Proceedings of the 2009 International Conference on Wavelet Analysis and Pattern Recognition, Baoding, 12-15 July 2009

RADAR TRANSMITTER CLASSIFICATION USING A NON-STATIONARY SIGNAL CLASSIFIER

MARTHINUS C. DU PLESSIS[1], JAN C. OLIVIER[2]

[1]Dept. of Electrical, Electronic and Computer Engineering, University of Pretoria, South Africa
[2]University of Pretoria and Defence Research, CSIR, Pretoria, South Africa
E-MAIL: christod@tuks.co.za, corne.olivier@up.ac.za

Abstract

This paper presents a classification method which discriminates between two radar transmitters based on the received pulses. A simple radar transmitter model is presented to which a non-stationary signal classifier is applied. The classifier is a support vector machine which is applied to the radar pulse's time-frequency representation. The time-frequency representation is refined using particle swarm optimization to increase the classification accuracy. The classification accuracy is tested in an additive white Gaussian noise channel. An acceptable classification accuracy is reported for component tolerances as small as 2% on the transmitter's modulator.

Keywords:

Classification; radar transmitter; non-stationary classification; time-frequency representation.

1. Introduction

With radar transmitter classification the classifier needs to distinguish between different radar transmitters of the same model and manufacturer based on the received pulses. The differences between two radar transmitters of identical model and make may be due to component tolerances in the pulse forming network (PFN), tolerances on the oscillator and small load mismatches between the oscillator and the antenna. Changes for this signal occur in both time and frequency which makes the classification challenging. A signal classifier and a radar transmitter model are presented in this paper. The classification performance on the model in an additive white Gaussian noise (AWGN) channel is given in the final section.

2. Signal classifier

The classifier is based on the classifier described in [1] with several significant changes. The classifier calculates a time-frequency representation (TFR) of the radar pulse which is used as an input to a support vector machine (SVM). The time-frequency representation is refined based on the expected classification performance.

2.1. Time-frequency representation

The time-frequency representation was calculated by using a Cohen class TFR. This is defined for the signal $x(t)$ as [2]

$$C_x^{\phi}(t,f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A_x(\xi,\tau)\,\phi(\xi,\tau)\,e^{2\pi j(t\xi - f\tau)}d\xi d\tau \quad (1)$$

where $\phi(\xi,\tau)$ is the kernel function. The above TFR was normalised before classification as [1]

$$NC_x^{\phi}(t,f) = \frac{\left|C_x^{\phi}(t,f)\right|}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left|C_x^{\phi}(t,f)\right|dtdf} \quad (2)$$

The narrowband ambiguity function is defined as

$$A_x(\xi,\tau) = \int_{-\infty}^{\infty} x\left(s+\frac{\tau}{2}\right)x^*\left(s-\frac{\tau}{2}\right)e^{-j2\pi\xi s}ds. \quad (3)$$

The kernel function defines the properties of the resulting distribution and several common distributions (such as the Wigner-Ville distribution) can be obtained for specific kernel functions. The radially Gaussian kernel function is defined as [3]

$$\phi(\xi,\tau) = e^{\frac{-\left(\xi^2+\tau^2\right)}{2\sigma(\theta)^2}} \quad (4)$$

978-1-4244-3729-0/09/$25.00 ©2009 IEEE

where $\theta = \arctan\left(\frac{\xi}{\tau}\right)$. The function $\sigma(\theta)$ is the spread function that completely defines the radially Gaussian kernel. Thus the search for the optimal kernel is equivalent to the search for the optimal spread function. To ensure that the resulting distribution is real the kernel function must satisfy $\phi(\xi,\tau) = \phi^*(-\xi,-\tau)$. This requirement is met for the radially Gaussian kernel if the spread function is π-periodic. Another desirable property of the kernel function is to be smooth [3]. This can be accomplished by requiring that the spread function is both continuous and differentiable.

A Bernstein expansion was used as a spread function (as opposed to the truncated Fourier series used in [1]). It was found that the optimization with particle swarm optimization (PSO) was faster than the truncated Fourier series case. The Bernstein expansion is defined as

$$B_n(t) = \sum_{i=0}^{n} \beta_i B_{i,n}(t) \tag{5}$$

where the Bernstein polynomials are

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}. \tag{6}$$

The spread function is therefore

$$\sigma(\theta) = \begin{cases} B_n(\theta/\pi) & 0 \le \theta \le \pi \\ B_n(\theta/\pi - 1) & \pi < \theta \le 2\pi \end{cases} \tag{7}$$

To ensure that the spread function is smooth (continuous and differentiable) the following linear constraints on the Bernstein coefficients must be met

$$\beta_0 = \beta_n \tag{8}$$

$$\beta_1 - \beta_0 = \beta_n - \beta_{n-1} \tag{9}$$

2.2. Criterion function

The ν-SVM was used as a classifier. The discriminant function is defined as

$$g(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b \tag{10}$$

where $k(\mathbf{x}, \mathbf{x}_i)$ is the SVM kernel function. The class label will therefore be $sgn(g(\mathbf{x}))$. The advantage of using a ν-SVM formulation over the standard C-SVM is that the choice of the ν parameter is more intuitive. The ν parameter is usually set to the expected error rate [4]. The Gaussian radial basis function kernel for the SVM is given as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma^2}\right) \tag{11}$$

The parameter vector $\boldsymbol{\theta} = [\beta_0\ \beta_1\ \ldots\ \beta_n\ \sigma]$ needs to be optimized for the classifier (if a linear kernel function is used σ is omitted). The output of the SVM is modelled as a Gaussian distribution (the normal assumption is supported in [1]). The training set $\mathcal{O}_N$ is randomly divided into two sets $\mathcal{O}_T$ and $\mathcal{O}_L$ – the $\mathcal{O}_L$ set is used to train the SVM and the $\mathcal{O}_T$ set is used to evaluate the performance. The probability of error for this is

$$\begin{aligned} K(\boldsymbol{\theta}|\mathcal{O}_L, \mathcal{O}_T) = \frac{1}{2} &+ \frac{1}{4} erfc\left(\frac{-\mu_-}{\sqrt{2}\sigma_-}\right) \\ &- \frac{1}{4} erfc\left(\frac{-\mu_+}{\sqrt{2}\sigma_+}\right) \end{aligned} \tag{12}$$

Where μ_-, μ_+, σ_- and σ_+ are the mean and standard deviation for the positive and negative classes in $\mathcal{O}_T$ respectively. The above procedure is repeated R times and the average is taken as the final criterion $K(\boldsymbol{\theta}|\mathcal{O}_N)$ [1]. When the optimal value for the parameter vector is found the SVM is trained on the full set $\mathcal{O}_N$.

2.3. Particle swarm optimization

Particle swarm optimization was used instead of a local search algorithm to calculate the optimal parameter vector for the criterion function. $\mathbf{x}_i(t)$ is the position of the ith particle at timestep t. The global best PSO is performed as follows [5]:

1. Initialize the swarm of particles so that each particle $\mathbf{x}_i(t)$ is random within the search space (at $t = 0$).
2. Evaluate the fitness function of each particle at its current position: $f(\mathbf{x}_i(t))$.
3. Calculate the best individual positions. If $f(\mathbf{x}_i(t)) < pbest_i$ then $pbest_i = f(\mathbf{x}_i(t))$ and $\mathbf{x}_{pbest_i} = \mathbf{x}_i(t)$.
4. Calculate the global best value. If $f(\mathbf{x}_i(t)) < gbest$ then $gbest = f(\mathbf{x}_i(t))$ and $\mathbf{x}_{gbest} = \mathbf{x}_i(t)$.
5. Calculate the velocity vector for each particle:

$$\begin{aligned} \mathbf{v}_i(t+1) = &w\mathbf{v}_i(t) + \rho_1(\mathbf{x}_{pbest} - \mathbf{x}_i(t)) \\ &+ \rho_2(\mathbf{x}_{gbest} - \mathbf{x}_i(t)) \end{aligned} \tag{13}$$

6. Update the position of the particle based on the velocity.

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{14}$$

7. Check if the convergence criterion is met - if not, go to step 2.

In general a set of linear equality constraints on the optimization problem can be expressed as

$$A\mathbf{x} = \mathbf{b} \tag{15}$$

If the initial particle positions satisfy the constraints and the initial velocity is zero then multiplying the velocity update by A gives

$$\begin{aligned} A\mathbf{v}_i(t) =& Aw\mathbf{v}_i(t-1) + \rho_1\left(A\mathbf{x}_{pbest} - A\mathbf{x}_i(t)\right) \\ &+ \rho_2\left(A\mathbf{x}_{gbest} - A\mathbf{x}_i(t)\right) \end{aligned} \tag{16}$$

which is zero. Any solution obtained by the PSO is thus feasible if the starting particles were feasible. This allows the constraints on the Bernstein expansion to be used with no additional overhead.

3. Transmitter model

A simple transmitter model was created that consists of a pulse-forming network and an oscillator. The transmitter model is not intended to be representative of all types of pulsed radar transmitters, its main use is to test the performance of the classifier. The only differences modelled between two transmitters are the tolerances on the components in the PFN.

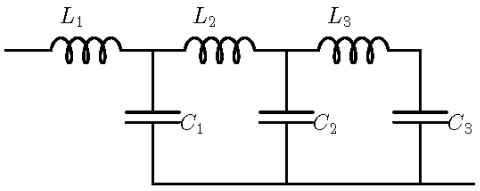

Figure 1. Structure of a pulse forming network

The current and voltage relationship in a magnetron is highly non-linear. The current through a magnetron is small until a certain voltage (Hartree voltage) is reached [6]. The current and voltage characteristics can be modelled as an ideal biased diode [7]. A Guillemin E-type network is illustrated in figure 1. This network is charged to a voltage V and then discharged into a load at voltage $\frac{V}{2}$. For this type of network all the components are designed to have identical values. The design equations are readily avaliable (e.g. [6], [7]) and will not be repeated here.

The system can be represented as a system of nonhomogeneous differential equations when the applied voltage $v(t)$ is higher than the turn-on voltage of the biased diode. A pulse-forming network with n sections can be modelled by a system of equations of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b} \tag{17}$$

where

$$\mathbf{x}(t) = \begin{bmatrix} i_1(t) \\ v_1(t) \\ i_2(t) \\ v_2(t) \\ \dots \\ i_n(t) \\ v_n(t) \end{bmatrix} \tag{18}$$

The notation for the the currents and voltages is as follows; the current flowing through L_i is $i_i(t)$ and the voltage across C_i is $v_i(t)$.

The constant matrix for a 3 section PFN is

$$\mathbf{A} = \begin{bmatrix} -\frac{R_\ell}{L_1} & -\frac{1}{L_1} & 0 & 0 & 0 & 0 \\ \frac{1}{C_1} & 0 & -\frac{1}{C_1} & 0 & 0 & 0 \\ 0 & \frac{1}{L_2} & 0 & -\frac{1}{L_2} & 0 & 0 \\ 0 & 0 & \frac{1}{C_2} & 0 & -\frac{1}{C_2} & 0 \\ 0 & 0 & 0 & \frac{1}{L_3} & 0 & -\frac{1}{L_3} \\ 0 & 0 & 0 & 0 & \frac{1}{C_3} & 0 \end{bmatrix} \tag{19}$$

and the nonhomogeneous term is

$$\mathbf{b} = \begin{bmatrix} \frac{V_s}{L_1} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{20}$$

The solution of the system is

$$\mathbf{x}(t) = c_1 e^{\lambda_1 t}\mathbf{u}_1 + c_2 e^{\lambda_2 t}\mathbf{u}_2 + \ldots + c_{2n} e^{\lambda_{2n} t}\mathbf{u}_{2n} - \mathbf{A}^{-1}\mathbf{b} \tag{21}$$

where λ_i and $\mathbf{u}_i$ are the eigenvalues and eigenvectors of $\mathbf{A}$. The constant values $c_1, c_2, \ldots, c_{2n}$ are calculated from the initial conditions (i.e. fully charged network).

The oscillator was modelled as a cosine signal of random phase with a small amount of frequency pushing. Frequency pushing occurs when the oscillating frequency in some oscillators (such as magnetrons) is dependent on the input current [8]. The frequency pushing was modelled as the input current multiplied be the pushing figure (MHz/A) over a small range of currents. The power of the oscillator is directly proportional to the power dissipated by the ideal biased diode model.

4. Results

4.1. Chirp signal

The classifier was tested against the chirp classification problem presented in [9]. The classifier was trained with a 100 chirp signals equally divided between the two classes. Each chirp signal contained 128 samples and the analytic form of the chirp signal was used. The performance is summarized in table 1. As can be seen from the results, the changes made to the original algorithm improved the results.

Table 1. Error rates on the linear chirp classification problem

Classifier	Error rate
Optimal kernel [9]	2.25%
SVM with linear kernel [1]	1.98%
SVM with linear kernel & PSO (this paper)	1.59%
SVM with RBF kernel [1]	1.40%
SVM with RBF kernel & PSO (this paper)	0.93%

4.2. Pulse in an AWGN channel

A series of $1\mu s$ pulses was generated by the transmitter model with a 7 section pulse forming network. The signal was downmixed to an intermediate frequency and sampled at 128 samples per pulse. Additive white Gaussian noise was added to the training vectors to achieve an SNR of 20dB in the 2% tolerance case and a SNR of 10dB in the 5% tolerance case. The training set in each case consisted of 60 pulses per class.

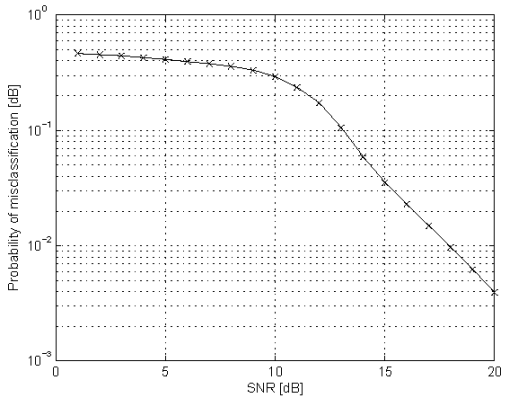

Figure 2. Classification accuracy for a 2% tolerance in an AWGN channel

5. Conclusion

A radar transmitter model and an associated signal classifier has been presented. By using a different spread function and a global search algorithm the performance of the classifier was improved for the chirp signal benchmark. The signal classifier was tested on a pulse generated by the radar transmitter model. A high classification accuracy was achieved for relatively low component tolerances.

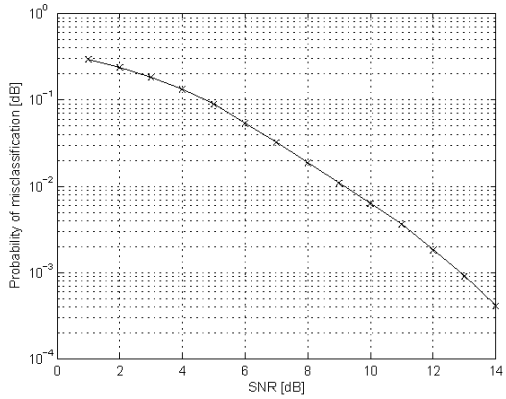

Figure 3. Classification accuracy for a 5% tolerance in an AWGN channel

References

[1] Manuel Davy, Arthur Gretton, Arnaud Doucet, and Peter J.W. Rayner, "Optimized support vector machines for nonstationary signal classification", IEEE Signal Processing Letters, Vol 9, No.12, pp. 442–445, Dec. 2002.

[2] Francois Auger and Franz Hlawatsch, "Time-Frequency analysis", Wiley-ISTE, London, 2008.

[3] Richard G. Baraniuk and Douglas L. Jones, "Signal-dependent time-frequency analysis using a radially gaussian kernel", Signal Processing, Vol 32, pp 263–284, 1993.

[4] Bernhard Scholkopf and Alexander J. Smola, "Learning with Kernels - Support Vector Machines, Regularization, Optimization and Beyond", MIT Press, Cambridge (MA), 2002.

[5] Andries P. Engelbrecht, "Computational Intelligence - An Introduction", Wiley, New York, 2003.

[6] George W. Ewell, "Radar transmitters", McGraw-Hill, New York, 1981.

[7] Norris G. Glasoe and J.V. Lebacqz, "Pulse Generators", MIT Radiation Laboratory Series, McGraw Hill, 1949.

[8] Merril I. Skolnik, "Radar Handbook", McGraw-Hill, New York, 1970.

[9] Manuel Davy and Christian Doncarli, "Optimal kernels of time-frequency representations for signal classification", Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, pp 581–584, Oct 1998.