

J. Lundén, L. Terho and V. Koivunen, “Waveform recognition in pulse compression radar systems,” in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Mystic, CT, USA, September 28–30, 2005, pp. 271–276.

© 2005 IEEE. Reprinted with permission.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the Helsinki University of Technology’s products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this material, you agree to all provisions of the copyright laws protecting it.

WAVEFORM RECOGNITION IN PULSE COMPRESSION RADAR SYSTEMS

Jarmo Lundén, Liisa Terho* and Visa Koivunen

Signal Processing Laboratory, SMARAD CoE
Helsinki University of Technology
P.O. Box 3000, FIN-02015 TKK, Finland

ABSTRACT

In this paper a system for recognizing pulse compression radar waveforms is introduced. The waveforms considered in this study are the linear frequency modulation (LFM), Costas codes, binary phase codes, and the Frank, P1, P2, P3, and P4 codes. A feature vector based on instantaneous signal properties, second- and higher-order statistics, and time-frequency distributions is computed from the received signals. Cyclic correlations are used in symbol rate estimation. Information theoretic measure is used to remove redundant components from the feature vector. The discrimination capability of the features is evaluated using an ensemble averaging early-stop committee of multilayer perceptrons. Bayesian MLP classifier is considered as well. In simulation the classifier attains over 97 % overall correct classification rate in signal-to-noise ratio (SNR) of 6 dB.

1. INTRODUCTION

Automatic waveform recognition is an important task in spectrum management and surveillance, communication and radar emitter identification, electronic support applications as well as in software defined radio (SDR) and intercept receivers. A key task in waveform recognition is to find a set of features that distinguish various waveforms from each other. Typically a supervised classifier is trained in order to recognize different waveforms emitted by a transmitter. In order to make the computational complexity lower, the dimensionality of the potential feature vector has to be reduced by removing the redundancy among the features. The problem of waveform classification has been earlier considered mostly in the context of modulation recognition in communication systems [1–6].

In this paper a method for recognizing the pulse compression waveforms used in radar systems is introduced. The employed features are based on instantaneous signal properties, second- and higher-order statistics, and time-frequency distributions. The pulse compression waveforms considered are the linear frequency modulation, Costas discrete frequency codes, binary phase codes, and the Frank, P1, P2, P3, and P4 codes. Cyclic correlations are used to estimate symbol rate. A hierarchical classifier structure is employed. The actual supervised classifier is ensemble averaging early-stop committee of multilayer perceptrons. Bayesian MLP structure is considered as well.

The paper is organized as follows. An overview of the classification system is presented in Section 2. The actual classifier is described in Section 3. The features employed in the recognition

*Liisa Terho is with the Finnish Defence Forces Technical Research Centre.

process are briefly described in Section 4. Simulation examples are provided in Section 5. The classifier performance is studied as a function of the signal-to-noise ratio. Highly reliable classifier performance is obtained already at low SNR regime.

2. CLASSIFICATION SYSTEM OVERVIEW

Fig. 1 shows the block diagram of the classification system. The goal of the classification system is to classify the received radar signal based on the pulse compression waveform to eight classes: linear frequency modulation, Costas discrete frequency codes, binary phase codes, and Frank, P1, P2, P3, and P4 polyphase codes.

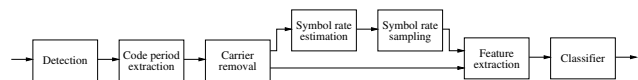


Fig. 1. Block diagram of the classification system.

The first five blocks of the classification system comprise the preprocessing stage. First the radar signal is detected. Then the signal is segmented to code periods, so that the features can be calculated from a signal that consists of only one code period. The carrier frequency is estimated and removed, i.e. the signal is transferred to the baseband. The carrier frequency is defined as the center frequency of the signal's frequency band. These three stages are assumed to be performed. As a result the complex envelope of the received radar signal is obtained.

In order to extract specific information about the phase sequences of the polyphase codes, the symbol rate is estimated and the signal is sampled at the symbol rate. A cyclic correlation based symbol rate estimator is employed [7]. These two stages complete the preprocessing stage of the classification system. However, the symbol rate sampling of the frequency modulated waveforms is not feasible. Hence, the complex envelope is used as such for feature extraction as well.

After the preprocessing stage, the features are extracted. Finally, the extracted features are fed to a supervised classifier that performs the classification.

3. WAVEFORM CLASSIFIER

The classifier structure consists of two independently operating parallel multilayer perceptron (MLP) networks. Fig. 2 illustrates the structure of the waveform classifier.

The classifier structure stems from the hierarchical classifier structure used in [1] for modulation classification of communication signals. However, the outputs of the classifiers used in this

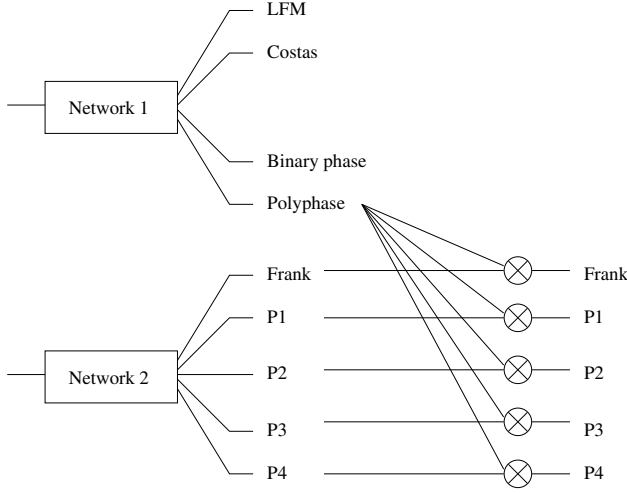


Fig. 2. Waveform classifier structure comprised of two parallel multilayer perceptron networks.

work approximate the posterior probabilities of the classes while in [1] the individual classifiers make a hard decision about the class. Consequently, the combination of the outputs of the classifiers is different as well. The interpretation and combination of the outputs of the networks in Fig. 2 are explained next.

The first three outputs of the network 1 approximate the posterior probabilities of the LFM, Costas, and binary phase classes given the input feature vector of the network 1. The fourth output of the network 1 approximates the combined probability of the polyphase classes given the input feature vector of the network 1. The outputs of the network 2 approximate the probabilities of the polyphase classes given the input feature vector of the network 2 and the assumption that the signal is from one of the polyphase classes. Finally, the posterior probabilities of the polyphase classes given the data pattern are approximated by multiplying the outputs of the network 2 with the output for the polyphase classes from the network 1 (see Fig. 2). The class with the highest posterior probability (i.e. output value) is chosen. The characteristics of the MLPs that ensure that the outputs of the MLPs can be interpreted as the posterior probabilities of the classes are given next.

3.1. Multilayer perceptron

The likelihood function of the data $D = \{(\mathbf{x}^n, \mathbf{t}^n)\}_{n=1}^N$ for classification problems for a network with one output y_k for each class $k = 1, \dots, c$ can be written as [8]

$$p(D|\mathbf{w}) = \prod_{n=1}^N \prod_{k=1}^c (y_k^n)^{t_k^n}, \quad (1)$$

where \mathbf{x}^n is the n th input pattern, \mathbf{t}^n is the corresponding target vector using 1-of- c coding, \mathbf{w} denotes the network parameters, and N is the number of data patterns.

The negative logarithm of the likelihood function (1) is the cross-entropy cost function. The cross-entropy cost function is a suitable choice for classification problems [8].

The corresponding output activation function for the cross-

entropy cost function is the softmax activation function [8, 9]:

$$y_k(\mathbf{x}) = \frac{\exp(a_k(\mathbf{x}))}{\sum_{j=1}^c \exp(a_j(\mathbf{x}))}, \quad (2)$$

where $a_k(\mathbf{x})$ is the induced local field of the output neuron k for an input \mathbf{x} . The induced local field vector $\mathbf{a}(\mathbf{x})$ of an MLP with one hidden layer with hyperbolic tangent activation function is given by

$$\mathbf{a}(\mathbf{x}) = \mathbf{w}_2^T \tanh(\mathbf{w}_1^T \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2, \quad (3)$$

where \mathbf{x} is the input vector, and \mathbf{w}_1 , \mathbf{b}_1 , \mathbf{w}_2 , and \mathbf{b}_2 are the hidden layer weights and biases, and the output layer weights and biases, respectively.

The above choices ensure that the outputs can be interpreted as the posterior probabilities of the classes given the input feature vector [8, 10].

Two different classifiers used in the experiments, the early-stop committee and the Bayesian MLP, that are based on the MLP model given above are presented next.

3.2. Early-stop committee

In early-stopping the training data is divided to two sets: the actual training data set and the validation set. The validation set is used to stop the training before the network overfits to the training data, i.e. when the validation error starts to increase the training is stopped.

Early-stopping can be very sensitive to initial values of the weights. However, by using a committee of early-stop MLPs with different partitions of the data to the training and validation sets for each MLP, the sensitivity to the initial values can be reduced [11]. This makes the early-stop committee (ESC) a robust benchmark method for more complex models.

The output of an early-stop committee using ensemble averaging is given by

$$\mathbf{y}(\mathbf{x}) = \frac{1}{L} \sum_{j=1}^L \mathbf{y}_j(\mathbf{x}), \quad (4)$$

where $\mathbf{y}_j(\mathbf{x})$ is the output vector of the committee member j for an input vector \mathbf{x} , and L is the number of committee members.

The scaled conjugate gradient algorithm [12] was used for training the early-stop MLPs.

3.3. Bayesian MLP

The Bayesian inference for neural networks is based on the posterior distribution of the network weights \mathbf{w} and the hyperparameters α :

$$p(\mathbf{w}, \alpha|D) = \frac{p(D|\mathbf{w})p(\mathbf{w}|\alpha)p(\alpha)}{p(D)} \quad (5)$$

where $p(D|\mathbf{w})$ is the likelihood given in (1), $p(\mathbf{w}|\alpha)$ is the prior distribution of the weights given the hyperparameters, $p(\alpha)$ is the prior distribution of the hyperparameters, and $p(D)$ is a normalization factor.

The predictive distribution of the outputs \mathbf{y} to a new data vector \mathbf{x} given the training data D is used for classification [13]:

$$p(\mathbf{y}|\mathbf{x}, D) = \int p(\mathbf{y}|\mathbf{x}, \mathbf{w})p(\mathbf{w}, \alpha|D)d\mathbf{w}d\alpha. \quad (6)$$

In general, analytic integration of (6) is not possible. One way of approximating the integrals is to use Markov chain Monte Carlo

(MCMC) methods. The idea in MCMC methods is to create Markov chains that produce samples from the posterior distribution of the parameters. In this work, the framework developed by Neal [14], in which the weights are sampled using the hybrid Monte Carlo (HMC) algorithm and the hyperparameters using Gibbs sampling, was used.

The prior for the weights is the Gaussian distribution:

$$\mathbf{w}_j \sim N(\mathbf{0}, \alpha_{w_j}^{-1} \mathbf{I}), \quad j = 1, 2, \quad (7)$$

$$\mathbf{b}_j \sim N(\mathbf{0}, \alpha_{b_j}^{-1} \mathbf{I}), \quad j = 1, 2, \quad (8)$$

where α 's are the precision (inverse of the variance) hyperparameters. The conjugate prior for the hyperparameters is the Gamma distribution [13]:

$$\alpha_j \sim \text{Gamma}(\mu_j, \nu_j) \propto \alpha_j^{\nu_j/2-1} \exp(-\frac{\nu_j}{2\mu_j} \alpha_j), \quad (9)$$

where μ_j is the mean and ν_j is the shape parameter.

The same value $\nu_0 = 0.5$ was used for the shape parameter in all hyperparameter distributions. The choice of μ depended on the distribution: $\mu_{b_1} = \mu_{b_2} = 400$, $\mu_{w_1} = 400N_i^{2/\nu_0}$, and $\mu_{w_2} = 400N_h^{2/\nu_0}$ where N_i and N_h are the number of input and hidden neurons, respectively. The choices correspond to non-informative hyperpriors.

4. FEATURES

A large number of features suitable for recognizing the pulse compression waveforms were explored. These include features based on instantaneous signal properties, second- and higher-order statistics, and time-frequency distributions. The features listed with references are as follows:

- Standard deviation of the instantaneous phase [2] and frequency [3]. The instantaneous frequency was median-filtered as suggested in [6] to suppress the spikes caused by the phase changes in the phase coded signals.
- The bandwidth feature from [15] using symbol rate sampled signal. Thus, rendering the feature to a measure of the autocorrelation sidelobes.
- The difference of the beginning and ending phases of the pulse.
- Power spectral density (PSD) based features: symmetry [4], the maximum of the PSD [4], and the maximum of the PSD of the squared signal.
- Zero-lag moments of the complex envelope. Moments up to eighth-order without any complex conjugated components were used.
- Zero-lag cumulants of the complex envelope. Second- to sixth-order cumulants were used. See [5] for discussion of using cumulants of the complex envelope for classification of digital modulations.
- Diagonal slice of a third-order cumulant of the complex envelope. The lags used were $-2, -1, 1, 2$.
- Time-lag of the maximum cross-correlation between pulse and time-reversed pulse [16].

- Features calculated from the Choi-Williams time-frequency distribution [16] (fixed threshold was used in the normalization algorithm). These features include in total 7 second-, third-, and fourth-order pseudo-Zernike moments as well as 3 other features targeting specific properties observed in the Choi-Williams distributions (CWDs) of the pulse compression waveforms. The three features are: the number of image components in the binary CWD, the time location of the peak power in the CWD, and the standard deviation of the width of the objects in the binary CWD.

The features defined for the network 1 were: the standard deviation of the instantaneous phase and frequency, the PSD features, the first- and second-order zero-lag moments and the second-order zero-lag cumulant of the complex envelope, and the features calculated from the Choi-Williams time-frequency distribution.

For the network 2 the features defined were: all the features defined for the network 1 except the standard deviation of the instantaneous frequency, and the rest of the features which all were calculated from symbol rate sampled signal (i.e. only the new features compared to ones defined for the network 1 were calculated from symbol rate sampled signal).

Total of 18 and 42 features for the networks 1 and 2 were defined, respectively. From these feature sets the final smaller dimensional feature vectors of the networks were selected using a feature selection algorithm introduced in [17]. The algorithm is based on the mutual information between the output classes and the features. The mutual information between the feature vector \mathbf{X} and the classes C is estimated as [17]:

$$\begin{aligned} \hat{I}(\mathbf{X}; C) &= H(C) - \hat{H}(C|\mathbf{X}) \\ &= \sum_{k=1}^c p(c_k) \log p(c_k) \\ &\quad - \frac{1}{N} \sum_{j=1}^N \sum_{k=1}^c \hat{p}(c_k|\mathbf{x}_j) \log \hat{p}(c_k|\mathbf{x}_j), \end{aligned} \quad (10)$$

where $H(\cdot)$ denotes the entropy, c_k is the k th class and c is the number of classes. N is the number of training data samples and \mathbf{x}_j is the j th sample of the training data. Class probabilities are calculated from the training data, i.e. $p(c_k) = N_k/N$ where N_k is the number training samples from class c_k . Probability density estimates $\hat{p}(c_k|\mathbf{x}_j)$ are obtained using the Bayes rule and Parzen windows [17].

The feature selection algorithm uses greedy selection: starting from an empty feature vector, the feature vector size is increased one by one by selecting the new feature that maximizes the estimated mutual information between the classes and the current feature vector (i.e. the previous features plus the new feature). The algorithm can be stopped when the change in the estimated mutual information is small.

For the feature selection a data set consisting of 300 samples per class was generated using the data specifications given in Table 1. Additive white Gaussian noise (AWGN) was added to the signals with a different signal-to-noise ratio randomly selected between 0 and 20 dB for each signal. The signal-to-noise ratio is defined in this work as

$$\text{SNR} = 10 \log_{10} \frac{\sigma_y^2}{\sigma_n^2}, \quad (11)$$

where σ_y^2 is the variance of the original signal, and σ_n^2 is the variance of the noise.

Using the mutual information based feature selection algorithm, the sizes of the feature vectors were reduced to 8 and 12 features, respectively for the networks 1 and 2. Fig. 3 depicts progress of the mutual information feature selection algorithm for selecting features for the network 1. The estimated mutual information starts to saturate after 6 features. However, the algorithm was not stopped until the change was less than 10^{-4} . Consequently, the selected feature vector consisted of 8 features. Fig. 4 shows the progress of the feature selection for the selected 12 features for the network 2 using the aforementioned stopping criterion.

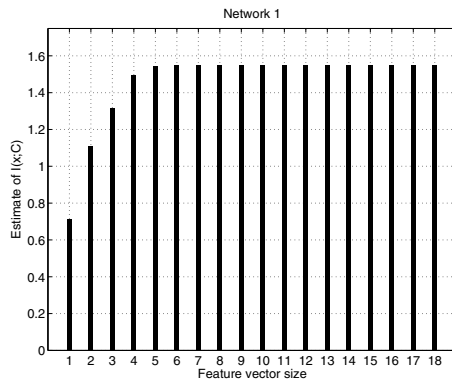


Fig. 3. The progress of the mutual information feature selection algorithm for the network 1. The number of selected features was 8.

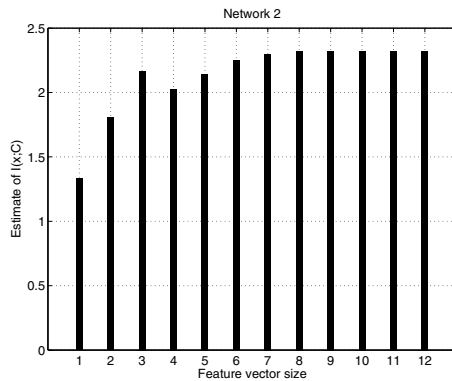


Fig. 4. The progress of the mutual information feature selection algorithm for the network 2 for the selected 12 features. Due to the approximations in the estimation of the probability distributions, the mutual information does not increase monotonically.

The final selected features for the network 1 were: standard deviation of the instantaneous phase, one second-order pseudo-Zernike moment, time location of the peak power in the CWD, standard deviation of the width of the objects in the binary CWD, number of image components in the binary CWD, and the three PSD features.

For the network 2 the selected features were: time-lag of the maximum cross-correlation between pulse and time-reversed pulse, time location of the peak power in the CWD, number of image components in the binary CWD, the three PSD features, standard

Table 1. Simulation parameters.

Parameter	Parameter value
General parameters	
Sampling frequency	12000 Hz
Polyphase codes	
Carrier frequency f_c	Uniform(3000,4000) Hz
Subpulse frequency	$\frac{1}{4}f_c$ when $N \in [3, 7]$ $\frac{1}{2}f_c$ when $N \in [8, 10]$
Frank, P1 codes N [18]	3–10
P2 codes N [18]	4, 6, 8, 10
P3, P4 codes ρ [18]	N^2 where $N \in [3, 10]$
Binary phase codes	
Carrier frequency f_c	Uniform(3000,4000) Hz
Subpulse frequency	$\frac{1}{4}f_c$ or $\frac{1}{2}f_c$
Barker, Quadratic residue sequences, Maximal length sequences, Gold codes	of different lengths
Costas codes	
Number of subpulses N	2, 4, 6
Lowest frequency f_0	500 Hz
Frequency hop Δf	Uniform(500,1000) Hz
Subpulse length τ_1	Uniform(5,10) ms
Linear frequency modulation	
Carrier frequency f_c	3000 Hz
Bandwidth Δf	Uniform(500,3000) Hz
Modulation period τ	Uniform(20,40) ms

deviation of the width of the objects in the binary CWD, one second-order pseudo-Zernike moment, autocorrelation sidelobe measure, standard deviation of the instantaneous phase, third-order cumulant with lag -1 , and one third-order pseudo-Zernike moment.

5. SIMULATION RESULTS

In order to test the classification performance of the proposed classification system and compare different classifier models, two classifier structures were trained and their performance was evaluated as function of the signal-to-noise ratio in additive white Gaussian noise channel using simulated data. One of the classifier structures was based on the early-stop committee model, and the other one was based on the Bayesian MLP model. That is, both of the individual classifiers in the structure (see Fig. 2) had the same model, i.e. either the early-stop committee or the Bayesian MLP. Each one of the individual classifiers had one hidden layer with 30 hidden neurons.

The early-stop committees consisted of 10 MLPs. In early-stop training, 10 % of the data was reserved for validation (different randomly selected set for each MLP). The remaining data patterns were used for training of the MLPs.

The Bayesian learning of the MLPs was carried out with the Flexible Bayesian Modeling (FBM) software¹. The number of iterations was 60000. The trajectory length was 100 and a window size of 10 was used. Step size adjustment factor was 0.04 and the momentum parameter had a value 0.9. For more information about the parameters, see [14] and the FBM software documentation. The network predictions were made using 100 evenly separated samples from the last 2000 iterations.

Table 1 lists the parameters used for generating both the training and testing data sets. The training data set consisted of 600

¹<http://www.cs.toronto.edu/~radford/fbm.software.html>

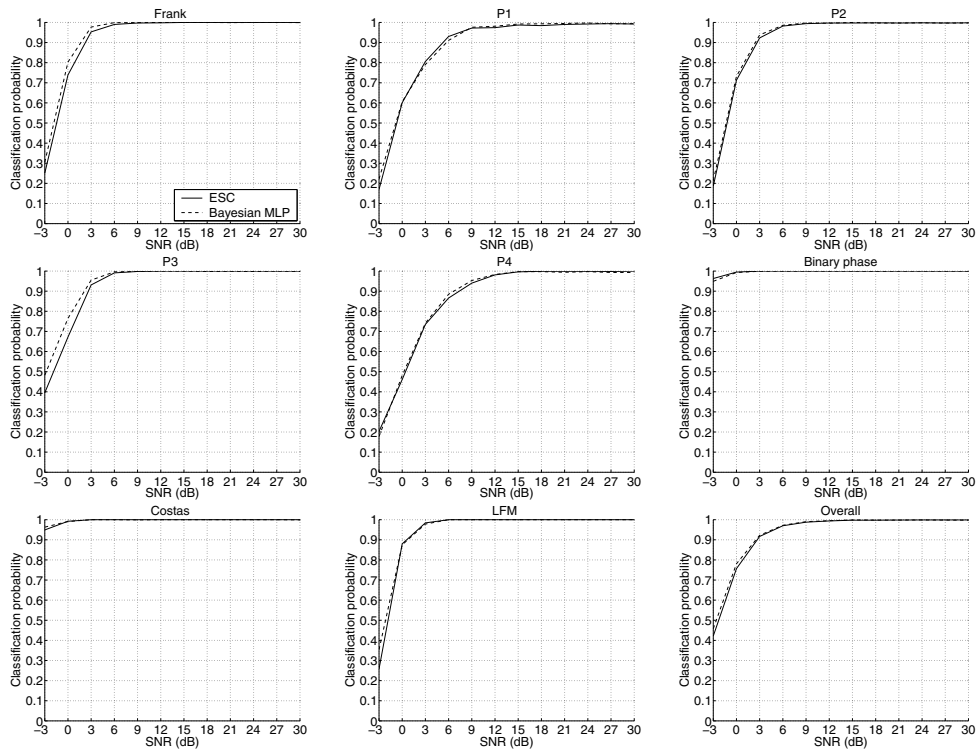


Fig. 5. Classification performance as a function of the SNR on data similar to the training data. The classification system has reliable performance: overall correct classification rate of 97 % is achieved at SNR of 6 dB with both classifier models.

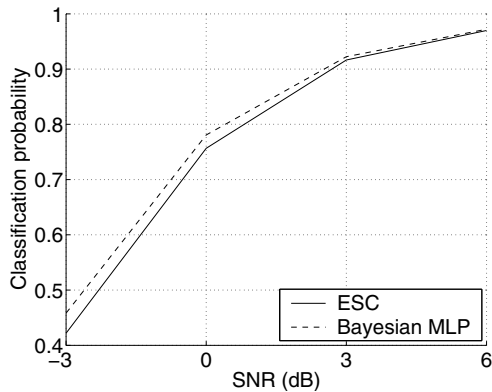


Fig. 6. The overall performances of the two classifier structures at low SNRs.

samples per class for different SNRs between 0 and 20 dB. This ensures that the classifier is able to operate in both noisy and noiseless environments. For testing purposes 1000 samples per class were generated for SNRs of $-3, 0, 3, \dots, 30$ dB. The data was normalized using simple linear rescaling, i.e. the mean was first subtracted and the result was divided by the standard deviation. Estimates for the mean and standard deviation were calculated from the training data.

Fig. 5 depicts the classification performance of the two classifier structures as a function of the signal-to-noise ratio. In the

figure the classification probabilities of the individual classes as well as the overall correct classification probability are plotted as a function of the signal-to-noise ratio. The overall correct classification rate of both classifier structures reaches 97 % at SNR of 6 dB. Both classifier structures have comparable performance at SNR of 6 dB and higher. The differences between the performances are perceived at SNRs below 6 dB. To highlight the differences, the overall performances at low SNRs are plotted in Fig. 6.

Fig. 5 shows that the binary phase and the Costas codes are reliably classified already at very low SNRs: the correct classification rate for the binary phase and the Costas codes is well over 90 % already at -3 dB. The most difficult codes to classify correctly appear to be the P1 and P4 codes. Table 2 reports the classification rates of the Bayesian MLP structure at SNR of 6 dB. It can be seen that the P1 and P4 codes are the most difficult to be discriminated from each other. This is also highlighted by the fact that even at high signal-to-noise ratios roughly 1 % of the P1 codes are incorrectly classified to the P4 codes. Although the P1 codes are derived from a step approximation to a linear frequency modulation while the P4 codes are derived directly from linear frequency modulation, the P1 and P4 codes are very similar. This is due to the fact that both codes have the largest phase increments in the ends of the code and the smallest in the center of the code. Further work on the features is required to improve the discrimination between the P1 and P4 codes.

Table 2. Confusion matrix for the Bayesian MLP at SNR of 6 dB. The overall correct classification rate was 97.2 %.

	Frank	P1	P2	P3	P4	Binary	Costas	LFM
Frank	99.9	0.1	0	0	0	0	0	0
P1	0	91.1	0.1	0	8.6	0	0	0.2
P2	0	0.6	98.6	0	0	0	0	0.8
P3	0	0	0	99.6	0	0	0	0.4
P4	0	8.0	0	0.1	88.5	0	0	3.4
Binary	0	0	0	0	0	100	0	0
Costas	0	0	0	0	0	0	100	0
LFM	0	0	0	0	0	0	0	100

6. CONCLUSION

In this paper the recognition of pulse compression radar waveforms was studied. A classification system was introduced and two supervised classifier models were tested: the early-stop committee and the Bayesian MLP. Both classifiers performed reliably on data similar to the training data: the overall correct classification rate reached 97 % at SNR of 6 dB for both classifiers. The Bayesian MLP slightly outperformed the early-stop committee at SNRs lower than 6 dB.

It was found that the classification performance for the P1 and P4 codes is worse than for the other codes because the statistical properties of the P1 and P4 codes are very similar. Future work is underway to find features that discriminate the P1 and P4 codes more effectively.

7. REFERENCES

- [1] C. Louis and P. Sehier, "Automatic modulation recognition with a hierarchical neural network," in *Military Communications Conference*, October 1994, vol. 3, pp. 713–717.
- [2] A. K. Nandi and E. E. Azzouz, "Automatic analogue modulation recognition," *Signal Processing*, vol. 46, no. 2, pp. 211–222, October 1995.
- [3] M. L. D. Wong and A. K. Nandi, "Automatic digital modulation recognition using artificial neural network and genetic algorithm," *Signal Processing*, vol. 84, no. 2, pp. 351–365, February 2004.
- [4] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 431–436, April 1998.
- [5] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 416–429, March 2000.
- [6] G. López-Risueño, J. Grajal, and O. Yeste-Ojeda, "Atomic decomposition-based radar complex signal interception," *IEE Proc. – Radar Sonar Navig.*, vol. 150, no. 4, pp. 323–331, August 2003.
- [7] P. Ciblat, P. Loubaton, E. Serpedin, and G. B. Giannakis, "Asymptotic analysis of blind cyclic correlation-based symbol-rate estimators," *IEEE Trans. Inform. Theory*, vol. 48, no. 7, pp. 1922–1934, July 2002.
- [8] C. M. Bishop, *Neural networks for pattern recognition*, Oxford University Press, 1995.
- [9] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing: Algorithms, Architectures and Applications*, F. Fogelman Soulié and J. Héroult, Eds., pp. 227–236. Springer-Verlag, 1990.
- [10] J. B. Hampshire II and B. A. Pearlmutter, "Equivalence proofs for multi-layer perceptron classifiers and the Bayesian discriminant function," in *Proc. 1990 Connectionist Models Summer School*, D. Touretzky, J. Elman, T. Sejnowski, and G. Hinton, Eds. 1990, pp. 159–172, Morgan Kaufmann, San Mateo, CA.
- [11] J. Lampinen and A. Vehtari, "Bayesian approach for neural networks — review and case studies," *Neural Networks*, vol. 14, no. 3, pp. 257–274, April 2001.
- [12] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.
- [13] D. Husmeier, W. D. Penny, and S. J. Roberts, "An empirical evaluation of Bayesian sampling with hybrid Monte Carlo for training neural network classifiers," *Neural Networks*, vol. 12, no. 4–5, pp. 677–705, June 1999.
- [14] R. M. Neal, *Bayesian learning for neural networks*, vol. 118 of *Lecture Notes in Statistics*, Springer-Verlag, 1996.
- [15] S. J. Roome, "Classification of radar signals in modulation domain," *Electronics Lett.*, vol. 28, no. 8, pp. 704–705, April 1992.
- [16] J. Lundén, L. Terho, and V. Koivunen, "Classifying pulse compression radar waveforms using time-frequency distributions," in *Proc. 39th Annual Conf. Information Sciences and Systems (CISS 2005)*, March 2005.
- [17] N. Kwak and C.-H. Choi, "Input feature selection by mutual information based on Parzen window," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 12, pp. 1667–1671, December 2002.
- [18] B. L. Lewis, F. F. Kretschmer, Jr., and W. W. Shelton, *Aspects of radar signal processing*, Artech House, Inc., 1986.