IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 10, Issue 6, Ver. I (Nov - Dec .2015), PP 84-90
www.iosrjournals.org

A Novel Approach of Poly-semantic Radar Signal to Neural Network

Mohammed Moazzam Moinuddin
Associate Professor, Dept. of ECE, Maulana Azad National Urdu University Polytechnic Bangalore, India,

Abstract—*Many pulse radar applications, target detection becomes very difficult when the echo consists of high background noise due to clutter. In this paper to overcome this limitation and to enhance detection performance a novel Radial basis neural network (RBNN) filter based on radial basis function neural network is developed. The RBNN filter is optimized by using computer search with poly-semantic codes as input. The filter is implemented by replacing the matched filters in the digital phase coded pulse compression system.*
Keywords - *Target detection, Radial basis neural network, Poly-semantic codes, Signal-to-sidelobe ratio, Integrated-sidelobe level.*

I. INTRODUCTION

The original inspiration for the Artificial Neural Network (ANN) technique [1], was from examination of the central nervous system and the neurons (axons, dendrites and synapse) which constitute one of its most significant information processing elements. In ANN model, a large number of neurons (processing elements) are connected together to form a network of nodes — hence the term neural networks. Parallel processing methods are used to solve many tasks where it is very difficult to define a conventional algorithm. Some of the applications of neural networks are Voice recognition, Data compression, Radar signature analysis, Radar imaging and others. All these problems involve large amounts of data, and complex relationships between the different parameters. Neural Networks can be used as matched filters to compress the coded waveforms at the receiver.

II. RBNN FILTER DESIGN

RBNN filter is designed by simulation model of radial basis function neural network using MATLAB functions [2-4], as shown in Fig.1. It consists of a hidden layer of radial basis neurons and an output layer of linear neurons. The typical shape of a radial basis transfer function used by the hidden layer is a nonlinear Gaussian pulse. Each linear output neuron forms a weighted sum of these radial basis functions.

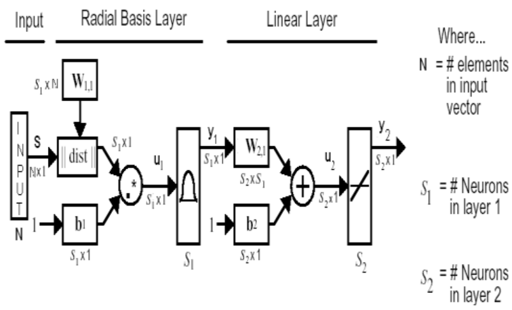


Fig.1 Architecture of Radial Basis Function Neural Network.

The function, *newrb* algorithm is used to create a two layer network. The first layer has *radbas* neurons, and calculates its weighted inputs with *dist* function, and its *net* input with *netprod* function. The second layer has *purelin* neurons, and calculates its weighted inputs with *dotprod* and its *net* inputs with *netsum*. Both the layers have biases**.** Each bias in the first layer is set to sqrt(-log(0.5))/spread.

The mathematical expressions [4] for each layer are given by

$$u_1 = \|W_{1,1} - S\| .* b_1 \quad (1)$$

$$y_1 = radbas(u_1) = \varphi(u_1) = e^{-\left(\frac{u_1}{\sigma}\right)^2} \quad (2)$$

$$u_2 = (W_{2,1} \cdot y_1 + b_2) \quad (3)$$

$$y_2 = purelin(u_2) = f(net) \quad (4)$$

where S is the input vector of length N, $W_{1,1}$ is the input weight matrix, $W_{2,1}$ is the hidden layer weight matrix, b_1 is the input bias vector, b_2 is the output bias vector, $\sigma = \frac{l}{\sqrt{2N}}$ is the spread of each radial basis function and *l* is the maximum Euclidean distance between any two centers.

During off-line training, the network minimizes the Sum Squared Error (SSE) on the training set to obtain optimum weight matrix.

The SSE is $\varepsilon = \frac{1}{2}\sum_{k=1}^{N}\left(d_k - y_{2,k}\right)^2$ (5)

where d_k, k = 1,2, … ,N is a desired response.

Training: The network is trained in off-line with the desired response by calling the function *net* = *newrb*(y_2, d, goal, spread). Initially the *radbas* layer has no neurons. Initial weights are taken as null vector.

At each iteration (called epoch), the input vector which results in lowering the network error, it is used to create a *radbas* neuron. The error of the new network is checked, and if that is low enough, then the process of *newrb* is completed. Otherwise, the next neuron in the hidden layer is added and weights are updated. This process is repeated until either the goal is met or the maximum number of neurons is reached.

Output: The output is obtained on-line by using *sim* function with a given input pattern of the received signal vector R. The function call is *output = sim(net, R).* The network with a SIMULINK block, *gensim(net, R)* is used to generate a simulation model for the trained network.

Consider an optimal binary code

$$S_N = [s_1, s_2, s_3, s_4, \ \ldots \ , s_{N-1}, s_N] \qquad (6)$$

If the binary sequence of length N is the input, then the time shifted input sequence patterns presented at the input nodes are represented in the matrix form as

$$[S] = \begin{bmatrix} s_1 & 0 & 0 & . & . & . & 0 & 0 \\ s_2 & s_1 & 0 & . & . & . & 0 & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ s_{N-1} & s_{N-2} & s_{N-3} & . & . & . & s_1 & 0 \\ s_N & s_{N-1} & s_{N-2} & . & . & . & s_2 & s_1 \\ 0 & s_N & s_{N-1} & . & . & . & s_3 & s_2 \\ 0 & 0 & s_N & . & . & . & s_4 & s_3 \\ \dot{0} & \dot{0} & \dot{0} & \dot{0} & & & \dot{0} & \dot{s}_N \\ 0 & 0 & 0 & 0 & & & 0 & 0 \end{bmatrix} \qquad (7)$$

The matrix [S] of dimension N by 2N consists of 2N sequence patterns including all zero patterns {0} which represents no input signal. Subsequently, there are 2N different sequence patterns presented at the input node in each epoch. The weight co-efficient and the number of hidden layer neurons were updated during training period, in order to achieve the desired response $(d_i, \ i = 1,2,3, \ \ldots \ , 2N)$. The '$d_i$' is set to 'zero' for all shifted patterns of the input sequence including all zero pattern (no input) and 'N' for zero-shifted input pattern which is applied at N^{th} input. That is, the desired response is $d = [0,0,0, \ \ldots \ , N, \ \ldots \ , 0,0,0]$, which gives ideal pulse compression characteristics with peak main lobe amplitude as N and zero side lobe levels. In the on-line process, the received sequence of length N is presented directly at the input layer of the trained network for pulse compression and side lobe suppression.

III. PERFORMANCE OPTIMIZATION

The performance optimization of receiver system is carried out by using poly-semantic codes as the input so that the output of the conventional filter yields optimum Signal-to-Sidelobe Ratio (SSR) and Integrated Sidelobe Level (ISL) values. But, when these optimum sequences are applied on filters based on neural networks, the filter output may not yield optimum values of SSR and ISL at a given length since the output performance depends on both the type of algorithm and the network parameters. When radial basis function neural network algorithm is employed, the default values of the parameters, the goal (mean squared error) and the spread (spread of radial basis functions) are generally taken as 0.0 and 1.0 respectively. These values do not yield optimum output values for specific applications. For good design, a spread should be selected such that it is larger than the distance between adjacent input vectors and smaller than the distance across the whole input space. In general, the goal is set at less than the error value of 10^{-07}.

In the present application where the Correlated Radial Basis Neural Network Sidelobe Suppression (CRBNSS) filter is used as a radar pulse compression and side lobe suppression filter, the output waveform is not an autocorrelation function. Hence, the filter output is trained as a correlated main lobe at zero time lag and mismatched sidelobes at other time lags, similar to the autocorrelation function. So, in the output waveform, it is considered that the main lobe amplitude as r(0) and sidelobe amplitudes as r(k), k =-N+1, -N+2, … , N-2, N-1. Since r(0) is set at 'N' in the desired response, no loss in SNR occurs during filtering. The spread of the network depends on the pattern and length of the input vector sequence. The optimization technique for CRBNSS filter is developed. The computer search to obtain a spread value called optimum spread value of the neural network which yields maximum SSR and minimum ISL. To obtain optimum spread values of the network, initially the network is trained with all combinations of input sequence patterns at each code length by varying the spread value of the network.

IV. SPREAD SELECTION FOR CRBNSS FILTER

The poly-semantic sequences of length 9 to 189 are used as the input sequences for the CRBNSS filter in order to evaluate the performance in terms of SSR and ISL. The SSR and ISL values at the output of the CRBNSS filter with different spread values of the network are calculated. The details of maximum SSR and minimum ISL values are tabulated in Table 1. The filter yields maximum SSR and minimum ISL values at particular spread value of the network at each code length. These spread values are called optimum spread values of the neural network. The optimum spread value is the spread of the network at which SSR value becomes maximum. For example Table 2 shows the SSR value at different spread values of the network for poly-semantic code of length 27. It is found that the network yields maximum SSR value of 354.52 at spread 0.6. Thus, for code length 27, the optimum spread value is 0.6. Similarly, for all code lengths the spread value can be obtained. All the codes resulted in maximum SSR value above 300 dB and minimum ISL value below -300 dB at corresponding optimum spread values.
It is also found that as spread value of the network increases from 0.1, the SSR value is increasing. At a particular spread value, the SSR becomes maximum and further SSR decreases as spread increases. This is because at high spread values, the neural network generates more side lobes.

Table 1 SSR and ISL values for poly-semantic codes at optimum spread values of CRBNSS filter.

Code Length	Spread	SSR (in dB)	ISL (in dB)
9	0.5	323.04	-313.52
18	0.5	341.67	-320.22
27	0.6	354.52	-331.17
36	0.7	355.77	-331.16
45	0.8	351.68	-326.20
54	0.9	337.32	-311.05
63	1.0	348.42	-321.48
72	1.2	327.86	-300.09
81	1.2	349.95	-320.43
90	1.3	328.74	-300.15
99	1.3	355.75	-325.66
108	1.3	328.72	-299.44
117	1.3	339.35	-309.72
126	1.5	335.59	-305.24
135	1.6	331.10	-303.71
144	1.6	337.14	-306.06
153	1.6	344.56	-313.76
162	1.7	335.55	-304.33
171	1.8	336.09	-303.72
180	1.8	337.07	-304.85
189	1.8	354.01	-324.61

Table 2 SSR values at different spreads of the network for code length 27.

S.No	Spread	SSR in dB
1	0.1	337.70
2	0.2	330.63
3	0.3	332.73
4	0.4	339.94
5	0.5	346.30
6	0.6	354.52
7	0.7	335.51
8	0.8	335.33
9	0.9	303.44
10	1.0	265.72

V. PERFORMANCE EVALUATION AND SIMULATION RESULTS

A. Noise Robustness

To evaluate the noise performance [5,6], the dispersed echo signals from the targets are modeled by considering all coded waveforms perturbed by Additive White Gaussian Noise (AWGN). MATLAB simulation command Y = *awgn(X, SNR)* is used for adding white Gaussian noise to the signal X. The SNR is in dB. The power of X is assumed as 0 dBW. If X is complex, then AWGN adds complex noise. In simulation, different noise levels are obtained by varying the noise strengths at different SNR values from 10 dB to -10 dB. The SSR and ISL are calculated by averaging these values over 100 runs at the output of the filter.

Fig. 2 and Fig. 3 show the variations of SSR and ISL values respectively for poly-semantic codes of lengths from 9 to 189 at optimum spread values of the network in different noisy environments with noise strengths of SNR -10dB, -5dB, 0dB, 5dB, and 10dB.

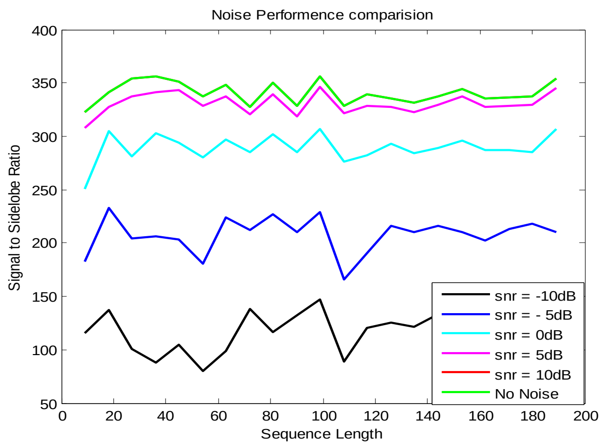


Fig.2 Variations of SSR for poly-semantic codes at optimum spread values of the network in noisy environment.

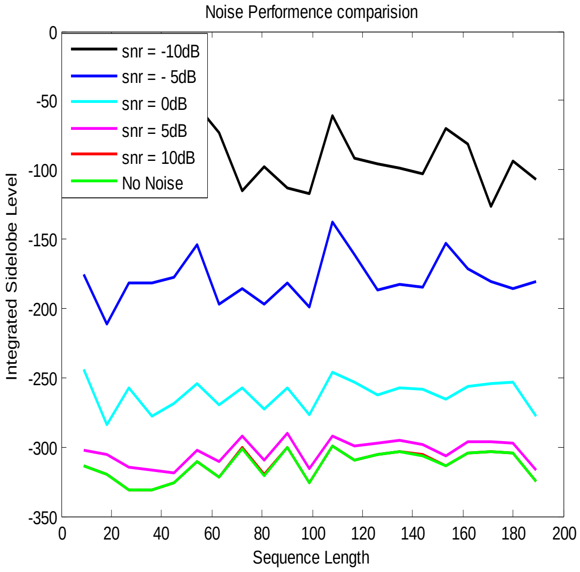


Fig.3 Variations of ISL for poly-semantic codes at optimum spread values of the network in noisy environment.

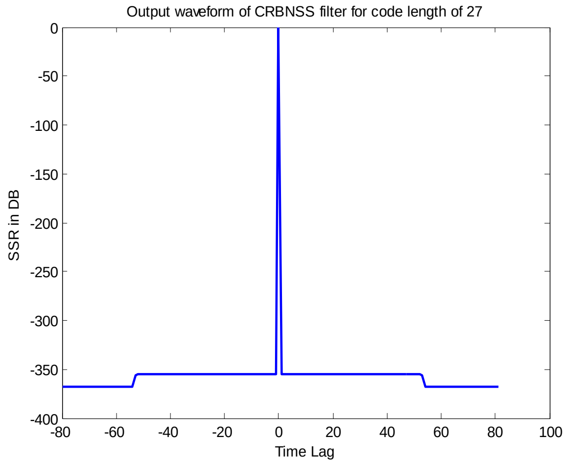
It is found that the codes of lengths 27, 36, 99, 189 in noisy environment yield very high SSR and very low ISL values compared to other codes. Also, in noisy environment no change is observed in the optimum spread values of the network for each code. At other spread values, the SSR values decrease to very low values. In the presence of noise, when the SNR increases from no noise to -10 dB, the maximum SSR and minimum ISL values degrade considerably. These values are tabulated in Table 3. The average SSR values degrade from 354.52 dB to 136.16 dB and the average ISL values degrade from -331.17 dB to -106.77 dB. Therefore, the filter results in optimal performance only at specified spread values of the network for a given code.

Table 3 SSR and ISL values at different noise strengths for poly-semantic codes of lengths 27, 36, 99, 189

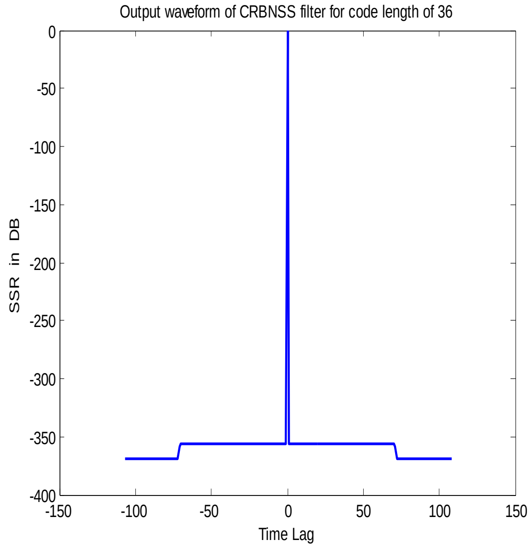
Code	**Spread**	**SSR(in dB) / ISL(in dB)**					
		Without Noise	**SNR 10dB**	**SNR 5dB**	**SNR 0dB**	**SNR -5dB**	**SNR -10dB**
27	0.6	354.52/ -331.17	353.85/ -330.50	337.79/ -314.47	280.93/ -257.66	204.67/ -181.54	100.32/ -77.46
36	0.7	355.77/ -331.16	355.77 /-331.16	341.51/ -316.94	302.69/ -278.19	206.36/ -181.95	87.92/ -63.76
99	1.3	355.75/ -325.66	355.75 /-325.66	346.20/ -316.12	306.44/ -276.41	229.35/ -199.49	147.05/ -117.53
189	1. 8	354.01/ -324.61	353.94 /-324.54	345.69/ -316.29	307.04/ -277.64	210.19/ -180.74	136.16/ -106.77

B. Output Waveforms for single target

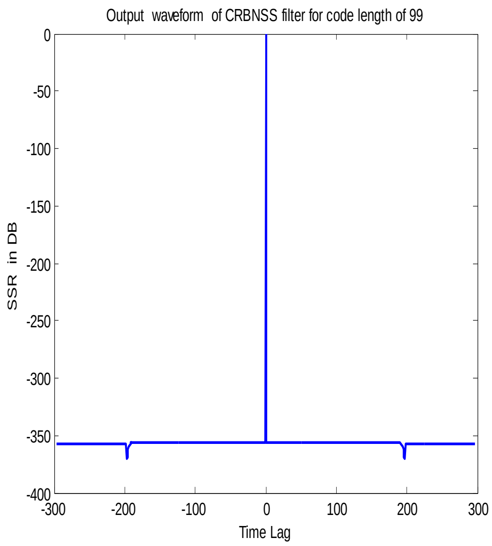
Fig.4. shows the output waveforms of the CRBNN filter for poly-semantic codes of lengths 27, 36, 99 and 189 in noise-free environment. It is observed that the side lobe levels are below -700 dB with respect to the main lobe level and the side lobe levels increase as time lag changes from zero to (N-1). The CRBNSS filter produces very high peak main lobe with negligible side lobes.


Output waveform of CRBNSS filter for code length of 27
SSR in DB
0
-50
-100
-150
-200
-250
-300
-350
-400
-80
-60
-40
-20
0
20
40
60
80
100
Time Lag

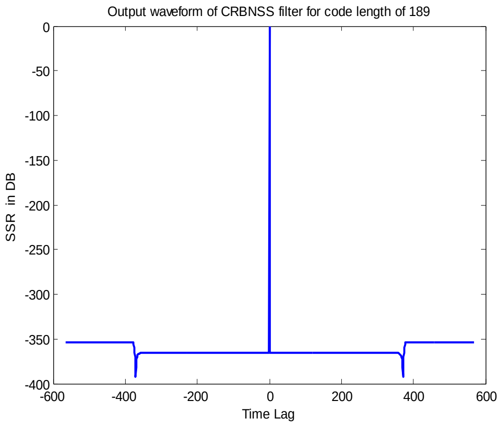
(a)


Output waveform of CRBNSS filter for code length of 36
SSR in DB
0
-50
-100
-150
-200
-250
-300
-350
-400
-150
-100
-50
0
50
100
150
Time Lag

(b)


Output waveform of CRBNSS filter for code length of 99
SSR in DB
0
-50
-100
-150
-200
-250
-300
-350
-400
-300
-200
-100
0
100
200
300
Time Lag

(c)



(d)

Fig.4. Waveforms at the output of the CRBNSS filter for poly-semantic codes of lengths (a) 27 (b) 36 (c) 99 and (d) 189 in noise-free environment.

Tables 4 shows that, the CRBNSS filter suppresses sidelobes very higher compared to these conventional filters.

Table 4 Comparison of SSR and ISL for poly-semantic code of length 36

FILTER	SSR(in dB) / ISL(in dB)			
	Without noise	SNR= 10dB	SNR= 0dB	SNR= -10dB
ACF	22.28 / -11.49	19.51 / -7.95	16.35 / -4.55	13.82 / -1.91
LS	24.00 / -15.68	19.99 / -8.85	16.52 / -4.78	13.95 / -1.95
LP	25.69 / -13.64	16.12 / -6.98	15.43 / -3.94	14.29 / -1.41
BP	42.74 / -35.07	28.12 / -21.51	19.02 / -12.32	15.31 / -7.38
NFNPC	61.24 / -55.57	44.26 / -40.77	37.01 / -33.68	32.45 / -28.51
CRBNSS	313.3 / -303.8	303.6 / -289.8	284.7 / -273.3	272.1 / -257.7

IV. CONCLUSIONS

The CRBNSS filter using radial basis neural networks have been designed and simulated for radar. Poly-semantic code is used to compare the pulse compression and side lobe suppression performance of the filter. The performance of the RBNN filters is optimized by selecting the spread values of the radial basis neural network to give maximum SSR and minimum ISL. The filter is analysed in the presence of additive noise and noise-free for a single target. It is found that at optimum spread values, CRBNSS filter yields maximum SSR values varying from 323.04 dB to 355.77 dB and minimum ISL values varying from -331.17 dB to -299.44 dB for all the codes. The RBNN filter has noise robustness up to the noise strength -10 dB SNR.

The output waveforms demonstrate that the side lobe suppression characteristics for RBNN filters are better than other conventional filters in noisy environments. Neural network approach is more efficient at low sequence lengths, but as the sequence length increases above 100, the processing speed drastically decreases. Also, neural network approach is not applicable for higher length sequences and high resolution radar signal design application

REFERENCES:

[1]. Haykin S, "Neural Networks: A Comprehensive Foundation", Macmillan College Publishing Company, Inc., New York, 1994

[2]. Khairnar D.G., Merchant S.N. and Desai U.B., "Radial basis neural network for pulse radar detection", Radar, Sonar & Navigation, IET, Vol.1, no.1, Feb 2007, pp.8-17.

[3]. Demuth H. and Beale M., "MATLAB neural network toolbox, Users guide", ver.2, The Math Works Inc., Natick, 1997.

[4]. Haygan M.T., Demuth H.B. and Beale M., "Neural Network Design", Thomson Asia Pte. Ltd, Singapore, 2002.

[5]. Duh F-B. and Juang C-F., "Radar Pulse Compression for Point Target and Distributed Target Using Neural Network", Journal of Information Science and Engineering, Vol.23, 2007, pp.183-201.

[6]. Bambang R., "Active noise cancellation using recurrent radial basis function neural networks", Asia-Pacific Conference on Circuits and Systems, Vol.2, 2002, pp.231-236.