

Automatic Classification of LFM Signals for Radar Emitter Recognition Using Wavelet Decomposition and LVQ Classifier

E. ŚWIERCZ*

Faculty of Electrical Engineering, Białystok University of Technology
Wiejska 45D, 15-351 Białystok, Poland

The paper presents a novel approach, based on the wavelet decomposition and the learning vector quantisation algorithm, to automatic classification of signals with linear frequency modulation, generated by radar emitters. The goal of radar transmitter classification is to determine the particular transmitter, from which a signal originated, using only the just received waveform. To categorise a current linear frequency modulation signal to the particular transmitter, the discrete wavelet decomposition of the received signal is accomplished in order to get a representative set of features with good classification properties. The learning vector quantisation algorithm with a previously defined set of features as an input of the learning vector quantisation neural net is proposed as the intelligent classification algorithm, which combines competitive learning with supervision. After the learning process, the learning vector quantisation algorithm is ready to perform the classification process for different data than data used in the learning stage. Simulation results show the high classification accuracy for experimentally chosen wavelets and suggested architecture of the learning vector quantisation classifier.

PACS: 84.40.Ua, 84.35.+i

1. Introduction

Emitter classification based on a collection of received radar signals is a subject of wide interest in both civil and military applications. The received signals usually consist of sequences of pulses with complex signals, such as linear frequency modulated, phase-shift keying (PSK) - and frequency-shift keying (FSK)-modulated signals et al., due to their use in low probability of intercept radar systems. The dense, complex, contaminated and continuously changing signal environment makes the classification process extremely difficult. Conventional algorithms of emitter identification are basically matching techniques, which perform a sequence of tests comparing measured values of a parameter with parameter values of emitters previously gathered in the library. For classification of radar signals, traditional methods often use radar signal parameters such as pulse amplitude, pulse width, radio frequency and time of arrival. In today's complex environment such methods often collapse or cannot work effectively. Therefore, in order to perform automatic classification, a novel approach is strongly required [1, 2].

In this paper the classification procedure is focused on transmitters generating signals with linear frequency modulation (LFM). Experiments have been performed only in simulation environment and only the signal model has been exploited to assess the capabilities of the new algorithm. There are no limitations on using the presented

algorithm for other classes of modulated signals. It is assumed that the intercepted LFM wave forms with different slopes of linear frequency modulation are assigned to different radar emitters (different classes). A key problem is to find a set of discriminative features which can distinguish such defined signals and classify them to a particular emitter regardless of non-stability of the slope (small changes of the slope) and noise distortions.

In the proposed approach, the discrete wavelet decomposition is used for extraction of characteristic features. Unfortunately, the wavelet decomposition generates a lot of data, so dimensionality reduction has to be performed. In this study a low dimension feature vector is formed by aggregation of properties of wavelet coefficients. Extracted features are processed by a classifier to select the most probable class. The supervised classification is considered where classes are known beforehand, and feature samples of each class are available.

The extraction of features from wavelet decomposition has to carry out three assumptions:

- the features should be obtained easily so as to speed up the whole process,
- the features should be in a low-dimensional space to reduce the computational complexity,
- the features should be discriminative.

In this manuscript the feature vector is an input of an intelligent classifier, based on the vector quantisation concept. The learning vector quantisation (LVQ) neural network has been chosen because of its ability of learning data classification, where the similar input vectors are

* corresponding author; e-mail: ewasw@pb.edu.pl

grouped into a region, represented by a so-called coded vector (CV).

2. LFM signals

The goal of radar transmitter classification is to determine the particular transmitter, from which a LFM signal originated, using only one received wave form. No localisation information is exploited to accomplish this task. Let us consider a single LFM signal in the interval $[0, T]$. The complex envelope of such a signal can be written as

$$s(t) = \exp(j\phi(t)), \quad 0 \leq t \leq T, \quad (1)$$

where the phase function $\phi(t) = b_0 + b_1t + b_2t^2$ is the second order polynomial. The instantaneous angular frequency can be computed as

$$\omega(t) = \frac{d\phi(t)}{dt} = b_1 + 2b_2t, \quad (2)$$

where b_1 is an initial frequency. The second derivative of the phase function $\phi(t)$ is equal to $k = 2b_2$. This quantity called the chirp rate or the frequency slope is related to the time duration T and the band width B of the signal as follows [3]:

$$k = \pm \frac{B}{T}. \quad (3)$$

In this study only the discrete sequence of samples of a LFM signal achieved by the sampling of the continuous wave with the sampling period $\Delta T = T/N$ is examined. A discrete time version of the length N of Eq. (1) can be written as a discrete expression

$$s(n) = \exp(j(b_0 + b_1n + b_2n^2)), \quad (4)$$

$$n = 0, 1, 2, 3, \dots, N - 1.$$

In the real scenario only the collection of LFM signals generated by a particular emitter is needed to create database rich enough for extracting unique properties of signals for training and testing steps of the neural classifier. If radar emits generated signals with a different kind of frequency modulation such as LFM-, PSK- or FSK-modulated, the classification process would be considerably easier because classes of signals are significantly different. The proposed classification method concerns only LFM modulation with a different chirp slope generated in simulation environment. It is assumed that each transmitter generates signals with the same parameters b_0 and b_1 , but the frequency slope parameter described by the parameter b_2 is a specific value for an individual transmitter. Let us denote the frequency slope k as

$$k = 2b_2 = \frac{f_{\text{normf}} - f_{\text{normi}}}{N - 1}, \quad (5)$$

where f_{normi} and f_{normf} are normalised frequencies lying in the range $[-0.5, 0.5]$ with respect to the sampling period (ratio of the frequency in hertz to the sampling frequency, with respect to the Shannon sampling theorem). This signal is called a chirp, and as its frequency content is varying with time, it is a non-stationary signal [4]. To simplify further simulations it is assumed that only the parameter b_2 is a changeable parameter and the

simplified form of LFM signals under considerations can be written as

$$s(n) = \exp(jb_2n^2). \quad (6)$$

Automatic recognition of the parameter value b_2 for LFM signals reformulated as the classification problem can be also treated as a part of a general problem of automatic modulation recognition [5].

3. Feature extraction via discrete wavelet decomposition

In this study, in order to classify the signals, a feature vector Φ is formulated by time-frequency processing of a signal. The main advantage of wavelets is that they have a varying window size, being wide for slow frequencies and narrow for the fast ones. Thus it leads to an optimal time-frequency resolution in all frequency ranges. Owing to the fact that windows are adapted to the transients of each scale, wavelets are able to process non-stationary signals. LFM signals belong to the class of non-stationary signals so that wavelet processing is a suitable tool for processing such signals. The idea of the *continuous wavelet transform* (CWT) is to project the signal $s(x)$ on a family of zero-mean functions (the *wavelets*) deduced from an elementary function (the *mother-wavelet*) by translations and dilations

$$C_s(b, a; \Psi) = \int_{-\infty}^{+\infty} s(x) \Psi_{b,a}^*(x) dx, \quad (7)$$

where

$$\Psi_{b,a}^*(x) = |a|^{-1/2} \Psi\left(\frac{x-b}{a}\right), \quad a > 0, b \in R, \quad (8)$$

is a family associated with the one-dimensional, continuous wavelet.

The variable a represents the scale, whereas b is the translation of the mother-wavelet. Calculating wavelet coefficients at every possible scale generates a lot of data. If we choose scales based on powers of two, so-called *dyadic* scales ($a = 2^i$, $b = ja = j2^i$), we can formulate a double indexed family of wavelets $\psi_{i,j}(x)$ from the mother-wavelet function in the form

$$\Psi_{i,j}(x) = 2^{-i/2} \Psi(2^{-i}x - j), \quad i \in Z, j \in Z. \quad (9)$$

It means that the wavelet representation $C(a, b)$ is viewed as a discrete set of coefficients $C(i, j)$. This is a case of continuous time, "discrete" analysis. A lot of wavelets (not all) are associated with their scale functions $\phi_{i,j}(x)$, which also can be expressed in the indexed form

$$\phi_{i,j}(x) = 2^{-i/2} \phi(2^{-i}x - j), \quad i \in Z, j \in Z. \quad (10)$$

It should be noted that the function $\psi(x)$ can be written as an indexed structure $\psi(x) = \psi_{0,0}(x)$ and the scale function $\phi(x)$ as the structure $\phi(x) = \phi_{0,0}(x)$, respectively.

The wavelets defined in such a way result in multi-resolution decomposition of a signal at different decomposition levels with different resolutions. Each level of decomposition is closely associated with the scale index i . The fundamental relations between two successive scales

are called “twin-scale relations” also known as the dilation equation

$$\phi(x) = \sqrt{2} \sum_j h(j) \phi(2x - j). \quad (11)$$

A scale function $\phi(x) = \phi_{0,0}(x)$ is a linear combination of scale function which can be expressed as an indexed form $\phi(2x - j) = 2^{-1/2} \phi_{-1,j}(x)$ according to the relation (10). Additionally, the wavelet equation for the wavelet function $\psi(x) = \psi_{0,0}(x)$ is defined as a different linear combination of scale function

$$\Psi(x) = \sqrt{2} \sum_j g(j) \phi(2x - j). \quad (12)$$

A sequence $h(k)$ can be treated as the finite impulse response (FIR) low-pass filter and a sequence $g(k)$ as the FIR high-pass filter.

The wavelet decomposition is usually defined in the equivalent way by a collection of detailed coefficients $d_{i,j}$ also called cD coefficients in MATLAB environment and approximation coefficients $a_{i,j}$ called cA coefficients, respectively. Coefficients at different levels of decomposition are generated through the $h(k)$ filter for approximation coefficients and through the $g(k)$ filters for detailed coefficients. A collection of coefficients using the multi-resolution approach defines the discrete wavelet transform (DWT). The decomposition algorithm starts with the signal $s(x)$ and two sets of coefficients: approximation coefficients $cA1$ and detailed coefficients $cD1$ are produced, then it calculates the successive levels of decomposition. The signal $s(x)$ at the M -th level of decomposition can be written as

$$s_M(x) = \sum_j a_{M+1,j} \phi_{M+1,j}(x) + \sum_{i=0}^M d_{i+1,j} \Psi_{i+1,j}(x). \quad (13)$$

Features for the classification task have been searched among some aggregated properties of wavelet coefficients at different levels of decomposition for different wavelets [6–9].

Numerical experiments have been run in MATLAB environment, so a discrete form of a signal of the length N has been used. It has been assumed that M is the reference level of decomposition. The MATLAB function *wavedec* performs multi-level decomposition and returns for each level of decomposition the wavelet decomposition vector containing one set of approximation coefficients cA_j of the length nA ($j = 1, \dots, nA$) and a few sets of detailed coefficients $cD_{i,j}$ for $i = 1, \dots, M$ with the appropriate length related to the level [9]. The MATLAB function *wenergy* has been used for computing the energy E_a (a scalar number), which is the percentage of energy corresponding to the approximation [9]. A vector E_d of the length M contains the percentages of energy corresponding to the detail. Having the full wavelet decomposition (all cA , cD coefficients) the following aggregated properties have been proposed:

Energy E :

$$E_a + \sum_{i=1}^M E_{d,i} = 100.$$

Cmean: Mean values of decomposition coefficients in a decomposition tree (a hierarchical structure of decomposition levels)

$$cA_{\text{mean}} = \frac{1}{nA} \sum_{j=1}^{nA} cA_j$$

— for approximation coefficients,

$$cD_{\text{mean},i} = \frac{1}{nD_i} \sum_{j=1}^{nD_i} cD_{i,j}$$

— for coefficients at detailed levels, $i = 1, \dots, M$.

Cabsmean: Means of absolute values of decomposition coefficients in a decomposition tree

$$cA_{\text{absmean}} = \frac{1}{nA} \sum_{j=1}^{nA} |cA_j|$$

— for approximation coefficients,

$$cD_{\text{absmean},i} = \frac{1}{nD_i} \sum_{j=1}^{nD_i} |cD_{i,j}|$$

— for coefficients at detailed levels, $i = 1, \dots, M$.

Power-mean: Average power (squares) of decomposition coefficients in a decomposition tree

$$cA_{\text{pow}} = \frac{1}{nA} \sum_{j=1}^{nA} (cA_j)^2$$

— for approximation coefficients,

$$cD_{\text{pow},i} = \frac{1}{nD_i} \sum_{j=1}^{nD_i} (cD_{i,j})^2$$

— for coefficients at detailed levels, $i = 1, \dots, M$.

Evar: Variances of decomposition coefficients in a decomposition tree

$$cA_{\text{var}} = \sum_{j=1}^{nA} (cA_j - cA_{\text{mean}})^2$$

— for approximation coefficients,

$$cD_{\text{var},i} = \sum_{j=1}^{nD_i} (cD_{i,j} - cD_{\text{mean},i})^2$$

— for coefficients at detailed levels, $i = 1, \dots, M$.

Ratio: Ratios of means of absolute values of decomposition coefficients in adjacent sub-bands in a decomposition tree (for coefficients at detailed levels only)

$$cD_{\text{absmeanratio},i} = \left(\frac{1}{nD_{i+1}} \sum_{j=1}^{nD_{i+1}} |cD_{i+1,j}| \right) / \left(\frac{1}{nD_i} \sum_{j=1}^{nD_i} |cD_{i,j}| \right)$$

— for coefficients at detailed levels, $i = 1, \dots, M - 1$.

From the set of quantities proposed above, ones with the best distinctive abilities are selected to create the feature vector in the feature space. The feature vector is the input of a neural classifier, based on the LVQ algorithm [5, 8].

4. The LVQ neural net

LVQ is a supervised version of vector quantisation [10]. As a supervised method, LVQ uses known target output classifications for each input pattern. The main idea is to cover the input space of samples with “codebook vectors” (CVs), each representing a region labelled with a class. A CV can be seen as a prototype of a class member, localised in the centre of a class or a decision region (“Voronoi cell”). CV can be described as a hidden neuron (“Kohonen neuron”) or a weight vector of the weights between all inputs and the regarded Kohonen neuron. A class can be represented by an arbitrary number of CVs, but one CV represents one class only.

The LVQ neural network is built as a feedforward net with one hidden layer of neurons (the Kohonen layer), fully connected with the input layer. Classes for each input pattern are predefined. During the training stage, the values of weights used to form the coded vectors are adjusted, according to the patterns of input samples, in order to match desired classes. The distance d_i of an input vector \mathbf{x} to the weight vector \mathbf{w}_i of each node in the Kohonen layer is computed. The node of a particular class, which has the smallest distance to the presented input vector (for example the Euclidean distance), is declared to be the winner

$$d_i = \|\mathbf{w}_i - \mathbf{x}\| = \sqrt{\sum_{j=1}^M (w_{ij} - x_j)^2}. \quad (14)$$

The weights will be moved closer to the class if it is the expected winning class, otherwise they will be moved away

$$\mathbf{W} = \begin{cases} \mathbf{w} + \alpha(\mathbf{x} - \mathbf{w}), \\ \mathbf{w} - \gamma(\mathbf{x} - \mathbf{w}). \end{cases} \quad (15)$$

After the training process is finished, the LVQ network is then ready for classifying an unknown input. The classification after learning is relied on finding a Voronoi cell, specified by the CV with the smallest distance to the input vector and assigning it to the labelled class.

5. Classification of LFM signals

In a real radar scenario the collection of acquired signals generated by radar emitters being in our interest is sufficient to perform the complete classification process. Acquired signals are usually divided into two parts: the first for learning and the second for testing. In the presented paper signals for extracting features, for learning and for training were obtained only through simulations. Let us assume that LFM signals belong to five

classes $\omega = (\omega_1 \omega_2 \omega_3 \omega_4 \omega_5)$ related to the five different slope coefficients k defined in (3). Each slope coefficient k is closely connected to the frequency, changing from f_{normi} to f_{normf} during a time interval of a signal. It has been assumed that the length of simulated signals is $N = 256$ and $f_{\text{normi}} = 0$. The normalised frequency f_{normf} takes values from the following set, $f_{\text{normf}} \in [0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.5]$, which imposes values for the slope coefficient $k_{\text{cen}} \in \Theta = [2 \ 4 \ 6 \ 8 \ 10] \times 10^{-4}$. These settings do not limit the generality of considerations and are used only for simplification in simulations, performed in this paper. The actual value of the parameter k “around” k_{cen} will be classified exactly as k_{cen} . It means that if the frequency slope k of a LFM signal generated by a particular radar emitter slightly changes in a random way (non-stability of a slope) the classifier will take a decision to assign this signal to the emitter with the nearest value of k_{cen} .

The training data representing each class has consisted of the set of 205 LFM signals (41 per each class) obtained by changing k in the range $[-5 \times 10^{-5}, +5 \times 10^{-5}]$ around $k_{\text{cen}} \in \Theta$ with the step equal to 2.5×10^{-6} . It is worth pointing out that the algorithm is performed for very small values. This means that considered signals are closely-spaced chirp signals from the numerical point of view.

The goal of the classifier is the selection of the most probable value of the element from the vector Θ , which corresponds to the selection of the class ω . The five-element feature vector as the input vector for the LVQ classifier has been built from aggregation of features of the wavelet decomposition.

Detailed examination of the characteristic values presented in Sect. 3 revealed that combinations of selected subsets of them are sufficiently good to create the classification space, i.e. the feature vectors to be passed to LVQ inputs. From examination of a variety of wavelet decompositions, reverse biorthogonal spline wavelets (*rbio* family) and the Daubechies wavelets (*db* family) turned out to be the most interesting [9].

For the *rbio2.2* wavelet a set of discriminative features for all classes has been successfully found. The feature vector was created from the following quantities: $\Phi = [Ea(A1), Ed(D1), Cabsmean(D4), var(D4), Ratio(D2/D1)]$ which reads: the energy of approximation coefficients at the first level of decomposition — with the short name *Ea(A1)*, the energy of detailed coefficients at the first level of decomposition — with the short name *Ed(D1)*, mean of absolute values of detailed coefficients at the fourth level of decomposition — with the short name *Cabsmean(D4)*, variance of detailed coefficients at the fourth level of decomposition — with the short name *var(D4)* and ratio of means of absolute values of detailed coefficients at the first and the second level of decomposition — with the short name *Ratio(D2/D1)*. There is not the only vector of features and a lot of different subsets of features are possible but the selected one gave good enough separation between classes for *rbio2.2* wavelet.

Figures 1–6 show examples of characteristic variables extracted from wavelet decomposition with *rbio2.2* employed up to the level 4th.

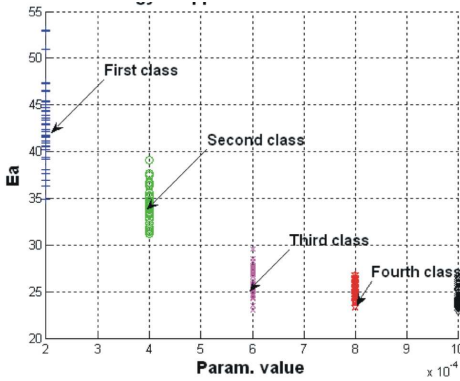


Fig. 1. Selection of energy of approximation $Ea(A1)$ because of good separation of the 1st class and the 2nd classes from 3th, 4th and 5th class.

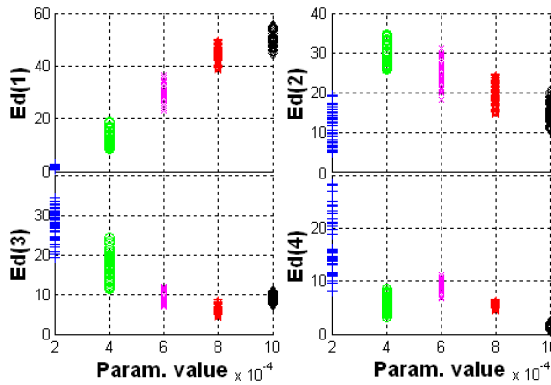


Fig. 2. Selection of energy of decomposition coefficients at detailed level $Ed(D1)$, because of its good class-discrimination properties except separation between 4th and 5th class.

Except for *rbio2.2* wavelet, the *db4* and *db5* wavelets were examined because of their potential possibilities of discrimination between classes. It was surprising that the vector of features $\Phi = [Ea(A1), Ed(D1), Cabsmean(D4), var(D4), Ratio(D2/D1)]$ created especially for *rbio2.2* wavelet turned out to be good enough for *db4* and *db5* wavelets because the classification accuracy obtained during the training was 100% both for the *rbio2.2* wavelet and for *db4* and *db5* wavelets.

Testing data are used to test whether the classifier is able to work also on the data that were not used in the training process. In real environment, acquired data are usually divided into a training part and a testing part. In simulation environment testing data have to be created numerically. In this paper noisy signals as testing signals have been obtained by adding the noise signal to LFM signals used previously in the training step.

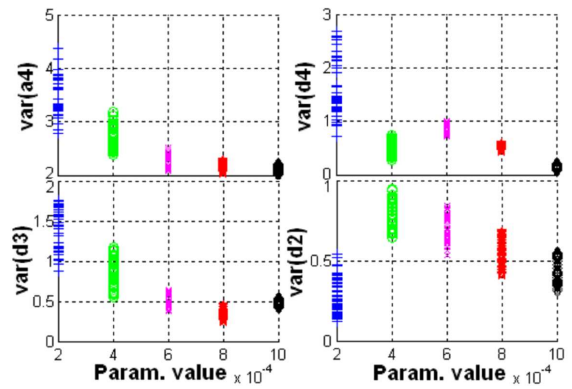


Fig. 3. Selection of variance of detailed coefficients $var(D4)$, because of good separation of 5th class from 1st, 3th and 4th class.

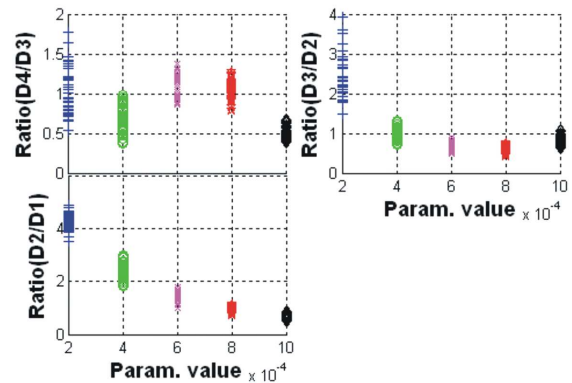


Fig. 4. Selection of $Ratio(D2/D1)$ — ratio of means of absolute values of detailed coefficients at the first and the second level of decomposition because of good class separation except the small overlap between 3th and 4th and 4th and 5th class.

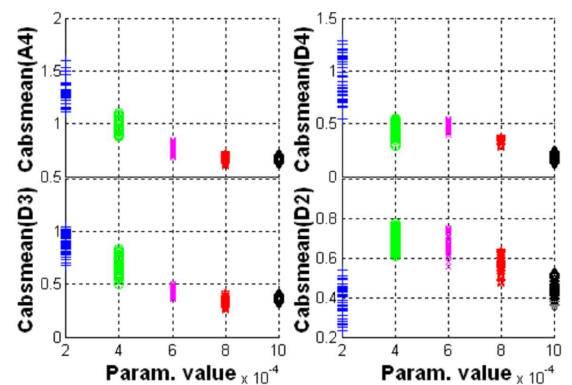


Fig. 5. Selection of mean of absolute values of detailed coefficients at the fourth level of decomposition $Cabsmean(D4)$ because of good separation of 5th class from all remaining classes.

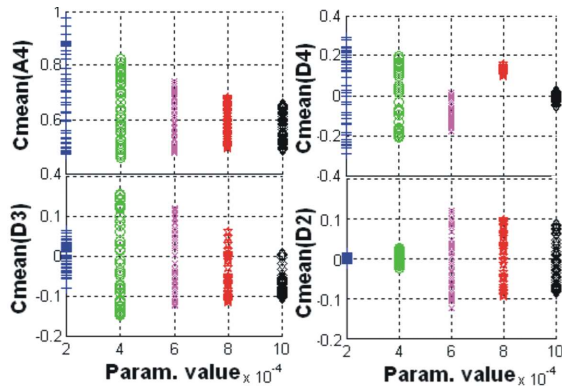


Fig. 6. Example of bad choice of properties because of large overlap of all classes.

Testing signals for the generalisation task were generated by slightly changing values of the parameter k used previously in the training, providing that new values stayed still inside the range $[-5 \times 10^{-5}, +5 \times 10^{-5}]$ around $k_{cen} \in \Theta$. All examined wavelets demonstrated very good generalisation properties and classification results achieved 100% effectiveness. Next the extrapolation ability was assessed for values of the parameter k which completely stay outside the interval $[-5 \times 10^{-5}, +5 \times 10^{-5}]$. The border $\pm 10^{-4}$ means that the parameter k having such a value can be equally likely assigned to two neighbouring classes. However, good performance of the network outside the training range should not be expected in any case.

Classification accuracy for noisy signals with increased variances has been presented in Fig. 7 for the *rbio2.2* wavelet, for the *db4* wavelet and for the *db5* wavelet with the same set of features

$$\Phi = [Ea(A1), Ed(D1), Cabsmean(D4), var(D4), Ratio(D2/D1)]$$

for all considered wavelets. It is worth pointing that in the experiment presented in Fig. 7 the set of features Φ was created especially for the *rbio2.2* wavelet and it is not the best choice for the *db4* wavelet and for the *db5* wavelet. In Fig. 8 the classification for noisy signals has been performed establishing the set

$$\Phi_{rbio2.2} = [Ea(A1), Ed(D1), Cabsmean(D4), var(D4), Ratio(D2/D1)]$$

with the best separating properties for the *rbio2.2* wavelet, the set

$$\Phi_{db4} = [Ea(A1), Ed(D1), Ed(D4), Power-mean(A4), Power-mean(D4)]$$

with the best separating properties for the *db4* wavelet and the set

$$\Phi_{db5} = [Ed(D1), Ed(D4), Power-mean(A4), Power-mean(D4), Ratio(D2/D1)]$$

with the best separating properties for the *db5* wavelet.

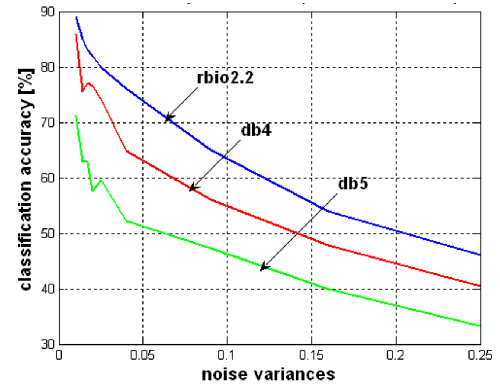


Fig. 7. Classification accuracy for noisy signals with one set of features for all wavelets.

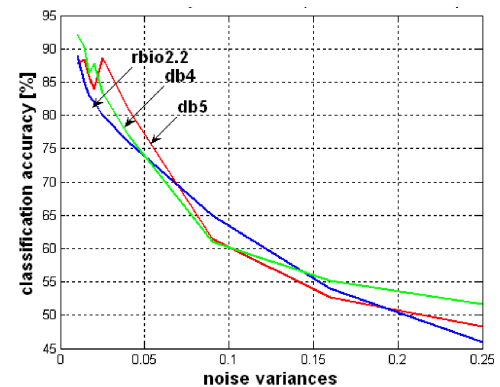


Fig. 8. Classification accuracy for noisy signals with sets of features for particular wavelets.

As it could be expected, increasing noise variance causes deterioration of classification. In the case of non-noisy signals, 100% classification accuracy has been attained for the same values of the parameter k .

It is obvious that the better class-discriminative features for particular wavelets are found, the better classification accuracy is obtained both in training and in testing stages.

The purpose of next experiments is to determine the range of the parameter k for which networks can extrapolate. Table I and Table II show the classification accuracy for values of the parameter k , which stay outside the interval (i.e. symmetrically on both sides of the central point from $\Theta = [2 \ 4 \ 6 \ 8 \ 10] \times 10^{-4}$), from which the values for network training have been chosen.

As it can be seen, the classification accuracy for values k outside the training range significantly decreases.

The wavelet *db5* causes some inconvenience because it shows better classification accuracy for a noisy signal with the set Φ_{db5} than with the set Φ , but at the same time shows worse classification accuracy in extrapolation with the set Φ_{db5} than with the set Φ . Owing to this

ambiguity the wavelet *db5* should be rejected from considerations. Selection of wavelets and a set of appropriate

input feature variables is an important issue in building the neural, as well as other classifiers.

TABLE I

Classification accuracy [%] for extended range of parameter variability. Wavelet decomposition up to 4th level with $\Phi = [Ea(A1), Ed(D1), Cabsmean(D4), var(D4), Ratio(D2/D1)]$ for all wavelets.

Range of parameter changes	Classification accuracy of non-noisy signals for <i>rbio2.2</i>	Classification accuracy of non-noisy signals for <i>db4</i>	Classification accuracy of non-noisy signals for <i>db5</i>
$(0.5; 0.75] \times 10^{-4}$	91.00%	87.00%	98.00%
$(0.75; 1] \times 10^{-4}$	72.00%	65.00%	73.00%
$(0.5; 1] \times 10^{-4}$	82.00%	74.00%	84.50%

TABLE II

Classification accuracy [%] for extended range of parameter variability. Wavelet decomposition up to 4th level with the set Φ fitted to the particular wavelet.

Range of parameter changes	Classification accuracy of non-noisy signals for <i>rbio2.2</i>	Classification accuracy of non-noisy signals for <i>db4</i>	Classification accuracy of non-noisy signals for <i>db5</i>
$(0.5; 0.75] \times 10^{-4}$	91.00%	94.72%	86.00%
$(0.75; 1] \times 10^{-4}$	72.00%	72.00%	66.00%
$(0.5; 1] \times 10^{-4}$	82.00%	83.50%	74.50%

6. Conclusions

The classification of LFM signals from different radar emitters, via classification of features extracted from the received signal, has been considered in the paper. The classification scheme consisted of data pre-processing with the use of discrete wavelet transform, feature extraction by aggregation of properties of decomposition coefficients and intelligent classification with the use of LVQ neural networks. The goal of a classifier is to partition the feature space into class-labelled decision regions.

In this paper the feature space is represented by “codebook vectors” (CVs), each representing a region labelled with a class. A neural LVQ classifier compares input features with CV with the nearest distance and points the class represented by this CV. Five-element input (5-dimensional feature vector) and five-element output (established number of classes) architecture of the neural classifier has been successfully developed. Simulation experiments showed that wavelet decomposition has the ability to separate signal features for different ranges of values of a changeable parameter k representing the slope of LFM signals. The classification system turned out to be robust to some level of noise for chosen wavelets *rbio2.2* and *db4* and has had certain extrapolation ability.

References

- [1] J. Liu, J.P.Y Lee, L. Li, Z.-Q. Luo, K.M. Wong, *IEEE Trans. Pattern Anal. Machine Intellig.* **27**, 1185 (2005).
- [2] M.-Q. Ren, Y.-Q. Zhu, Y. Mao, J. Han, in: *Proc. 2007 Int. Conf. on Wavelet Analysis and Pattern Recognition, Beijing (China)*, University of Science and Technology, Beijing 2007, p. 1442.
- [3] K. Kulpa, in: *Proc. Int. Conf. on Radar*, IEEE Cat. No.03EX695, *Adelaide (Australia)*, 2003, p. 235.
- [4] F. Auger, P. Flandrin, P. Goncalves, O. Lemoine, *Time-Frequency Toolbox For Use with Matlab*, CNRS, France, Rice University, USA, 1995–1996.
- [5] E. Świercz, in: *Proc. XI Int. Radar Symp. IRS-2010, Vilnius (Lithuania)*, 2010, p. 599.
- [6] Y. Al-Assaf, *Comput. Industr. Eng.* **47**, 17 (2004).
- [7] E.D. Ubeyli, *Expert Systems Appl.* **34**, 1954 (2008).
- [8] E. Świercz, in: *Computational Methods and Experimental Measurements XIV*, Eds. C.A. Brebbia, G.M. Carlomagno, WIT Press, Southampton 2009, p. 271.
- [9] M. Misiti, Y. Misiti, G. Oppenheim, J.-M. Poggi, *Wavelet Toolbox For Use with MATLAB*, The MathWorks, Inc., 24 Prime Park Way, Natick, MA 01760–1500, 1996–1997.
- [10] L. Fausett, *Fundamental of Neural Networks Architectures, Algorithms, and Applications*, Prentice-Hall, Inc., Upper Saddle River, NJ 1994.