Detection of Two Targets in Noisy Environment using Neural Network

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

***Abstract*-This paper addresses the radar signal design for the detection of two target with improved sidelobe suppression ratio using neural network. The simulation results indicate a significant in terms of noise tolerance for targets detection compared to conventional binary sequence.**

***Keywords*-Poly-semantic radar, Hamming back-track, sidelobe suppression, radial basis neural network.**

I. INTRODUCTION

In many pulse Doppler radar applications, target detection becomes very difficult when the echo consists of high Doppler shift and strong background noise due to clutter. To overcome this limitation and to enhance the detection performance [1], a novel Radial basis neural network (RBNN) filter based on radial basis function neural network is developed in this paper. This filter is also known as Correlated radial basis neural network sidelobe suppression (CRBNSS) filter. MATLAB functions are used to develop simulation algorithms for radial basis function neural network. The RBNN filter is optimized by using computer search with poly-semantic sequences as input.

II. RBNN FILTER DESIGN

RBNN filter is designed by simulation model of radial basis function neural network using MATLAB functions[2, 3] 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.

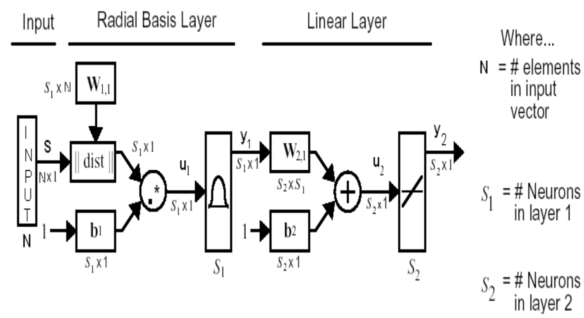


Fig.1 Architecture of Radial Basis Function Neural Network.

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

$$u_1 = \|W_{1,1} - S\| .* b_1 \tag{1}$$

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

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

$$y_2 = purelin(u_2) = f(net) \tag{4}$$

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

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

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.

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.

III. PERFORMANCE OPTIMIZATION

The performance optimization of receiver system is carried out by using poly-semantic sequences of length 9 to 189 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 filter 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 to 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 the 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 side lobes at other time lags, similar to the autocorrelation function. So, in the output waveform, the main lobe amplitude as $r(0)$ and side lobe 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 developed in this chapter is a 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 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. Table I. shows the maximum SSR and minimum ISL values for poly-semantic sequences at optimum spread values of CRBNSS filter. The CRBNSS filter yields maximum SSR and minimum ISL values at particular spread value of the network at each poly-semantic sequence 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 and ISL value becomes minimum.

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

Sequence 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

It is observed that all the poly-semantic sequences resulted in maximum SSR value above 300 dB and minimum ISL value below -300 dB at corresponding optimum spread values.

Table II. SSR values at different spreads of the network for poly- semantic sequence of 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

Table II. shows the SSR value at different spread values of the network for poly-semantic sequence of length 27. It is observed that the network yields maximum SSR value of 354.52 at spread 0.6. Thus for poly-semantic sequence of length 27, the optimum spread value is 0.6. Similarly for poly-semantic sequences of other length, the spread value can be obtained.

It is also observed that as spread value of the network increases from 0.1, the SSR value also increases. 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.

V. PERFORMANCE EVALUATION AND SIMULATION RESULTS

A. *Noise Robustness*

To evaluate the noise performance, all poly-semantic sequences which are perturbed by Gaussian noise of different SNRs of 10 dB to -10 dB are considered to be input sequence. Fig.2 and Fig.3 shows the variations of SSR and ISL at the output of CRBNSS filter for poly-semantic sequences under noise-free and noisy environments at different SNRs of 10 dB to -10 dB when the two stationary targets are separated by 2 SPDA.

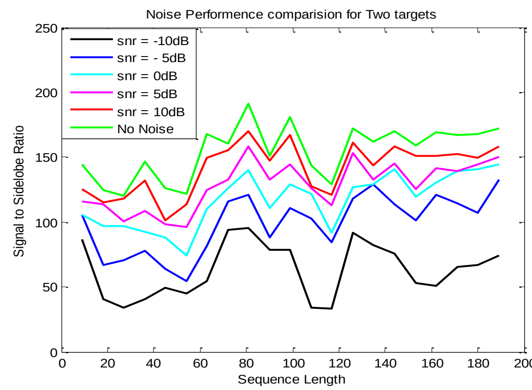


Fig. 2 Variations of SSR at the output of CRBNSS filter for poly-semantic sequences under noise-free and noisy environments at different SNRs of 10 dB to -10 dB when the two stationary targets are separated by 2 SPDA

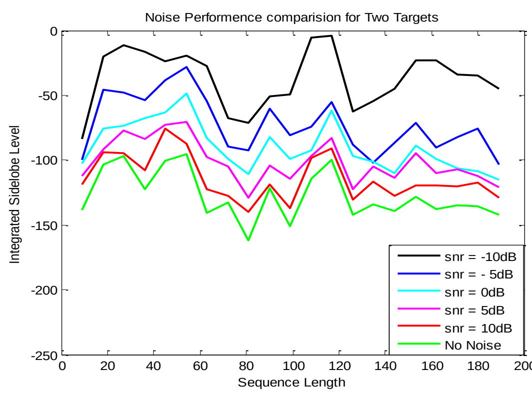
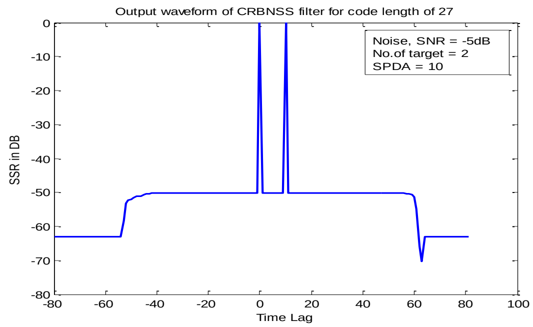


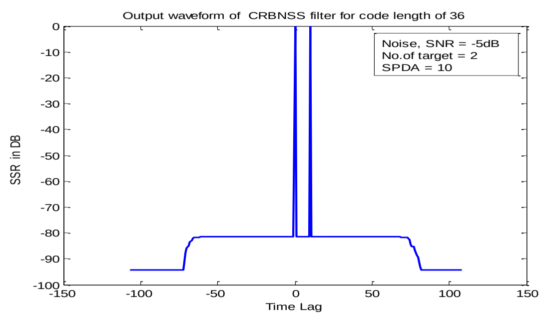
Fig. 3 Variations of ISL at the output of CRBNSS filter for poly-semantic sequences under noise-free and noisy environments at different SNRs of 10 dB to -10 dB when the two stationary targets are separated by 2 SPDA.

B. *Output Waveforms for Two Targets*

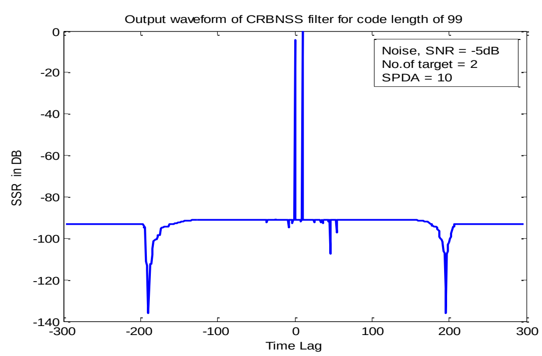
Fig.5.17 shows the output waveforms of CRBNSS filter for the poly-semantic sequences of length 27, 36, 99 and 189 for two targets separated by 10 SPDA with SNR = -5 dB.



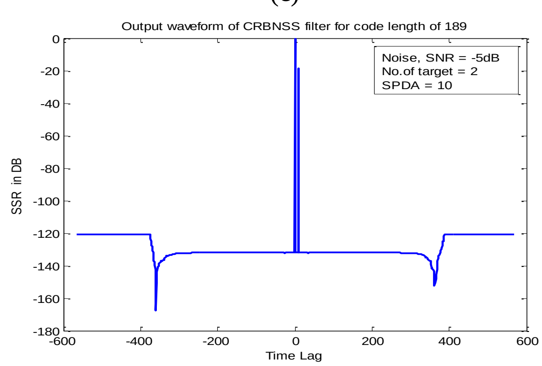
(a)



(b)



(c)



(d)

Fig.4 Waveforms at the output of the CRBNSS filter for poly-semantic sequence of length (a) 27 (b) 36 (c) 99 and (d) 189 for two targets separated by 10 SPDA with SNR = -5 dB.

It is observed that the side lobe levels are suppressed below -50 dB compared to main lobe level. The CRBNSS filter produces very high main lobe peak with negligible side lobes.

VI.CONCLUSIONS

The CRBNSS filter using radial basis neural networks have been designed and simulated for radar signal design. The performance of the RBNN filter is optimized by selecting the spread values of the radial basis neural network to give maximum SSR and minimum ISL. It is observed that the SSR values degrade from 354.52 dB to 127.31 dB, and ISL values degrade from -331.17 to -96.91.

It is observed that the side lobe levels are suppressed below -50 dB compared to main lobe level. The CRBNSS filter produces very high main lobe peak with negligible side lobes.

Analysis of the results proved that the CRBNSS filter with optimal spread values provide significant advantage of robustness in Doppler shift and additive noise on SSR and ISL for the single target and two targets. Neural network approach is more efficient at low sequence lengths, but as the sequence length increases above 100, the processing speed drastically decreases. Neural network approach provides better performance for poly-semantic sequences of length up to 189 for high resolution radar signal design applications.

REFERENCES

[1] 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.

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

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

[4] Duh F.B. and Juang C-F., “Radar Pulse Compression for Point Target and Distributed

Target Using Neural Network", Journal of ISE, Vol.23, 2007, pp.183-201.

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