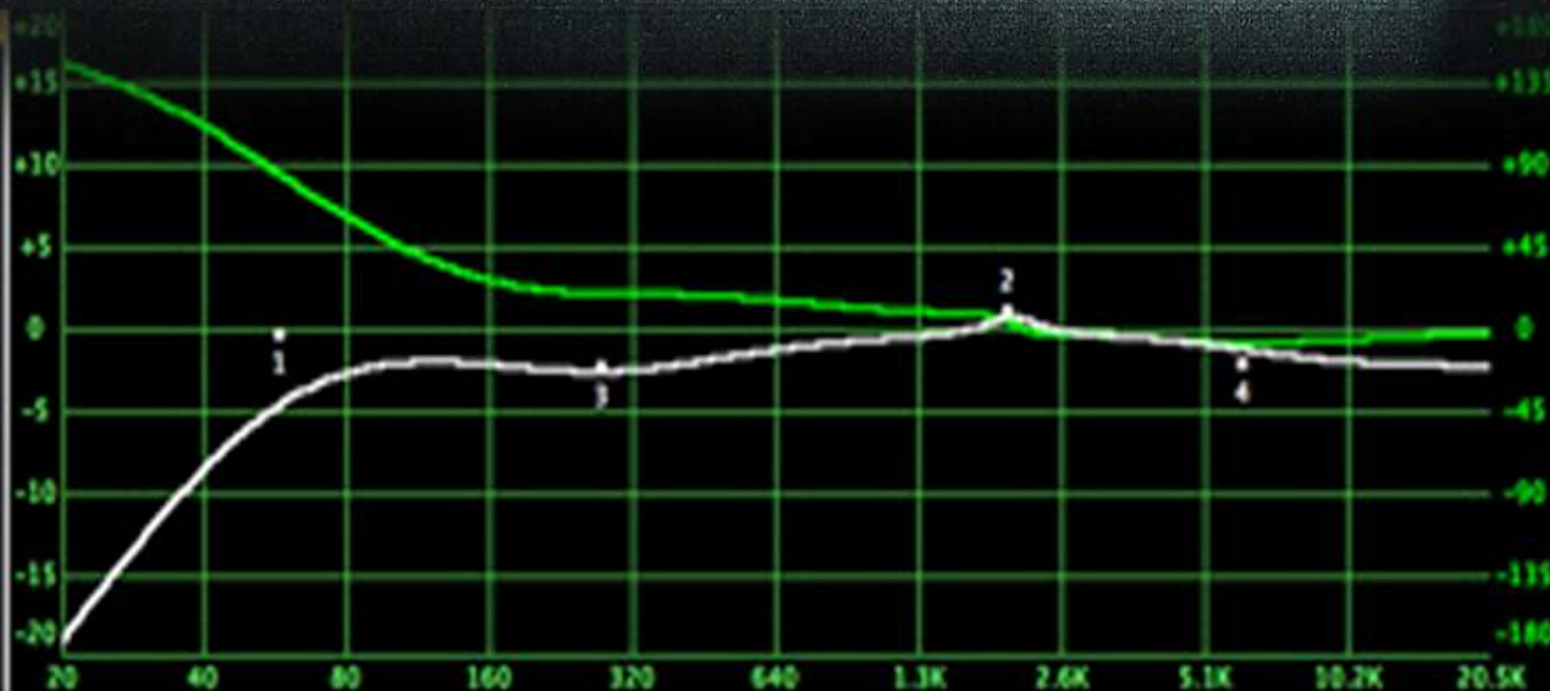


# Signal Processing: An International Journal (SPIJ)

ISSN : 1985-2339

VOLUME 3, ISSUE 6

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR



# **Signal Processing: An International Journal (SPIJ)**

**Volume 3, Issue 6, 2010**

**Edited By**  
**Computer Science Journals**  
[www.cscjournals.org](http://www.cscjournals.org)

**Editor in Chief Dr. Saif alZahir**

## **Signal Processing: An International Journal (SPIJ)**

Book: 2010 Volume 3 Issue6

Publishing Date: 31-01-2010

Proceedings

ISSN (Online): 1985-2339

This work is subjected to copyright. All rights are reserved whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication of parts thereof is permitted only under the provision of the copyright law 1965, in its current version, and permission of use must always be obtained from CSC Publishers. Violations are liable to prosecution under the copyright law.

SPIJ Journal is a part of CSC Publishers

<http://www.cscjournals.org>

© SPIJ Journal

Published in Malaysia

Typesetting: Camera-ready by author, data conversion by CSC Publishing Services – CSC Journals, Malaysia

**CSC Publishers**

## Editorial Preface

This is six issue of volume three of the Signal Processing: An International Journal (SPIJ). SPIJ is an International refereed journal for publication of current research in signal processing technologies. SPIJ publishes research papers dealing primarily with the technological aspects of signal processing (analogue and digital) in new and emerging technologies. Publications of SPIJ are beneficial for researchers, academics, scholars, advanced students, practitioners, and those seeking an update on current experience, state of the art research theories and future prospects in relation to computer science in general but specific to computer security studies. Some important topics covers by SPIJ are Signal Filtering, Signal Processing Systems, Signal Processing Technology and Signal Theory etc.

This journal publishes new dissertations and state of the art research to target its readership that not only includes researchers, industrialists and scientist but also advanced students and practitioners. The aim of SPIJ is to publish research which is not only technically proficient, but contains innovation or information for our international readers. In order to position SPIJ as one of the top International journal in signal processing, a group of highly valuable and senior International scholars are serving its Editorial Board who ensures that each issue must publish qualitative research articles from International research communities relevant to signal processing fields.

SPIJ editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for SPIJ. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. SPIJ provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

## **Editorial Board Members**

Signal Processing: An International Journal (SPIJ)

# Editorial Board

## Editor-in-Chief (EiC)

**Dr. Saif alZahir**  
*University of N. British Columbia (Canada)*

## Associate Editors (AEiCs)

**Professor. Raj Senani**  
*Netaji Subhas Institute of Technology (India)*

**Professor. Herb Kunze**  
*University of Guelph (Canada)*

**Professor. Wilmar Hernandez**  
*Universidad Politecnica de Madrid (Spain )*

## Editorial Board Members (EBMs)

**Dr. Thomas Yang**  
*Embry-Riddle Aeronautical University (United States of America)*

**Dr. Jan Jurjens**  
*University Dortmund (Germany)*

**Dr. Teng Li Lynn**  
*The Chinese University of Hong Kong (Hong Kong)*

**Dr. Jyoti Singhai**  
*Maulana Azad National institute of Technology (India)*

# Table of Contents

Volume 3, Issue 6, January 2010.

## Pages

- 172 - 179 Switched Multistage Vector Quantizer  
**M Satya Sai Ram, Dr.P.Siddaiah, Dr.M.Madhavi Latha**
- 180 - 190 Vehicle Noise Pattern Recognition by Self-Organizing Maps  
**Stefan Twieg, Holger Opfer, Helmut Beikirch**

## Switched Multistage Vector Quantizer

**M. Satya Sai Ram**

*Chalapathi Institute of Technology  
Guntur, AP, INDIA*

m\_satyasairam@yahoo.co.in

**P. Siddaiah**

*K.L.College of Engineering  
Guntur, AP, INDIA*

siddaiah\_p@yahoo.com

**M. Madhavi Latha**

*J.N.T.U College of Engineering  
Hyderabad, A.P, INDIA*

mlmakkena@yahoo.com

---

### Abstract

This paper investigates the use of a new hybrid vector quantizer called switched multistage vector quantization (SWMSVQ) technique using hard and soft decision schemes, for coding of narrow band speech signals. This technique is a hybrid of switch vector quantization technique and multistage vector quantization technique. SWMSVQ quantizes the linear predictive coefficients (LPC) in terms of the line spectral frequencies (LSF). The spectral distortion performance, computational complexity and memory requirements of SWMSVQ using hard and soft decision schemes are compared with split vector quantization (SVQ) technique, multistage vector quantization (MSVQ) technique, switched split vector quantization (SSVQ) technique using hard decision scheme, and multi switched split Vector quantization (MSSVQ) technique using hard decision scheme. From results it is proved that SWMSVQ using soft decision scheme is having less spectral distortion, computational complexity and memory requirements when compared to SVQ, MSVQ, SSVQ and SWMSVQ using hard decision scheme, but high when compared to MSSVQ using hard decision scheme. So from results it is proved that SWMSVQ using soft decision scheme is better when compared to SVQ, MSVQ, SSVQ and SWMSVQ using hard decision schemes in terms of spectral distortion, computational complexity and memory requirements but is having greater spectral distortion, computational complexity and memory requirements when compared to MSSVQ using hard decision.

**Keywords:** Linear predictive coding, Hybrid vector quantizers, Product code vector quantizers, Code book generation.

---

### 1. INTRODUCTION

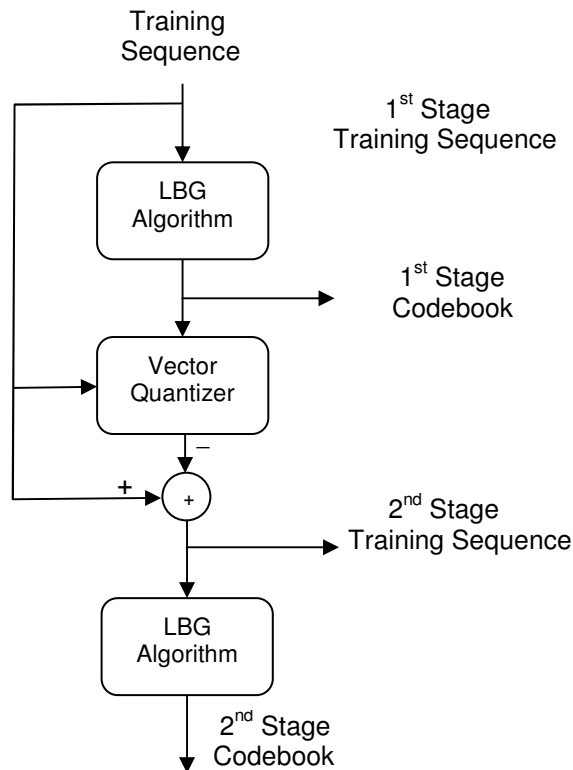
Speech coding is a means of compressing digitized speech signal for efficient storage and transmission, while maintaining reasonable level of quality. Most speech coding systems were designed to support telecommunication applications, with frequency contents band limited between 300 and 3400Hz, i.e., frequency range of narrow band speech coding. In telecommunication applications speech is usually transmitted at 64 kbps, with 8 bits/sample and with a sampling rate of 8 KHz. Any bit-rate below 64 kbps is considered as compression. This



paper deals with switched multistage vector quantizer, it is used in linear predictive coding (LPC) for quantization of line spectral frequencies (LSF) [1-2]. This technique is a hybrid of switch vector quantization technique and multistage vector quantization technique [3-7]. Switched multistage vector quantization is a lossy compression technique. As quality, complexity and memory requirements of a product affects the marketability and the cost of the product or services. The performance of SWMSVQ is measured in terms of spectral distortion, computational complexity and memory requirements. Switched multistage vector quantizer involves the connection of the vector quantizers in cascade where the difference between the input vector and quantized vector of one stage is given as an input to the next stage, each stage of the vector quantizer consists of codebooks connected in parallel. In this work two codebooks are connected in parallel so as to maintain a tradeoff between the switch bits and the number of codebooks to be searched at each stage of the quantizer i.e., when only two codebooks are connected in parallel with soft decision scheme the input vector is quantized by using the two codebooks connected in parallel, with hard decision scheme the input vector is quantized in only one of the two codebooks connected in parallel [7]. As only one bit is required for the two switches in both the soft and hard decision schemes, there can be an improvement in the spectral distortion performance with soft decision scheme when compared to the hard decision scheme. The aim of this article is to examine the performance of switched multistage vector quantizer and to compare its performance with other product code vector quantization schemes like split vector quantization [7-8], multistage vector quantization, switched split vector quantization and multi switched split vector quantization.

## 2. SWITCHED MULTISTAGE VECTOR QUANTIZER

The generation of the codebooks at different stages of switched multistage vector quantizer is shown in Figure 1.



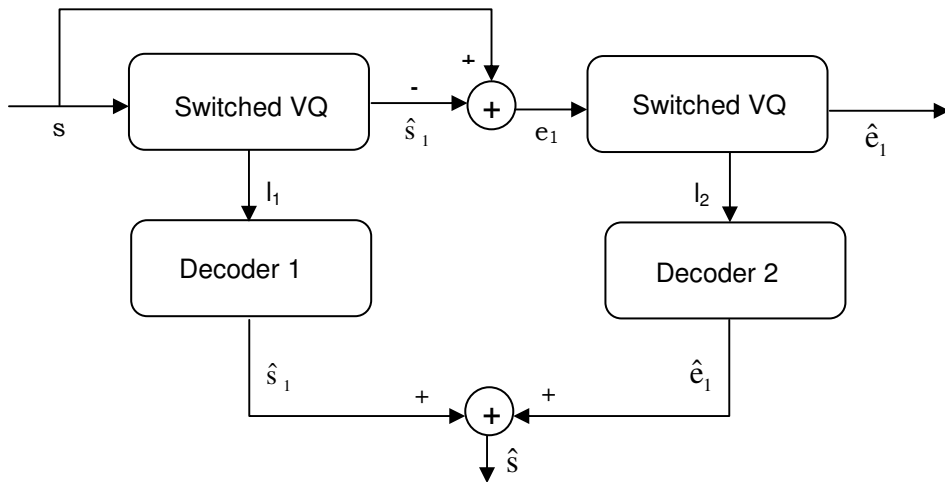
**Figure1.** Codebook Generation at different stages

The basic idea of switched multistage vector quantizer is to use n stages and m switches. With m switches at each stage of the vector quantizer, the performance of quantization has been improved by decreasing the computational complexity and memory requirements. For a particular switch the generation of codebooks at different stages is shown in Figure 1.

- Initially the codebook at the first stage is generated by using the Linde, Buzo and Gray (LBG) algorithm [9] by using training set of line spectral frequencies (LSF) as an input [10-12].
- Secondly the difference vectors at the input of the second stage are generated by using the training sets of the first stage and the codebooks generated at the first stage of the quantizer.
- The training difference vectors at the input of the second stage are used to generate the codebooks of the second stage.
- The above process can be continued for the required number of stages for generating the codebooks.

An n x m Switched Multistage vector quantizer is shown in Figure 2. Where n corresponds to the number of stages, m corresponds to the number of switches. SWMSVQ involves the following steps

- Each vector 's' to be quantized is switched from one codebook to the other connected in parallel at the first stage of the quantizer so as to obtain the quantized vectors.



( $l_i$  denotes the index of  $i^{\text{th}}$  quantizer)

**Figure2.** Block Diagram of Switched Multistage Vector Quantizer

- Extract the quantized vector with minimum distortion from the set of quantized vectors at the first stage i.e.  $\hat{s}_1 = Q[s]$ .
- Compute the quantization error resulting at the first stage of quantization and let the error be  $e_1 = s - \hat{s}_1$ .
- The error vector at the first stage is given as an input to the second stage so as to obtain the quantized version of the error vector i.e.  $\hat{e}_1 = Q[e_1]$ .

This process can be continued for the required number of stages. Finally the quantized vectors from each stage are added up and the resulting vector will be the quantized version of the input vector 's' given by  $\hat{s} = Q[s] + Q[e_1] + Q[e_2] + \dots$ . Where  $Q[s]$  is the quantized version of the input

vector at the first stage,  $Q [e_1]$  is the quantized version of the error vector at the second stage and  $Q [e_2]$  is the quantized version of the error vector at the third stage and so on.

### 3. SPECTRAL DISTORTION

In order to objectively measure the distortion between the quantized and unquantized outputs a method called the 'spectral distortion' is often used in narrow band speech coding. For an  $i^{\text{th}}$  frame the spectral distortion,  $SD_i$  (in dB) between the quantized and unquantized vectors is given by [13]

$$SD_i = \sqrt{\frac{1}{(f_2 - f_1)} \int_{f_1}^{f_2} [10 \log_{10} S_i(f) - 10 \log_{10} \hat{S}_i(f)]^2 df} \text{ (dB)} \quad (1)$$

Where  $S_i(f)$  and  $\hat{S}_i(f)$  are the LPC power spectra of the unquantized and quantized  $i^{\text{th}}$  frame respectively. The frequency  $f$  is in Hz, and the frequency range is given by  $f_1$  and  $f_2$ . The frequency range used in practice for narrow band speech coding is 0-4000Hz. The average spectral distortion  $SD$  [13] is given by

$$SD = \frac{1}{N} \sum_{n=1}^N SD_i \quad (2)$$

Frames having spectral distortion greater than 1dB are called as outlier frames. The conditions for transparent speech coding are.

- The mean of the spectral distortion (SD) must be less than or equal to 1dB.
- The number of outlier frames having spectral distortion greater than 4dB must be zero.
- The number of outlier frames having spectral distortion between 2 to 4dB must be less than 2%.

### 4. COMPLEXITY AND MEMORY REQUIREMENTS

The computational complexity of a switched multistage vector quantizer using hard decision scheme is given by

$$\text{Complexity}_{\text{SWMSVQ HARD}} = \sum_{j=1}^P \left( 4n \left( 2^{b_m} + 2^{b_j} \right) - 2 \right) \quad (3)$$

The computational complexity of a switched multistage vector quantizer using soft decision scheme is given by

$$\text{Complexity}_{\text{SWMSVQ SOFT}} = \sum_{j=1}^P \left( \left( 4n 2^{b_m} - 1 \right) + \left( \sum_{k=1}^{P_l} 4n 2^{b_{jk}} - 1 \right) \right) \quad (4)$$

The memory requirements of a switched multistage vector quantizer using hard decision scheme is given by

$$\text{Memory}_{\text{SWMSVQ HARD}} = \sum_{j=1}^P n \left( 2^{b_m} + 2^{b_j} \right) \quad (5)$$

The memory requirements of a switched multistage vector quantizer using soft decision scheme is

$$\text{Memory}_{\text{SWMSVQ SOFT}} = \sum_{j=1}^P \left( \left( n2^{b_m} \right) + \left( \sum_{k=1}^{P_1} n2^{b_{j k}} \right) \right) \quad (6)$$

Where

$n$  is the dimension of the vector

$b_m$  is the number of bits allocated to the switch vector quantizer

$b_j$  is the number of bits allocated to the  $j^{\text{th}}$  stage

$m = 2^{b_m}$  is the number of switching directions

$P$  is the number of stages

$b_{jk}$  is the number of bits allocated to the  $j^{\text{th}}$  stage  $k^{\text{th}}$  codebook

$P_1$  is the number of codebooks connected in parallel at each stage

## 5. RESULTS AND DISCUSSION

Table1. Spectral distortion, Complexity, and Memory requirements for 3-Part Split vector quantization technique

Bits / frame	SD(dB)	2-4 dB	>4dB	Complexity (kflops/frame)	ROM (floats)
24	1.45	0.43	0	10.237	2560
23	1.67	0.94	0	8.701	2176
22	1.701	0.78	0.1	7.165	1792
21	1.831	2.46	0.2	5.117	1280

Table2: Spectral distortion, Complexity, and Memory requirements for 3-stage multistage vector quantization technique

Bits / frame	SD(dB)	2-4 dB	>4dB	Complexity (kflops/frame)	ROM (floats)
24	0.984	1.38	0	30.717	7680
23	1.238	1.2	0.1	25.597	6400
22	1.345	0.85	0.13	20.477	5120
21	1.4	1.08	0.3	15.357	3840

Table3: Spectral distortion, Complexity, and Memory requirements for 2- switch 3-part switched split vector quantization technique using hard decision scheme

Bits / frame	SD(dB)	2-4 dB	>4dB	Complexity (kflops/frame)	ROM (floats)
24	0.957	1.06	0	8.78	4372
23	1.113	1.29	0.14	7.244	3604
22	1.119	0.52	1.3	5.196	2580
21	1.127	1.3	0.56	4.428	2196

**Table4:** Spectral distortion, Complexity, and Memory requirements for 2- switch 3-stage switched multistage vector quantization technique using hard decision scheme

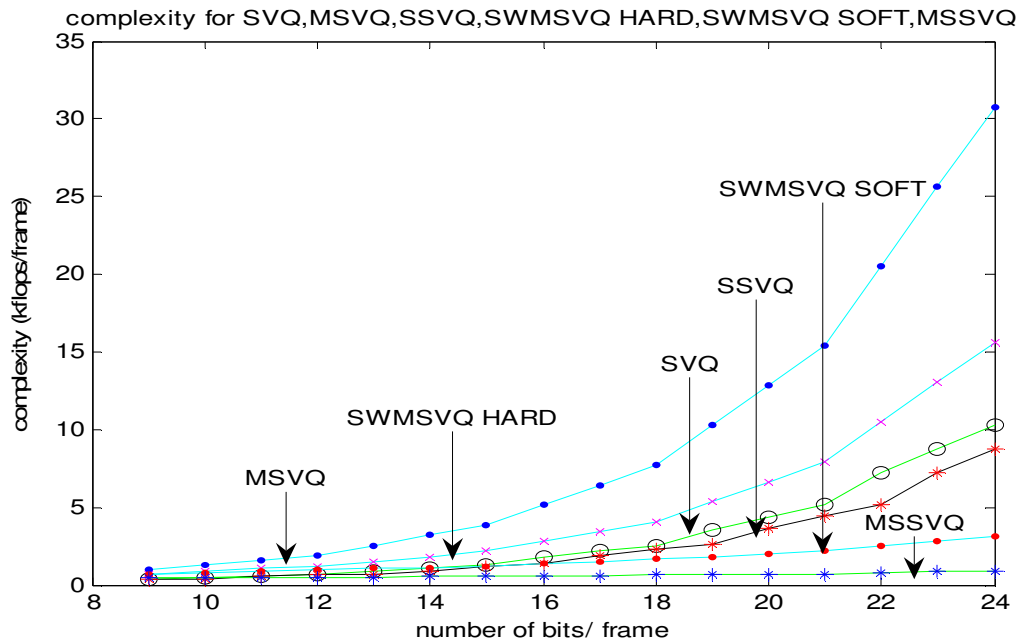
Bits / frame	SD(dB)	2-4 dB	>4dB	Complexity (kflops/frame)	ROM (floats)
24	0.93	1.4	0	15.594	3900
23	1.131	0.83	1.12	13.034	3260
22	1.134	0.42	1.56	10.474	2620
21	1.163	1.16	0.38	7.914	1980

**Table5:** Spectral distortion, Complexity, and Memory requirements for 2- switch 3-stage switched multistage vector quantization technique using soft decision scheme

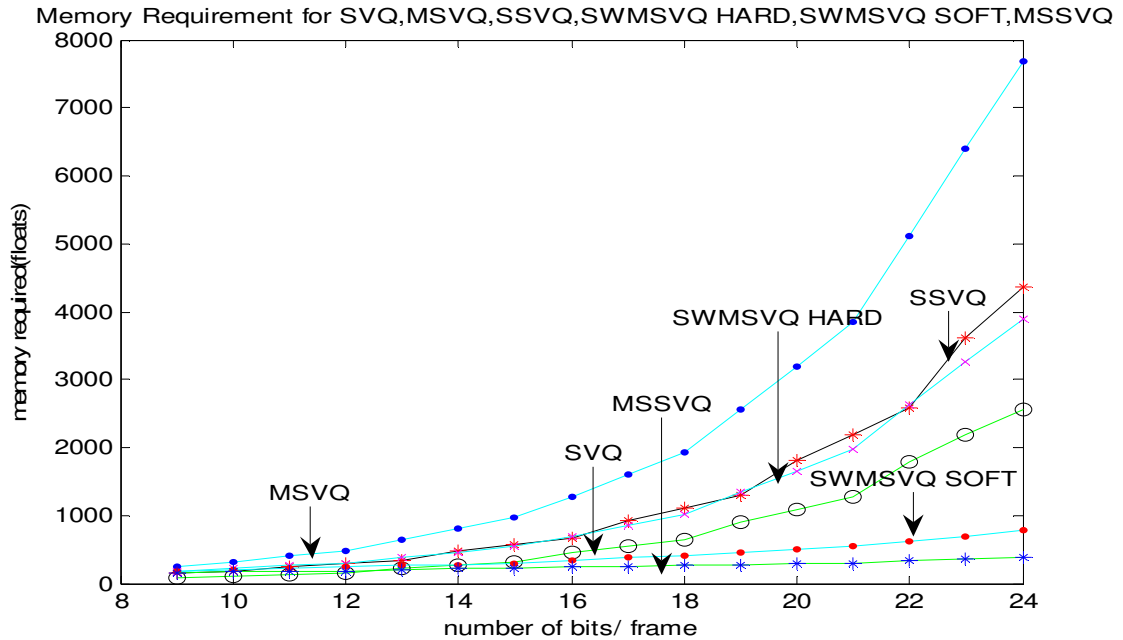
Bits / frame	SD(dB)	2-4 dB	>4dB	Complexity (kflops/frame)	ROM (floats)
24	0.91	0.56	0.81	3.111	780
23	0.87	1.05	0.31	2.791	700
22	1.1	1.45	0.63	2.471	620
21	1.18	0.6	1.89	2.151	540

**Table6:** Spectral distortion, Complexity, and Memory requirements for a 3-stage 2-switch 3-part multi switched split vector quantization technique using hard decision scheme

Bits / frame	SD(dB)	2-4 dB	>4dB	Complexity (kflops/frame)	ROM (floats)
24	0.0322	0	0	0.9	396
23	0.0381	0	0	0.836	364
22	0.0373	0	0	0.772	332
21	0.0377	0	0	0.708	300



**Figure.3.** Complexity for SVQ, MSVQ, SSVQ Hard, SWMSVQ Hard, SWMSVQ Soft and MSSVQ Hard



**Figure.4.** Memory requirements for SVQ, MSVQ, SSVQ Hard, SWMSVQ Hard, SWMSVQ Soft and MSSVQ Hard

Tables 1 to 4 shows the spectral distortion measured in dB, computational complexity measured in Kilo flops/frame, and memory requirements (ROM) measured in floats at various bit-rates for a 3-part split vector quantizer, 3-stage multistage vector quantizer, 2-switch 3-part switched split vector quantizer using hard decision scheme, 2-switch 3-stage switched multistage vector quantizer using hard and soft decision schemes and 3-stage 2-switch 3-part multi switched split vector quantizer using hard decision scheme. From Tables 1 to 4 and from Figures 3 & 4 it can be observed that SWMSVQ using soft decision scheme has less spectral distortion, computational complexity and memory requirements when compared to SVQ, MSVQ, SSVQ using hard decision scheme, and SWMSVQ using hard decision scheme but with a slight increase in spectral distortion, computational complexity and memory requirements when compared to MSSVQ using hard decision scheme. For SWMSVQ using hard decision scheme the spectral distortion is less when compared to SVQ, MSVQ, and SSVQ using hard decision scheme but with a slight increase in spectral distortion when compared to SWMSVQ using soft decision scheme and MSSVQ using hard decision scheme. The computational complexity of SWMSVQ using hard decision scheme is less when compared to MSVQ, and the memory requirements are less when compared to MSVQ and is in a comparable manner with SSVQ using hard decision scheme.

## 6. CONCLUSION

From results it is proved that SWMSVQ using soft decision scheme provides better trade-off between bit-rate and spectral distortion, computational complexity, and memory requirements when compared to all the product code vector quantization techniques except for MSSVQ using hard decision scheme. So SWMSVQ using soft decision scheme is proved to be better when compared to SVQ, MSVQ, SSVQ and SWMSVQ using hard decision scheme, and is having comparable performance when compared to MSSVQ using hard decision. The advantage with soft decision scheme is, with increase in the number of stages or codebooks per stage or splits per codebook the number of available bits at each stage, codebook, split gets decreased there by the computational complexity and memory requirements gets decreased, but the disadvantage is that there will be a limit on the number of stages, codebooks per stage and on the number of splits.

## 7. REFERENCES

1. Atal, B.S, "The history of linear prediction", IEEE Signal Processing Magazine, Vol. 23, pp.154-161, March 2006.
2. Harma, A. "Linear predictive coding with modified filter structures", IEEE Trans. Speech Audio Process, Vol. 9, pp.769-777, Nov 2001.
3. Gray,R.M., Neuhoff, D.L.. "Quantization", IEEE Trans. Inform. Theory, pp.2325-2383, 1998.
4. M.Satya Sai Ram., P.Siddaiah., M.MadhaviLatha, "Switched Multi Stage Vector Quantization Using Soft Decision Scheme", IPCV 2008, World Comp 2008, Las vegas, Nevada, USA, July 2008.
5. M.Satya Sai Ram., P.Siddaiah., M.MadhaviLatha, "Multi Switched Split Vector Quantization of Narrow Band Speech Signals", Proceedings World Academy of Science, Engineering and Technology, WASET, Vol.27, pp.236-239, February 2008.
6. M.Satya Sai Ram., P.Siddaiah., M.MadhaviLatha, "Multi Switched Split Vector Quantizer ", International Journal of Computer, Information, and Systems science, and Engineering, IJCISSE, WASET, Vol.2, no.1, pp.1-6, May 2008.
7. Stephen, So., & Paliwal, K. K, "Efficient product code vector quantization using switched split vector quantiser", Digital Signal Processing journal, Elsevier, Vol 17, pp.138-171, Jan 2007.
8. Paliwal., K.K, Atal, B.S, "Efficient vector quantization of LPC Parameters at 24 bits/frame", IEEE Trans. Speech Audio Process, pp.3-14, 1993.
9. Linde ,Y., Buzo , A., & Gray, R.M, "An Algorithm for Vector Quantizer Design", IEEE Trans.Commun, Vol 28, pp. 84-95, Jan.1980.
10. Bastiaan Kleijn., W. Fellow, IEEE, Tom Backstrom., & Paavo Alku, "On Line Spectral Frequencies", IEEE Signal Processing Letters, Vol.10, no.3, 2003.
11. Soong, F., & Juang, B, "Line spectrum pair (LSP) and speech data compression", IEEE International Conference on ICASSP, Vol 9, pp 37- 40, 1984.
12. P.Kabal and P. Rama Chandran, "The Computation of Line Spectral Frequencies Using Chebyshev polynomials" - IEEE Trans. On Acoustics, Speech Signal Processing, vol 34, no.6, pp. 1419-1426, 1986.
13. Sara Grassi., "Optimized Implementation of Speech Processing Algorithms", Electronics and Signal Processing Laboratory, Institute of Micro Technology, University of Neuchatel, Breguet2,CH2000 Neuchatel, Switzerland, 1988.
14. M.Satya Sai Ram, P. Siddaiah, and M.Madhavi Latha, "Usefullness of Speech Coding in Voice Banking," Signal Processing: An International Journal, CSC Journals, Vol 3, Issue 4, pp 37- 40, pp. 42-54, Oct 2009.

## Vehicle noise pattern recognition by Self-Organizing Maps

**Stefan Twieg**

*Volkswagen AG/Group Research  
Wolfsburg, Germany*

stefan.twieg@volkswagen.de

**Holger Opfer**

*Volkswagen Group Japan  
VTT, Tokyo, Japan*

opfer.holger@vgj.co.jp

**Helmut Beikirch**

*Faculty of Computer Science and  
Electrical Engineering  
University Rostock, Germany*

helmut.beikirch@uni-rostock.de

---

### Abstract

Interior vehicle acoustics are in close connection with our quality opinion. The noise in vehicle interior is complex and can be considered as a sum of various sound emission sources. A nice sounding vehicle is objective of the development departments for car acoustics. In the process of manufacturing the targets for a qualitatively high-valuable sound must be maintained. However, it is possible that production errors lead to a deviation from the wanted vehicle interior sound. This will result in customer complaints where for example a rattling or squeak refers to a worn-out or defective component. Also in this case, of course, the vehicle interior noise does not fulfill the targets of the process of development. For both cases there is currently no possibility for automated analysis of the vehicle interior noise. In this paper an approach for automated analysis of vehicle interior noise by means of neural algorithms is investigated. The presented system analyses microphone signals from car interior measured at real environmental conditions. This is in contrast to well known techniques, as e.g. acoustic engine test bench. Self-Organizing Maps combined with feature selection algorithms are used for acoustic pattern recognition. The presented system can be used in production process as well as a standalone ECU in car.

**Keywords:** signal classification, SOM, car noise, pattern recognition.

---

### 1. MOTIVATION

Car interior noise is a sum of different sound emission sources, e.g. engine, wind, chassis or suspension. Furthermore, it is a challenging task to examine objective assessment criteria for a sound. In production process acoustic and vibration technology is used e.g. for analysis at engine test bench. Engines and test benches are tested in [1] to recognize damages of failure parts. Suppliers are also checking their parts before delivery by similar methods [2], [3] to achieve a reliable and correct product. Commonly, the acoustic analysis is done in defined environmental conditions to get no interference with unknown noises. In this case quite simple mathematic is



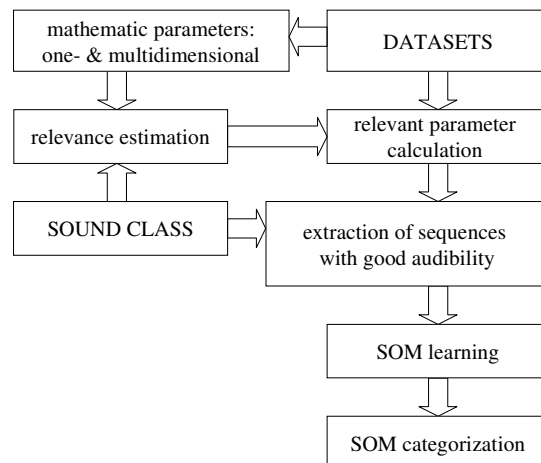
used to detect and to interpret changes. This method is in contrast to the method shown in this paper where an undefined environment influences the acoustic conditions. For assessment of car interior sound by algorithms the time signal which is recorded by microphone in car interior has to be fragmented in different mathematic parameters. The challenge is to find parameters which are corresponding to failures or changes of car part sounds and not to acoustic changes in environment. Thus the parameters have to be explicit relevant for sounds of interest. To analyze these relevant parameter knowledge storage is required to build up experience of parameter behavior. Therefore a neural net by Kohonen [4], the self-organizing map (SOM), models an input space where similar patterns can be categorized. For the described purpose the Kohonen-map is combined with parameter relevance detection algorithm. As a result the developed system assesses continuously the microphone signals, e.g. in two categories like “error-free” and “error” sound. In future the described method can be used for applications of sound assessment at production-line end (at test-bench or test-drive) or in car combined with hands-free kit.

## 2. MODEL-LAYOUT FOR PATTERN RECOGNITION

An adequate amount of data sets is the basis for acoustic pattern recognition. For this purpose different noticeable sounds are stored in database. They were recorded at different conditions like:

- Car type,
- Driving situation (longitudinal and transversal dynamic),
- Car parts (new vs. used and damaged),
- Environment (weather, pavement).

In the following analysis one car type was chosen. For this one 1430 measurements with a length of 15 to 30 seconds are stored in database. In Figure 1, you can see the model for datasets analysis and pattern recognition. As next step the one (e.g. rms, sound pressure level) or multidimensional (e.g. spectrum, DCT, MFCC) parameters are calculated. However, psycho-acoustic (e.g. loudness, roughness) parameters are calculated too. In the next step the focal point is



**FIGURE 1:** Model-design for classification of acoustic in car recordings.

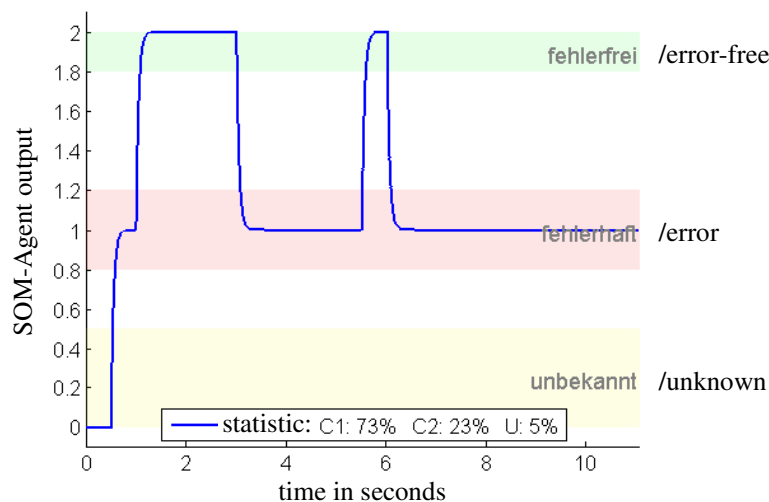
relevance detection to identify parameters which are corresponding to noticeable sounds caused by wear (e.g. steering rod ball joint) or damage. The relevance detection algorithm has to identify the connection between:

- the sound classes (e.g. “error-free” and “error” sounds),
- sound classes and parameters.

Each parameter is rated and ordered by rank information. The parameters with best relevance are used as input features for SOM. In the next step the datasets with best audibility of defined sound classes are used for training process. There are two defined sound classes “error” and

“error-free” sound. For more detailed information it is also possible to define more sub-sound-classes, e.g. different noise emitting parts. The sound class representing sequences are extracted from database, whereas a huge variance of recording situations is very important. After learning process the weights of the neural net are adopted. Now the SOM represents the input space. In labeling process the sound class information is added. Thereafter it is possible to classify unknown test datasets and to get a feedback of associated sound class.

In the following the described method is combined in a so-called SOM-Agent which will monitor and assess microphone signals. The knowledge – the neural weights – of the input space and the sound classes – from labeling process – are transferred to the agent. Relevant parameters are calculated for the sampled audio data and will be used as input features for the SOM-Agent. Identification of sound class active neurons will follow. In Figure 2 the created SOM-Agent characteristic curve for one recording of 11 seconds is shown. The 2 defined sound classes and an “unknown” class – in case of no active neuron – are represented by an output value in range 0 to 2 on y-axis. The x-axis represents time. Statistics are also computed for the three classes.



**FIGURE 2:** Autonomous detection of noticeable acoustic changes in car interior. The SOM-Agent characteristic curve (blue) toggles between three sound classes “error-free”, “error” and “unknown” sound, e.g. if output value is “1” the microphone recording was made in a car with damaged or worn-out parts.

### 3. FUNDAMENTALS

As described in previous section the shown method consists of two main components. First one is identification and calculation of sound class relevant parameters. For the second step these are the input features for a self-organizing map which will categorize them in defined sound classes.

#### 3.1 Relevance Identification

Several methods exist for analyzing and calculation of feature influence, feature relevance, feature (subset) selection or feature extraction. Reduction of feature space to the essential is main target of several classification tasks. Anyway, the connectivity between the parameters and classes is determined. Therefore the correlation coefficient [5] – as linear method – or the information gain [6], [7] – as non-linear method – can be used. In Equation 1 the calculation rule for correlation coefficient, also known as normalized covariance is given. Thus, the linear connection between the vectorial values  $x$  and  $y$  is estimated.

$$r(x, y) = \frac{n \sum xy - (\sum x)(\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2} \sqrt{n(\sum y^2) - (\sum y)^2}} \quad (1)$$

On the other hand the information gain ( $IG$ , Equation 2 at [6]) is calculated by entropy  $H$  and represents a non-linear probabilistic method. The information gain returns the contribution of one attribute to class decision making.

$$IG(S) = H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v) \quad (2)$$

Whereas  $S$  is the set and  $A$  represents the class information of  $S$ .  $S_v$  is the subset of the selected class.

Further known filter methods (parameter selection by pre-processing) are for example: Mutual Information, Fisher Discriminant, RELIEF, PCA, ICA, FCBF (Fast Correlation Based Filtering, [8], [9]). As you can see in [8], [9] the FCBF algorithm solves the two main aspects of feature subset selection. First one is in our case the decision whether mathematic parameter (feature) is relevant or not for a acoustic condition (sound class). In second step the algorithm determines whether a relevant parameter is redundant.

The symmetrical uncertainty (SU, Equation 3) is used for determination of nonlinear connection between the two sources  $x$  and  $y$ . It is based on information gain but it is normalized on interval [0,1].

$$SU(x, y) = 2 \left[ \frac{IG(x|y)}{H(x) + H(y)} \right] \quad (3)$$

In correlation analysis for a dataset the relevance and redundancy are assessed by this formula. If we are assuming that the values of one source contain same information like second source the SU output will be 1. If SU output is 0 the sources are independent of each other. The symmetrical uncertainty is used in FCBF algorithm as measure for correlation between two sources. In this algorithm two main aspects are shown.

First one is the SU calculation for the feature-class-SU:

- Serves as measure for correlation between each feature and the class information.
- For each feature the SU to all classes is calculated and stored in a list.
- For this list a threshold determines if features SU is too low. A feature is deleted if threshold is undercut.
- For each class the residual features are ordered in descending order and stored in a final list.

In second process the redundancy examination occurs by feature-feature-SU.

- Serves as a measure for correlation between each combination of the listed features.
- For each feature the SU is calculated to all other features.
- Comparison of feature-feature-SU and feature-class-SU.
- Features with very high SU are deleted.
- The remaining features are representing the best features in dataset.

In most cases the efficiency of the algorithm is shown by a classifier. For this case the recognition rate is the measure of assess. Anyway, as well known no algorithm will be the best or worst solution for all datasets. In most cases algorithms are optimized for all-round use or for dataset. As you can see in analysis (Figure 3, [9]) of UCI library [10] datasets the FCBF algorithm has best performance for *CoIL2000* (compared to three reduction methods). For *Lung-Cancer* and *Splice* dataset on the contrary all methods have nearly same recognition rates.

Title	FCBF <sub>(log)</sub>		FCBF <sub>(0)</sub>		Full Set		ReliefF		CFS-SF		FOCUS-SF	
	Acc	Acc	p-Val	Acc	p-Val	Acc	p-Val	Acc	p-Val	Acc	p-Val	
Lung-cancer	83.33	86.67	0.34	78.33	0.34	84.17	0.85	86.67	0.34	87.5	0.46	
Promoters	93.27	93.27	1	91.55	0.55	87.82	0.25	95.18	0.17	90.45	0.40	
Splice	93.95	96.14	<b>0.00</b> <sup>+</sup>	95.52	<b>0.00</b> <sup>+</sup>	91.32	<b>0.00</b> <sup>-</sup>	93.54	0.24	94.36	<b>0.08</b> <sup>+</sup>	
USCensus90	97.94	97.88	0.19	93.49	<b>0.00</b> <sup>-</sup>	97.97	0.17	97.99	0.65	97.87	0.44	
CoIL2000	93.94	93.92	0.34	78.68	<b>0.00</b> <sup>-</sup>	93.89	0.66	92.92	<b>0.01</b> <sup>-</sup>	83.22	<b>0.00</b> <sup>-</sup>	
Chemical	71.91	67.73	<b>0.02</b> <sup>-</sup>	60.90	<b>0.00</b> <sup>-</sup>	71.26	0.77	70.51	0.35	66.35	<b>0.00</b> <sup>-</sup>	
Musk2	84.59	84.59	1	84.78	0.51	84.59	1	64.87	<b>0.00</b> <sup>-</sup>	83.53	<b>0.01</b> <sup>-</sup>	
Arrhythmia	67.48	65.73	0.45	60.88	<b>0.01</b> <sup>-</sup>	55.79	<b>0.00</b> <sup>-</sup>	69.05	0.45	69.06	0.56	
Isolet	50.06	83.33	<b>0.00</b> <sup>+</sup>	84.10	<b>0.00</b> <sup>+</sup>	60.90	<b>0.00</b> <sup>+</sup>	87.31	<b>0.00</b> <sup>+</sup>	71.03	<b>0.00</b> <sup>+</sup>	
Multi-feat	95.9	95.65	0.50	94.1	<b>0.01</b> <sup>-</sup>	67.65	<b>0.00</b> <sup>-</sup>	96.15	0.64	93.7	<b>0.02</b> <sup>-</sup>	
L/W/T	-	1/2/7		5/2/3		3/1/6		2/1/7		4/2/4		

**Figure 3:** Comparison of recognition rates of FCBF, ReliefF, CFS-SF and FOCUS-SF algorithm. ([9], “Acc records 10-fold cross-validation accuracy rate (%) and p-Val records the probability associated with a paired two-tailed t-Test.”).

### 3.2 Self-Organizing-Map

The neurons of a self organizing map [4] are represented by a multidimensional weight vector. If the neural net is adopted the weight vectors are already adjusted (self organized without supervise) to the input space (defined by the features/parameters). Field of application is the recognition of clusters in high dimensional data. Each input node has a connection by weights to each neuron of the map. One way of visualization is the component plane where the weights are shown as contour or surface plot. However, more detailed information for SOM-visualization (e.g. U-Matrix) methods is discussed in [11].

Representation of the  $n$  SOM-neurons  $w_1, \dots, w_n$  is done by the weight vectors  $\vec{w}_1, \dots, \vec{w}_n$ . The number of elements in  $\vec{w}$  is equal to the number  $m$  of input features  $x_1, \dots, x_m$ . The initialization of the weights is done randomly or by prior input space knowledge. Adjustment and allocation of neurons to the input space is made in different ways, e.g. as line, two- or three-dimensional map. The adaptation is done by a learning rule with defined step size. The interconnected neurons are adjusted by means of following equation:

$$\vec{w}_k(n+1) = \vec{w}_k(n) + \alpha \cdot (\vec{x}(n) - \vec{w}_k(n)) \quad (4)$$

In Equation 4 the calculation rule (from [4]) for adaptation of weight vectors  $\vec{w}_k$  for selected neuron  $k$  is shown. The step size  $\alpha$  is chosen in interval  $0 < \alpha < 1$ . There is no general rule for determination of  $\alpha$ . If *batch learning* is used the input vector  $\vec{x}$  with  $m$  features is randomly selected out of dataset. If we have a closer look to the learning process the adjustment of the weights results in a movement of the neuron in the direction of the training vector. The selection of the trained neuron occurs by a distance function, e.g. Euclid distance. The neuron which has lowest distance to input vector  $\vec{x}$  is adopted. However, the algorithm has to be updated in the way that the connected neurons are covering the complete input space to get a map which is topology preservative. For this purpose a neighborhood function  $\varphi$  defines a radius for the selected winner neuron which has lowest distance to input vector. Thus, neurons inside the radius were adopted by the input vector, too. Furthermore a distance function  $d(w_i, w_k)$  allows a division in different neighborhood-distances  $R$  for a weighted neuron adaptation. Symbolically neurons which are closest to the winner neuron have the distance  $R=1$  and neurons in next radius have a distance of  $R=2$  and so on. The degree of adaptation which defines the amount of movement for the winning neuron in the direction of input vector  $\vec{x}$  depends on the neighborhood function  $\varphi(w_i, w_k) \leq R$  with a range of values in  $[0,1]$ . The neurons  $w_i$  with a small distance will be adapted much more than neurons with a big distance. Also the design of the neighborhood function can be varied depending on computational resources.

- Computationally efficient Stair-Neighborhood function

$$\varphi(w_i, w_k) = \begin{cases} \left( \frac{1}{d(w_i, w_k)} \right); & d(w_i, w_k) \leq R \\ 0; & \text{sonst} \end{cases} \quad (5)$$

- Computationally intensive Gaussian-Neighborhood function

$$\varphi(w_i, w_k) = \begin{cases} e^{-\left( \frac{d(w_i, w_k)^2}{2\sigma^2} \right)}; & d(w_i, w_k) \leq R \\ 0; & \text{sonst} \end{cases} \quad (6)$$

In comparison Equation 6 realizes a more detailed and non-linear neighborhood-function than Equation 5. Finally, after insertion in Equation 4 the learning rule is shown updated in Equation 7.

$$\bar{w}_i(n+1) = \bar{w}_i(n) + \alpha \cdot \varphi(w_i, w_k) \cdot (\bar{x}(n) - \bar{w}_i(n)) \quad (7)$$

Commonly the adaptation process is done by a defined amount of training cycles. The adaptation process shown in Equation 4 and 7 can be optimized by a fast adjustment at the beginning and a slow one at the end of training. This behavior can be realized by a time-varying step-size  $\alpha(t)$ .

$$\alpha(t) = \alpha \left( 1 - \frac{t}{T_E} \right) \quad (8)$$

By means of the fixed ratio between actual time  $t$  to the number of cycles  $T_E$  in Equation 8 a linear function is given. Non-linear step-size functions are also possible alternatives. Also it is possible to modify the neighborhood functions (Equation 5 and 6) with a time descending factor. For interpretation of the self-organizing map two main aspects were analyzed:

- Characteristics of SOM-neuron-weights in dependence of input vector.
- Class assignment (sound classes) of neurons with a maximum reaction (smallest distance) at defined stimulation (a.k.a. labeling process).

#### 4. METHOD DESCRIPTION AND EVALUATION

As a basis for the training of the SOM-Agent (see Figure 2) a noise database with different measurements is used from which the mathematical properties are calculated. For training, 11 different noise types were defined which come from worn-out coupling rods, wheel bearings, suspension strut mountings and other running gear components from the front of a defined motor vehicle type. Noises which originate from the normal sound of a series vehicle without worn-out components represent a further noise type. In sum there are 129 measurements - with a length between 10 and 30 seconds - for which finally two training classes are defined:

“C1”: error

“C2”: error-free

The time constant for the window length for which the parameters are calculated and averaged is  $250ms$ . For each of the 129 datasets several parameter-sets are calculated and marked as class one or two. For the dimension reduction and relevance determination for the features the FCBF-algorithm was selected. From the 17 available parameters the parameters shown in Table 1 are determined as relevant by the FCBF-algorithm described in section C1. The training process runs  $T$  cycles, in which a randomly selected parameter-set is presented to the network. In next step all measurements are used for evaluation by SOM-Agent. Therefore all 129 acoustic records are

Roughness (Time)
Speed
Sharpness (Time)
Spectral Centroid
Yaw Rate
Loudness (Time)

**TABLE 1:** The 6 most relevant parameters are estimated by means of FCBF-algorithm.

presented one after the other and the characteristic curve shown in figure 2 is calculated. The parameters must be estimated for the acoustic signals continuously. Thus, these time continuous parameters equal time variable data rows where each column set is an input vector of self-organizing map. Through determination of the active neurons in dependence of the class the acoustic time signals can be assessed and equipped with class information (“C1” or “C2”). For reliable condition recognition a buffer (e.g. ring-buffer with 12000 samples) is filled with these data rows and is analyzed periodically. For this purpose a subset of the stored data in buffer is used to estimate the class. Depending on distribution of active neurons the following output is calculated:

- SOM-Agent output  $s \in N$  in interval  $s = [0,1,2]$ , where
  - $s = 1$  corresponds with error-noise class “C1” and
  - $s = 2$  corresponds with error-free-noise class “C2”.
- SOM-output  $s = 0$  if input data didn't activate any neuron. That's why Zero corresponds to an unknown state “U”.

For visualization a low pass filtered curve is generated from signal  $s$  :

$$s_{LP}(n) = \left(1 - \frac{1}{L}\right) s_{TP}(n-1) + \frac{1}{L} \cdot s(n) \quad (9)$$

A statistical evaluation is done afterwards either for the values of  $s$  or  $s_{LP}$ . In the case of strong fluctuating curves of  $s$  the curve of  $s_{LP}$  will generate an averaged value over the window length characterized by  $L$ . However diffuse values (e.g.  $s = 1.5$ ) are not used for statistic analysis, that's why thresholds are implemented (have a look at colored areas in Figure 2). These thresholds are optimized for a clear assessment between two classes. The probability of occurrence for the classes (in the case of use of  $s_{LP}$  within the threshold regions) is estimated over the length of time signals. In the legend of Figure 2 the probability of occurrence is shown for the analyzed record. As a result class “C1” was determined by SOM-Agent for 70% of analyzed time.

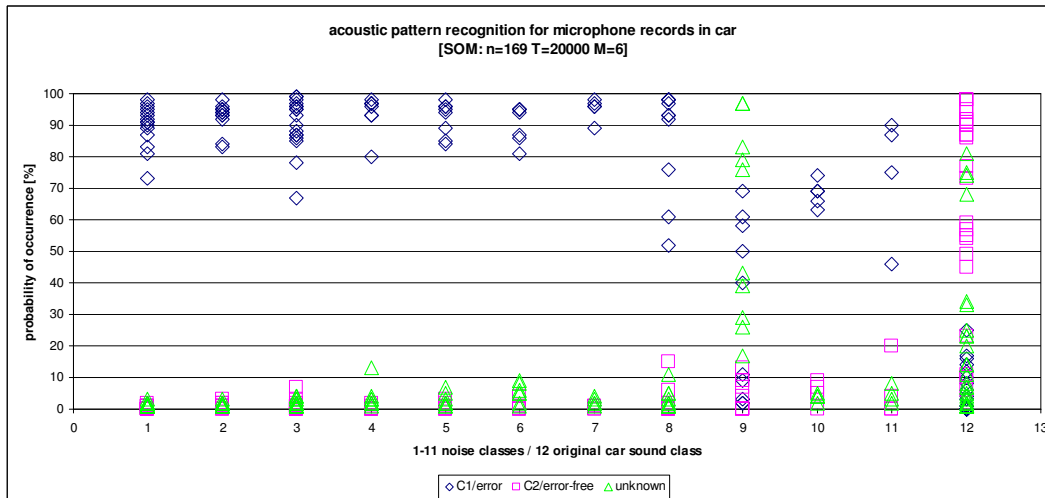
For training process and parameterization of self-organizing map the obviously noticeable sequences are extracted out of 129 measurements. Hence a dataset with 2456 patterns with 17 parameters is generated. The 6 most relevant parameters were presented to 2 different self-organizing maps with 169 or 1000 neurons and 6 inputs. The Gaussian function described in Equation 6 was used as a neighborhood function. Neurons within the neighborhood radius  $R$  (defines the width of the bell curve) are adopted by following rule (out of Equation 7, 8, 9):

$$\bar{w}_i(n+1) = \bar{w}_i(n) + \alpha \left(1 - \frac{t}{T_E}\right) \cdot e^{-\left(\frac{d(w_i, w_k)}{R}\right)^2} \cdot (\bar{x}(n) - \bar{w}_i(n)) \quad (10)$$

As distance function, the Euclidean distance is used:

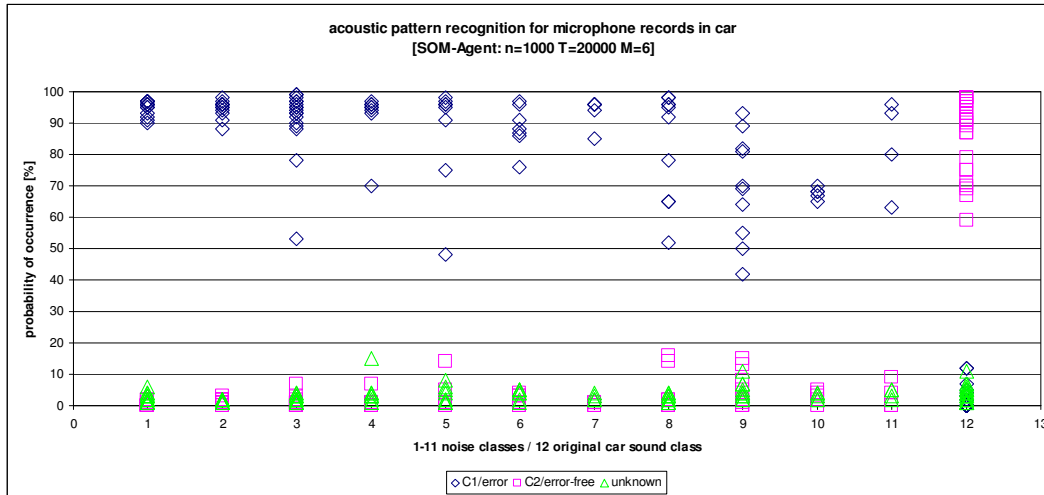
$$d(w_i, w_k) = \|w_i - w_k\| = \sqrt{\sum_{x=1}^N (w_{ix} - w_{kx})^2} \quad (11)$$

In figure 4 the evaluation of the recognition rates is shown for the SOM1-agent (169 neurons). On x-axis the acoustic error conditions (class "C1") in the range 1 to 11 are reproduced. Number 12 is the acoustic original state without any worn-out or damaged part (class "C2"). On y-axis the averaged probabilities of occurrence are reproduced.



**Figure 4:** SOM1-Agent statistic: A calculated probability of occurrence for 129 measurements, SOM with 169 neurons and 6 parameters is used. Legend defines whether the analyzed measurements are assessed as “error”, “error-free” or “unknown” condition.

The results of SOM1-Agent (Figure 4) show that primarily the noise classes 1-8 are assessed much better than 9-11. Just in the case of noise type 9 many measurements are classified as “unknown”. Also many of the original sound records (class 12) are assessed as unknown condition (some records are assessed for more than 70% as unknown). This behavior shows that no neuron for one of the defined classes “error” or “error-free” was activated. This bad performance could be caused by too little records for this noise type. Anyway, the noise of a worn-out drive shaft is only noticeable for a few situations when the car is accelerated or high steering angle at low speed occurs. This is in contrast to steering rods where noticeable noise is emitted in many driving situations. Definitely all records without any noticeable noise are not considered for the database but there are still records where the noise occur e.g. for only one time in 10 seconds. However, for this case the SOM-Agent shall assess the moment where noise is noticeable as “error” class and all other as “error-free”.



**Figure 5:** SOM2-Agent statistic: A calculated probability of occurrence for 129 measurements, SOM with 1000 neurons and 6 parameters is used. Legend defines whether the analyzed measurements are assessed as “error”, “error-free” or “unknown” condition.

In comparison the SOM2-Agent (Figure 5) assesses much less time domain data (from 129 measurements) as “unknown”. Just the original sound records are classified to more than 50% of the time for the correct category “C2” (“error-free”). This is a result of the increased amount of neurons (1000), the corresponding expansion of the input space and the more detailed information of the trained input space. In contrast to SOM1-Agent, 6 times more neurons are used.

### 5. CONSLUSSION & FUTURE WORK

The SOM2-Agent produces a reliable statement of the matching category for the analyzed 129 acoustic in-car records. In comparison to SOM1-Agent (Figure 4 and 5) much less time-parts are declared as “unknown”. All car interior time signals of an error-free car are classified for more

<i>class</i>	<i>C1 [%]</i>	<i>C2 [%]</i>	<i>U [%]</i>
1	95,0	0,3	2,4
2	94,6	1,3	1,6
3	90,9	1,1	2,1
4	92,0	1,3	4,4
5	87,0	2,9	3,6
6	88,7	1,7	3,7
7	93,4	0,6	2,6
8	83,5	3,4	2,4
9	69,5	4,6	4,8
10	67,6	2,8	3,0
11	83,0	3,3	3,8
<b>12</b>	<b>1,8</b>	<b>87,3</b>	<b>3,4</b>

**Table 2:** Averaged probabilities of occurrence (SOM2-Agent) for 129 measurements and for different error noise (class 1-11) and error-free (class 12) records.

than 50% of time to the right category “C2”. The probability of occurrence of the other two categories is near less than 10%. Anyway, noises of worn-out/components are assessed at least for more than 40% of analyzed time to the right category/class. So it is important to make a decision by means of the relations between all statistic outputs “C1”, “C2” and “U”. For example, if



“C2” is much higher than “C1” it is likely that the analyzed signal contains no information for a worn-out part.

In Table 2 all averaged probabilities of occurrence of SOM2-Agent are listed. The noise types of the class 9 and 10 are recognized worst. All other noise types emitted from defect or worn-out components are recognized as a “error”-sound (“C1”) correctly via at least 83 % of the analyzed time. The “error-free”-sound (“C2”) - reproduced through class 12 - is evaluated above 87 % of the analyzed time correctly by the agent. However, the average shows one more time that the introduced method works well but nevertheless further improvements will be focused on the outliers. The presented method seems to be good for pattern recognition in acoustic records due to combination of traditional pattern recognition, signal processing and statistic analyze. However, it is understood that the results will be proof for the analyzed environment conditions (e.g. car type, microphone type, measurement procedure). Furthermore, the analysis of the influence of different car types on the features and pattern recognition is necessary in further studies. It was shown that the number of the neurons has a decisive influence on the accuracy and reliability of the SOM-agent. It has to be examined also by means of further measurements how far the number of neurons, adaptation accuracy and over-fitting influence the training and assessment process.

## 6. REFERENCES

- [1] B. Orend, I. Meyer, *Schadensfrüherkennung mittels Körperschall*. MTZ, 2009
- [2] R. Jahnke, *Beitrag zur akustischen Qualitätsprüfung im Bereich industrieller Fertigung*. TU Berlin, 2001
- [3] K. Stepper, *Ein Beitrag zur akustischen Güteprüfung von Komponenten der Kfz.- und Automatisierungstechnik*. TU Berlin, 1999
- [4] T. Kohonen, *Self-organizing maps*, Springer Verlag, 2001
- [5] S. Twieg, H. Opfer, H. Beikirch, *Correlation based method for acoustic condition recognition*. 1st IEEE Germany Student Conference, Hamburg University of Technology, 2009
- [6] K.-M. Würzner, *Textkategorisierung: Entscheidungsbäume*. Universität Bielefeld, 2003
- [7] A.-W. Moore, *Information Gain*. Carnegie Mellon University Pittsburgh, 2003
- [8] L. Yu, H. Liu, *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution*. ICML, 2003
- [9] L. Yu, H. Liu, I. Guyon, *Efficient Feature Selection via Analysis of Relevance and Redundancy*. Journal of Machine Learning Research, pp. 1205-1224, 2004
- [10] A. Asuncion, D. Newman, *UCI Machine Learning Repository*, University of California, Irvine, School of Information and Computer Sciences, 2007
- [11] T. Fincke, V. Lobo, F. Baco, *Visualizing self-organizing maps with GIS*, GI Days 2008 Münster, 2008
- [12] J. Biesiada, W. Duch, *Feature Selection for High-Dimensional Data: A Kolmogorov-Smirnov Correlation-Based Filter*, Advances in Soft Computing, Springer Verlag, pp. 95-104, 2005
- [13] O. Chapelle, *Feature selection for Support Vector Machines*, Max Planck Institute for Biological Cybernetics, Tübingen, 2005

[14] M. A. Hall, L. A. Smith, *Feature subset selection: a correlation based filter approach*. International Conference on Neural Information Processing and Intelligent Information Systems. © Springer, pp. 855-858, 1997

[15] Zhipeng Li & Fuqiang Liu, Cellular Automaton Based Simulation for Signalized Street. International Journal of Engineering and Interdisciplinary Mathematics, vol. 1, no. 1, pp. 11-22, 2009

[16] Niemann, H. (1988). *Klassifikation von Mustern*. Springer Verlag, Berlin, Heidelberg, New York, Tokyo, 1988

# CALL FOR PAPERS

**Journal:** Signal Processing: An International Journal (SPIJ)

**Volume:** 3 **Issue:** 6

**ISSN:** 1985-2339

**URL:** <http://www.cscjournals.org/csc/description.php?JCode=SPIJ>

## About SPIJ

The International Journal of Signal Processing (SPIJ) lays emphasis on all aspects of the theory and practice of signal processing (analogue and digital) in new and emerging technologies. It features original research work, review articles, and accounts of practical developments. It is intended for a rapid dissemination of knowledge and experience to engineers and scientists working in the research, development, practical application or design and analysis of signal processing, algorithms and architecture performance analysis (including measurement, modeling, and simulation) of signal processing systems.

As SPIJ is directed as much at the practicing engineer as at the academic researcher, we encourage practicing electronic, electrical, mechanical, systems, sensor, instrumentation, chemical engineers, researchers in advanced control systems and signal processing, applied mathematicians, computer scientists among others, to express their views and ideas on the current trends, challenges, implementation problems and state of the art technologies.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for SPIJ.

## SPIJ List of Topics

The realm of International Journal of Signal Processing (SPIJ) extends, but not limited, to the following:

- Biomedical Signal Processing
- Communication Signal Processing
- Detection and Estimation
- Earth Resources Signal Processing
- Industrial Applications
- Optical Signal Processing
- Acoustic and Vibration Signal Processing
- Data Processing
- Digital Signal Processing
- Geophysical and Astrophysical Signal Processing
- Multi-dimensional Signal Processing
- Pattern Recognition

- Radar Signal Processing
- Signal Filtering
- Signal Processing Technology
- Software Developments
- Spectral Analysis
- Stochastic Processes
- Remote Sensing
- Signal Processing Systems
- Signal Theory
- Sonar Signal Processing
- Speech Processing

## **CFP SCHEDULE**

**Volume:** 4

**Issue:** 1

**Paper Submission:** January 31 2010

**Author Notification:** February 28 2010

**Issue Publication:** March 2010

## CALL FOR EDITORS/REVIEWERS

CSC Journals is in process of appointing Editorial Board Members for ***Signal Processing: An International Journal (SPIJ)***. CSC Journals would like to invite interested candidates to join **SPIJ** network of professionals/researchers for the positions of Editor-in-Chief, Associate Editor-in-Chief, Editorial Board Members and Reviewers.

The invitation encourages interested professionals to contribute into CSC research network by joining as a part of editorial board members and reviewers for scientific peer-reviewed journals. All journals use an online, electronic submission process. The Editor is responsible for the timely and substantive output of the journal, including the solicitation of manuscripts, supervision of the peer review process and the final selection of articles for publication. Responsibilities also include implementing the journal's editorial policies, maintaining high professional standards for published content, ensuring the integrity of the journal, guiding manuscripts through the review process, overseeing revisions, and planning special issues along with the editorial team.

A complete list of journals can be found at <http://www.cscjournals.org/csc/byjournal.php>. Interested candidates may apply for the following positions through <http://www.cscjournals.org/csc/login.php>.

*Please remember that it is through the effort of volunteers such as yourself that CSC Journals continues to grow and flourish. Your help with reviewing the issues written by prospective authors would be very much appreciated.*

Feel free to contact us at [coordinator@cscjournals.org](mailto:coordinator@cscjournals.org) if you have any queries.

## **Contact Information**

### **Computer Science Journals Sdn Bhd**

M-3-19, Plaza Damas Sri Hartamas  
50480, Kuala Lumpur MALAYSIA

Phone: +603 6207 1607  
          +603 2782 6991  
Fax:      +603 6207 1697

### **BRANCH OFFICE 1**

Suite 5.04 Level 5, 365 Little Collins Street,  
MELBOURNE 3000, Victoria, AUSTRALIA

Fax: +613 8677 1132

### **BRANCH OFFICE 2**

Office no. 8, Saad Arcad, DHA Main Bulevard  
Lahore, PAKISTAN

### **EMAIL SUPPORT**

Head CSC Press: [coordinator@cscjournals.org](mailto:coordinator@cscjournals.org)  
CSC Press: [cscpress@cscjournals.org](mailto:cscpress@cscjournals.org)  
Info: [info@cscjournals.org](mailto:info@cscjournals.org)

COMPUTER SCIENCE JOURNALS SDN BHD  
M-3-19, PLAZA DAMAS  
SRI HARTAMAS  
50480, KUALA LUMPUR  
MALAYSIA