Lambda_{16} = \bigcup_{i=1}^{16} \left\{ \begin{array}{l} \{2D_{16} + [0^{16}]\} \\ \quad \bigcup \{2D_{16} + [1^{16}]\} \\ \{2D_{16} + [0101010101010101]\} \\ \quad \bigcup \{2D_{16} + [1010101010101010]\} \\ \{2D_{16} + [0011001100110011]\} \\ \quad \bigcup \{2D_{16} + [1100110011001100]\} \\ \ldots \\ \{2D_{16} + [0110100110010110]\} \\ \quad \bigcup \{2D_{16} + [1001011001101001]\} \end{array} \right\} \qquad (3)$$

That is, $2E_{16}$ is a subset of Λ_{16}, as the first set of eqn. 3 shows. In $2E_{16}$ there are 480 and 61 920 lattice points at distances of $2\sqrt{(2)}$ and 4, respectively, from the origin. Although E_{16} is not the most dense regular root lattice in 16-D, its first and second shells have 62 400 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_{16}-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_{16} is 16 times simpler compared with Λ_{16}. Table 1 gives the exact number of mathematical operations performed by the E_{16} and Λ_{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 × 288 pixel

Table 1 COMPARISON BETWEEN ENCODING COMPLEXITY OF E_{16}, Λ_{16}, AND 8 bit VQ

	LQ-E_{16}	LQ-Λ_{16}	VQ$_8$-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