

Improving Performance in Neural Network Based Pulse Compression for Binary and Polyphase Codes

Aditya V. Padaki

P.E.S. Centre for Intelligent Systems
PESIT Campus,
100 Feet Ring Road, BSK 3rd Stage,
Bangalore, India

aditya.padaki@gmail.com

Koshy George

P.E.S. Centre for Intelligent Systems;
Dept. of Telecommunication Engineering,
P.E.S. Institute of Technology
Bangalore, India

kgeorge@pes.edu

ABSTRACT

Pulse compression is important for improving range resolution, and the application of neural networks for pulse compression has been well-explored in the past. However, the practical importance of extracting rather weak echoes of targets that are either distant, or have small radar cross-section, appears to have been overlooked. Addressing this issue, neural networks with improved performance are developed in this paper for both Barker and Polyphase codes. We demonstrate that our networks perform better in such practical situations together with better noise tolerance and range resolution.

Categories and Subject Descriptors

I.5.2 [Pattern Recognition]: Design Methodology – *classifier design and evaluation, feature evaluation and selection, pattern analysis.*

I.5.3 [Pattern Recognition]: Applications – *signal processing, waveform analysis.*

J.2 [Physical Sciences and Engineering]: Electronics

General Terms

Algorithms, Performance, Design.

Keywords

Pulse compression, neural networks, binary codes, polyphase codes

1. INTRODUCTION

Pulse Compression is an important technique used in radar to improve the range resolution while maintaining a low transmitted power, and maintain a good signal-to-noise ratio [20]. All pulse compression techniques are essentially matched filtering. Such filters are optimal when the signals are embedded in additive white Gaussian noise. However, the large sidelobes of a matched

filter can increase the probability of false alarm. Several techniques for sidelobe suppression have been proposed in the literature, and many of these are based on the method of least squares [1] and minimum mean square estimation [3]. While the results obtained using these techniques are very encouraging, they also involve inversion of considerably large matrices. Such computations in real-time are rather difficult, and hardware implementations are rather demanding.

Artificial Neural Networks (ANNs) are universal approximators and offer ease in hardware implementation. Given any continuous function $f(\cdot)$ defined on a compact set, there exists an ANN represented by $F(\cdot)$ that can approximate $f(\cdot)$ to any desired accuracy [4, 6, 10]. Neural networks for pulse compression was first explored in [13], and subsequently by several researchers [2, 5, 11, 17, 18, 19]. Here, the objective is to make the ANN approximate an ideal autocorrelation sequence.

The sequences generally used in the aforementioned references to train the ANNs are time-shifted sequences of the adopted codes. During the testing phase, these sequences are added with white Gaussian noise of different noise intensities resulting in varying signal-to-noise ratio (SNR). However, in practice, the noise power across the range cells remains largely the same; the probable value depends on the specific radar system and its environment. Further, in practical applications, the objective is to detect targets at the farthest distance. Moreover, targets with small radar cross section are also required to be detected. Both imply a requirement of detecting rather weak echoes. Unfortunately, the immense importance of extracting rather weak echoes of targets out of the noise appears to be overlooked. Indeed, as illustrated by an example presented in Section 2, the performance of neural networks in the presence of weak echoes is rather unsatisfactory when they are trained ignoring this reality. Therefore, the primary focus of this paper is to design neural networks that can successfully deal with these practical situations.

The second focus of this paper is to develop neural networks (with aforementioned capabilities) for both binary phase and polyphase codes. The latter has been ignored in most of the aforementioned references. It may be pointed out that modern radar tend to employ polyphase codes which are sequences of complex numbers with constant magnitude but with variable phase. This permits easier construction of longer sequences resulting in better range resolutions. The principal drawback, however, is the need for a more complex matched filter.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

In the context of pulse compression, it has been reported that feedforward artificial neural networks (FFNNs) take longer time to train compared to radial basis function networks (RBFNs) [11, 18]. Further, such networks exhibit poor robustness [12]. The latter is primarily due to the lack of generalisation resulting from fewer number of training patterns. (For good generalisation, the size of the training set N must satisfy the condition

$$N = O\left(\frac{N_p}{\varepsilon}\right), \text{ where } N_p \text{ is the number of free parameters and}$$

ε is the permissible error [7].) In this paper, however, we concentrate on feedforward artificial neural networks since such networks are relatively easier to implement in hardware. Further, in order to deal with practical situations mentioned earlier, the size of the training set N considered in this paper is much higher, and compares well with the condition in [7].

This paper is organised as follows: In Section 2 we design a FFNN for Barker-13 code, and compare the results obtained with that available in the literature. The design of FFNN for polyphase codes is considered in Section 3.

2. BARKER-13 CODES

In this section we train a FFNN for pulse compression where the pulses have been modulated by Barker-13 code: $S = \{1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1\}$. We use a fully connected FFNN with an input layer consisting of 13 nodes (corresponding to the length of the Barker-13 code), a hidden layer with five neurons, and one output neuron. The chosen activation function for the hidden layer is $\phi(v) = a \tanh(bv)$, and a linear activation function for the output layer.

We note that the number of hidden neurons generally considered for pulse compression is three. However, this results in a higher probability of false alarm with simulations indicating a rate of about one false alarm in fifty trials. This is primarily due to the fact that the corresponding FFNN has not approximated the ideal autocorrelation sequence to the desired accuracy. Since it is well-known that the approximating capacity of FFNNs depend on the number of hidden neurons [16], it is clear that the number of hidden neurons is to be increased. Indeed, experience indicate that a choice of five hidden neurons considerably improves the situation, and the improved overall performance worth the slight increase in complexity.

Moreover, the training sequences generally used to train the neural networks largely overlook the need to detect targets with weak echoes. For instance, consider a FFNN with an input layer of 13 nodes, hidden layer of 3 neurons, and one output neuron trained for a normalised Barker-13 code. The performance of this trained network for a received radar signal power of -20 dB with an SNR of 40 dB is shown in Fig. 1. Evidently, on an average, the target cannot be detected. On the contrary, the performance of a matched filter is quite satisfactory.

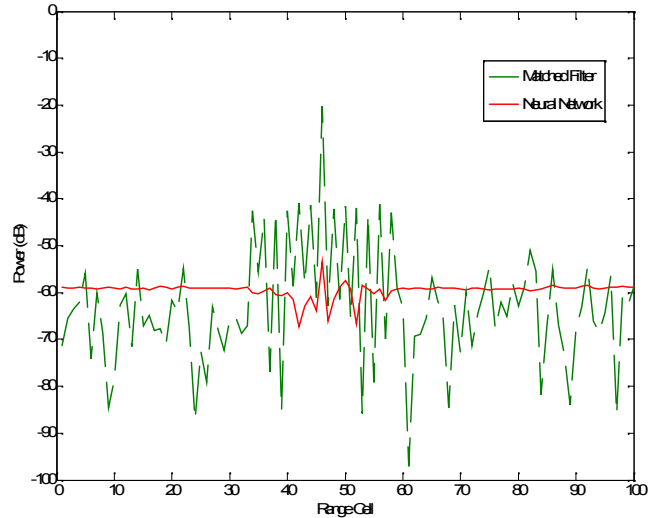


Figure 1. Performance of a neural network not trained for weak echoes.

In order to account for rather weak echoes from possibly distant targets or targets with low radar cross section, the network is trained for various target-return powers. The chosen power levels are 0 dB, -6 dB, -12 dB, -18 dB, -24 dB, and -30 dB

corresponding to $\left(\frac{1}{2}\right)^n S$, $n = 0, 1, \dots, 5$. (For simplicity, if a

target-return has a power level of x dB, that target is referred to in the sequel as a ' x dB target'.) For each power level, time shifted codes are presented to the neural network resulting in 26 input patterns. Thus, $N = 156$. The desired signal is an ideal autocorrelation function: peak when the autocorrelation lag is zero, and zero elsewhere. We use the back propagation algorithm [7] to adjust the free parameters in the network.

2.1 Simulation Results and Discussions

The performance of the trained neural network is discussed in this section. For brevity, the target is assumed to be in the 46th range cell; similar results are obtained when a target is present in any other range cell. The amplitudes and powers of the outputs of a matched filter and the neural network are compared in Figures 2 and 3 for a 0 dB target with an SNR of 50 dB. The matched filter has six equal sidelobes at -22.3 dB on either side of the peak; i.e., the signal-to-sidelobe ratio (SSR) is 22.3 dB. On the contrary, by virtue of the chosen training sequences, the SSR obtained when using the neural network is 50.9 dB which is higher compared to that obtained using FFNN in [5, 13], RBF in [11], or other techniques such as least squares inverse filter [1], but lesser than if a FFNN is used with Bayesian regularisation [12]. This scenario is further tested with SNRs 10 dB, 20 dB, and 40 dB, and the resulting SSR (averaged over several simulation experiments) summarised in Table 1. Clearly, there is a decrease in the SSR with a decrease in SNR. However, the SSR obtained is still better than that obtained for a conventional matched filter.

Table 1. Barker Code: SSR for targets with different power returns.

Target Power	SNR (dB)			
	10	20	40	50
0 dB	37.28	41.92	48.45	50.15
-10 dB	24.31	32.01	40.05	46.91
-20 dB	22.31	29.20	42.12	45.12
-40 dB	9.48	15.02	44.12	47.42

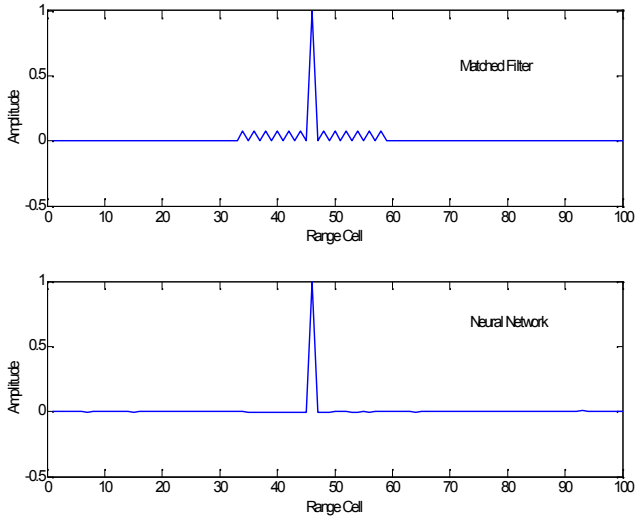


Figure 2. Barker Code: Comparison of amplitude plots for a 0 dB target.

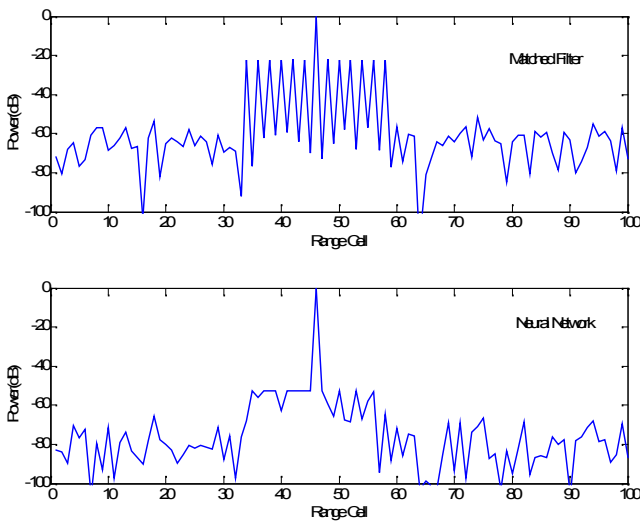


Figure 3. Barker Code: Comparison of power plots for a 0 dB target.

A comparison of the amplitudes and powers of the outputs of a matched filter and the neural network for a -10 dB target at 40 dB SNR are respectively shown in Figures 4 and 5. Similarly, a comparison of outputs of matched filter and the neural network for a -20 dB target and a -40 dB target, for SNRs 30 dB and 10 dB, are respectively depicted in Figures 6 and 7. The SSR for these targets at different SNRs (averaged over several simulation experiments) are also summarised in Table 1.

It is evident from the results depicted in these figures and summarised in the table that the performance of an FFNN trained for different target-return powers is quite satisfactory. A comparison of the output powers for all cases is shown in Fig. 8. For clarity, the same comparison for range cells from 33 to 59 are shown in Fig. 9. Clearly, beyond the range of cells centred around the one that contains the target, the sidelobe levels are lower than that of the noise floor. Further, the sidelobe levels are reasonably low enabling very weak target returns to be detected despite very low SNRs.

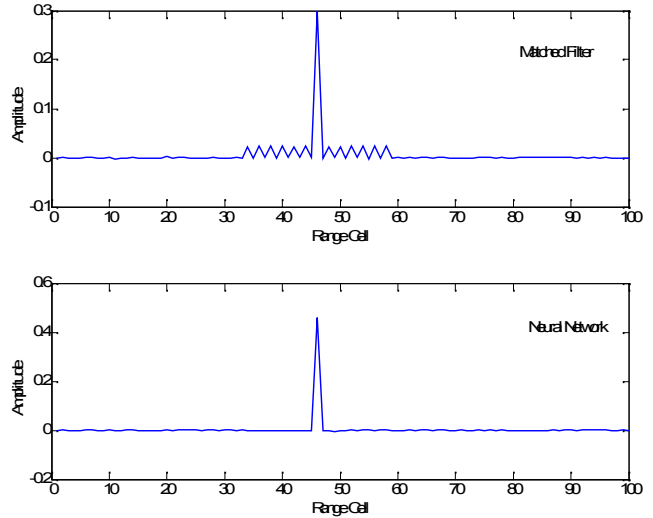


Figure 4. Barker Code: Comparison of amplitude plots for a -10 dB target.

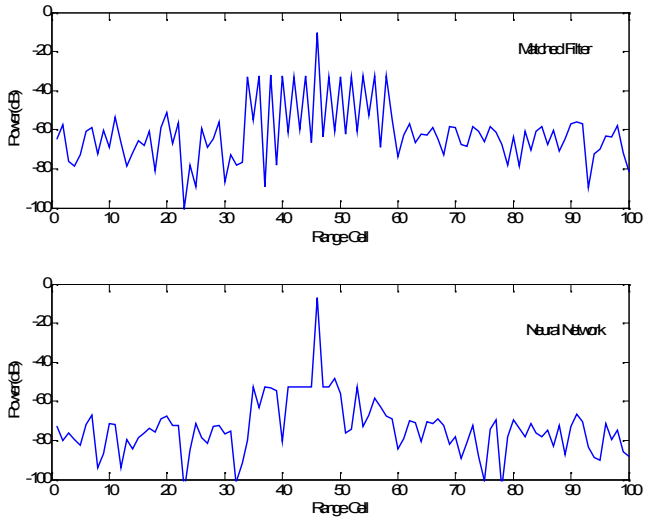


Figure 5. Barker Code: Comparison of power plots for a -10 dB target.

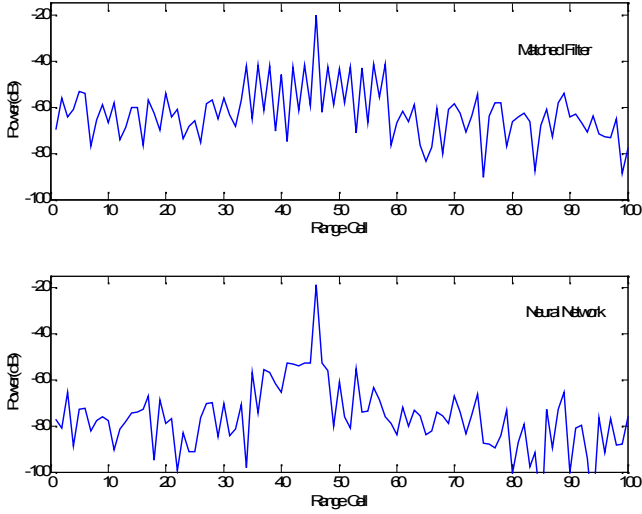


Figure 6. Barker Code: Comparison of power plots for a -20 dB target.

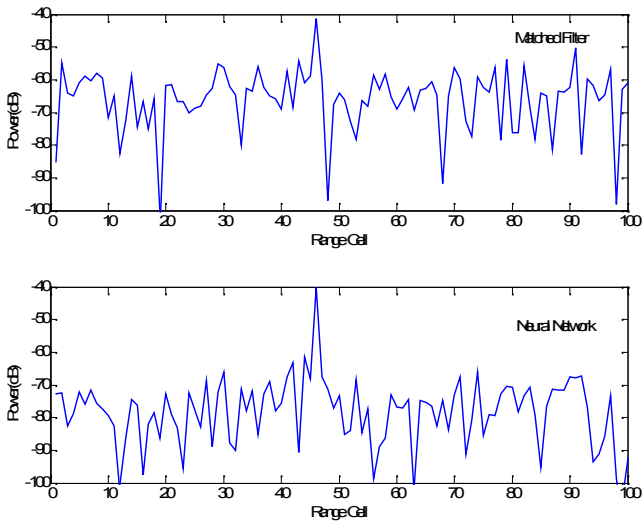


Figure 7. Barker Code: Comparison of power plots for a -40 dB target.

Except for situations with high SNRs, it is not possible to do this with matched filtering for weak targets. As mentioned earlier, to the best knowledge of the authors, there are no similar reported results in the literature for neural-network based pulse compression. Further, it is evident from Fig. 1 and the discussions earlier in this section, when the neural network is not trained with weaker echoes, as is generally reported, the performance in the presence of such weak echoes is rather unsatisfactory.

3. POLYPHASE CODES P3

In this section we design a FFNN for the Lewis-Kretschmer P3 polyphase code [15] defined by

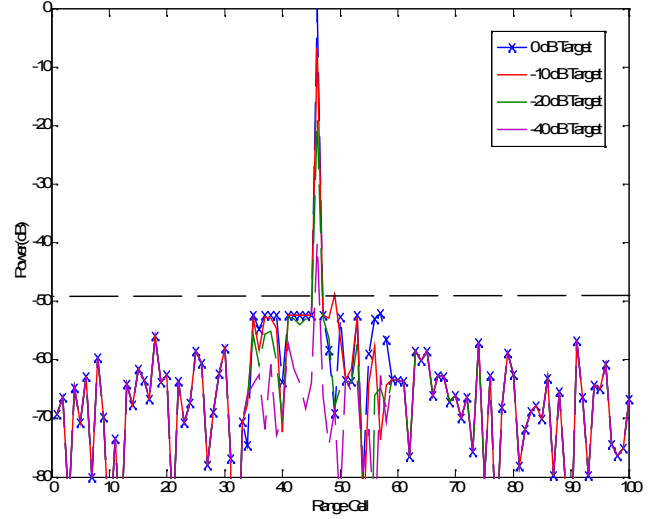


Figure 8. Barker Code: Comparison of power plots for targets with different power returns.

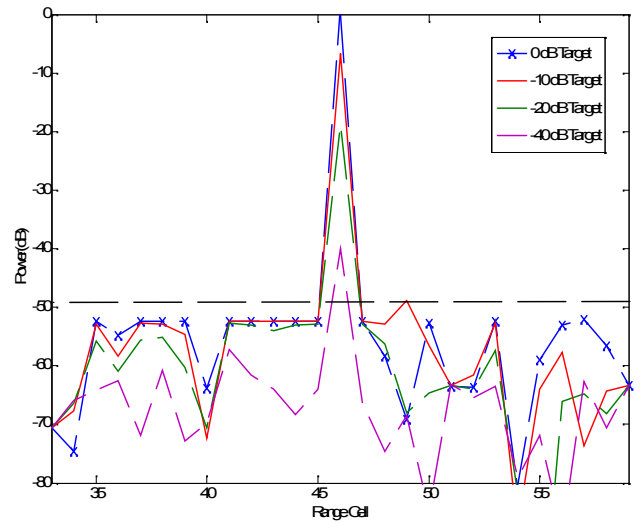


Figure 9. Barker Code: Comparison of power plots for targets with different power returns.

$$s(n) = \exp\left(\frac{j\pi n^2}{M}\right), \quad 0 \leq n \leq M-1$$

We chose $M = 30$. Since the polyphase component has an in-phase and a quadrature-phase component, the resulting network should have the capacity to deal with complex numbers, and are referred to as *complex neural networks*.

Indeed there are two ways to obtain complex neural networks [8]: In the first approach, the real and imaginary parts are treated separately, and two neural networks are trained, one for the real part, and the other for the imaginary part. Since all the free

parameters of both neural networks are real, the standard back propagation algorithm (BPA) can be used to train the networks. In the second approach, the free parameters are taken as complex numbers, and a modified algorithm known as the complex BPA [14] is used to train the network. It has been shown that the two approaches are equivalent [8, 9]. In this paper we use the first approach for the following reasons: The code pattern used has distinct in-phase and quadrature phase components (i.e., the real and imaginary parts). Hardware implementation is comparatively simpler as complex arithmetic operations are not required.

Thus, we choose two fully-connected feedforward neural networks, one for the in-phase and the other for quadrature component. Each network consists of three layers, an input layer with 30 source nodes, a hidden layer with nine neurons, and one output neuron. The number of hidden neurons is considerably higher compared to the neural network for Barker codes due to the complexity of the polyphase codes. The chosen activation function for the hidden layer is $\phi(v) = a \tanh(bv)$, and a linear activation function for the output layer. Similar to the case of neural network developed for Barker-13 code, the number of hidden neurons is chosen to reduce the probability of false alarm.

The two networks for P3 codes are trained using sequences chosen in a manner similar to that of Barker-13 codes. Accordingly the network is trained for target-return powers of 0 dB, -3 dB, -6dB, -12 dB, -18 dB, -24 dB, and -30 dB. (An additional training pattern with power -3 dB is introduced to improve the sidelobe suppression.) Time shifted codes are presented to the network as training patterns; the training set size $N = 420$. In contrast to the training sequences of Barker codes, experience indicate that the neural networks are more robust when the training sequences are corrupted by noise for the following reason: While the general structure of a Barker code remains largely unaffected with the addition of noise, there is a considerable difference in the polycodes when noise is added. The desired signal is again an ideal autocorrelation function: peak when the autocorrelation lag is zero, and zero elsewhere.

3.1 Simulation Results and Discussions

The performance of the neural network is discussed here. The target is assumed to be in the 41st range cell; similar results are obtained when a target is present in any other range cell. The amplitudes and powers of the outputs of a matched filter and the neural network are compared in Figures 10 and 11 for a 0 dB target at 50 dB SNR. The matched filter has peak sidelobe at -20.85 dB to either side of the peak; i.e., the signal-to-sidelobe ratio (SSR) is 20.85 dB. On the contrary, by virtue of the chosen training sequences, the SSR obtained when using the neural network is 45.18 dB. This scenario is further tested with SNRs 10 dB, 20 dB and 40 dB. It is repeated for all power returns and the resulting SSR (averaged over several experiments) is summarised in Table 2. For brevity, only the comparison of power plots for a -40 dB target is shown in Fig 12.

Table 2. SSR for targets with different power returns for P3 codes.

Target Power	SNR (dB)			
	10	20	40	50
0 dB	15.12	23.98	36.44	43.51
- 10 dB	14.28	22.74	30.52	34.61
- 20 dB	10.24	17.48	27.73	30.95
- 40 dB	8.52	9.25	9.74	9.91

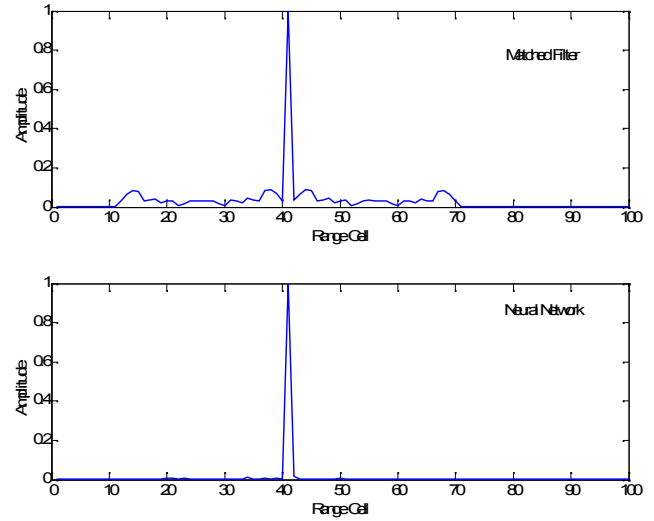


Figure 10. P3 Code: Comparison of amplitude plots for a 0 dB target.

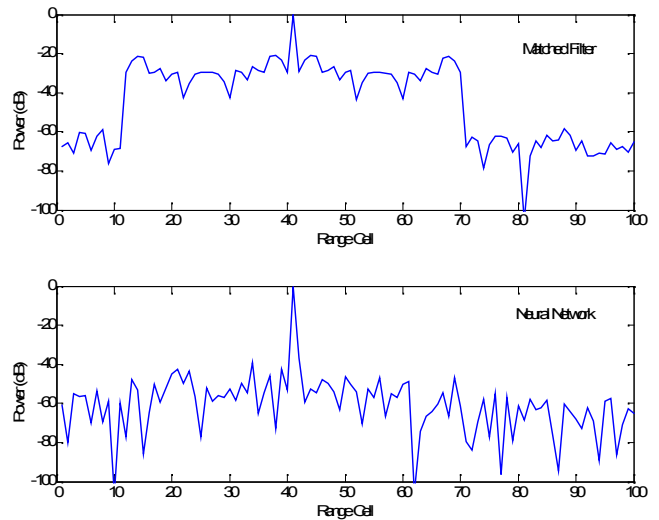


Figure 11. P3 Code: Comparison of power plots for a 0 dB target.

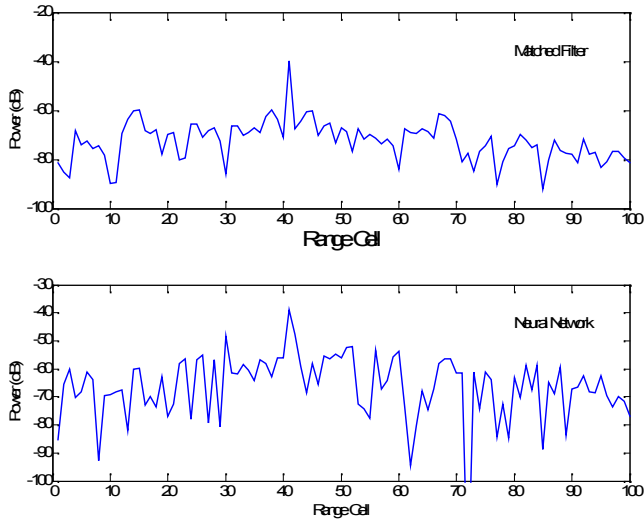


Figure 12. P3 Code: Comparison of power plots for a -40 dB target.

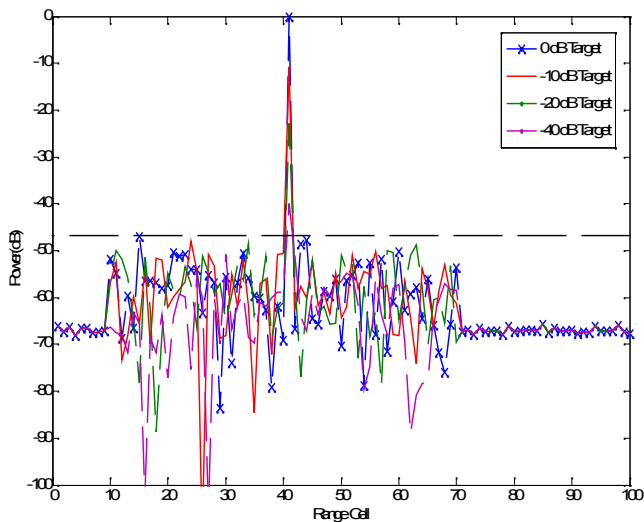


Figure 13. P3 Code: Comparison of power plots for targets with different power returns.

The aforementioned results indicate that the performance of a neural network for P3 code is quite satisfactory. The least SSR obtained is 8.52 dB for a target of -40 dB at 10 dB SNR. However, for lower SNRs it is quite possible that this target may not be detected. Further, from Fig. 12 it can be observed that the sidelobe suppression is comparable to that of a matched filter. Nonetheless, it is to be noted that in contrast to a matched filter, the sidelobe levels remain practically the same at different target power levels at the same noise level of -50 dB, as illustrated in Figures 13 and 14. This implies that targets with rather weak returns can still be detected.

4. NOISE TOLERANCE

From the results presented in Sections 2 and 3, it is abundantly clear that the neural networks presented in this paper are quite robust toward noise. The SSR is typically lower for distant targets

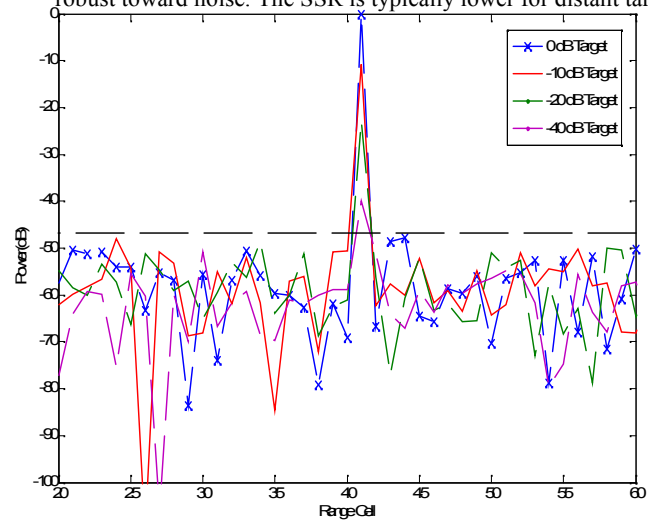


Figure 14. P3 Code: Comparison of power plots for targets with different power returns.

or targets with smaller radar cross-section. This is primarily because of the signal power itself being low and the neural network being forced to give a constant output in the range cells without the targets for all power returns. Thus, to maintain the sidelobe levels constant for all the power returns, the networks give a relatively low SSR to lower power targets. Nevertheless, the SSR provided is more than satisfactory and sufficient, since the primary objective of target detection is always met.

Previously, RBFNs have been considered for the development of robust noise tolerance networks [11, 18, 19]. In contrast, our FFNNs are able to provide a substantial noise tolerance, enabling the target detection at not only various SNR, but also for various powers of radar return.

5. RANGE RESOLUTION

Range resolution is the ability of the radar to distinguish between two targets that are close to each other in range. It has been reported that neural networks for Barker code can distinguish two targets in consecutive range cells; however, due to the structure of the Barker code, a second target placed 5 range cells apart is not detected when the target power ratio is more than 15 [11, 13]. However, using the neural network developed in this paper, this problem does not arise even with target power ratio of 50. This is illustrated in Fig. 15. The range resolution capability of P3 code is illustrated in Fig. 16. Here two targets (of power ratio 10) are placed 2 range cells apart.

6. CONCLUSION

In this paper, we developed a robust neural-network based pulse compression with the capability of detecting targets of varied power returns. We use the feedforward neural network (FFNN) structure for ease of hardware implementation. FFNNs are

developed for both Barker-13 and P3 codes. Extensive simulations indicate the efficacy of our networks to detect rather weak echoes even at low SNRs, provide satisfactory noise tolerance and range resolution capability.

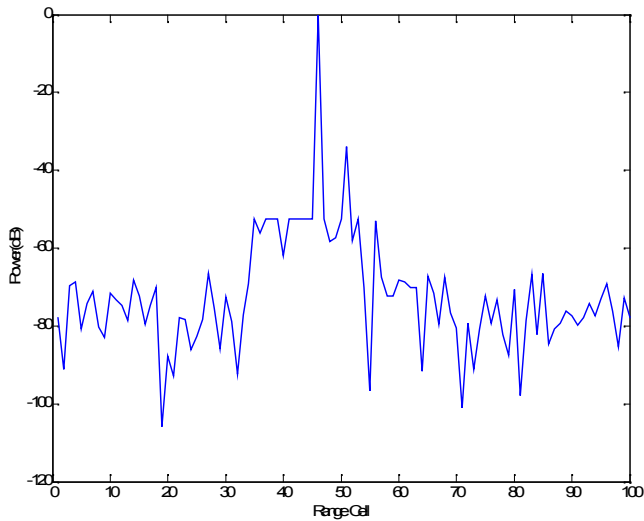


Figure 15. Range Resolution for Barker-13 codes. Second Target is easily detected.

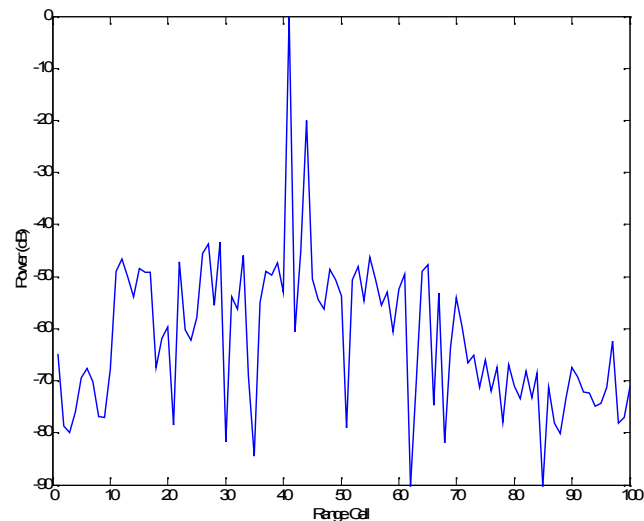


Figure 16. Range Resolution for P3 codes. Second Target is easily detected.

7. REFERENCES

[1] Ackroyd, M. H., and Ghani, F. 1973. Optimum mismatched filters or sidelobe suppression. *IEEE Transactions on Aerospace and Electronic Systems*, 9 (March 1973), 214–218.

[2] Baghel, V. 2009. Multiobjective optimization – new formulation and application to radar signal processing.

Master's thesis, Department of Electronics and Communication, National Institute of Technology, Rourkela, India

[3] Blunt, S. D., and Gerlach, K. 2003. A novel pulse compression scheme based on minimum mean square error reiteration. In *Proceedings of the IEEE International Radar Conference (Adelaide, Australia, Sept. 2003)*, 349–353.

[4] Cybenko, G. 1989. Approximation by superpositions of sigmoidal function. *Mathematics of Control, Signal, and Systems*, 2, 303–314.

[5] Duh, F. B., and Juang, C. F. 2005. Radar pulse compression for point target and distributed target using neural network. *Journal of Information Science and Engineering*, 13 (June 2005), 183–201.

[6] Funahashi, K. 1989. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2, 183–192.

[7] Haykin, S. 1999. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, NJ, USA, 2nd edition.

[8] Haykin, S. 2001. *Adaptive Filter Theory*. Prentice Hall, NJ, USA, 4th edition.

[9] Haykin, S., and Ukraine, A. 1993. Neural networks for adaptive signal processing. In *Adaptive System Identification and Signal Processing Algorithms*, Kalouptisidis, N., and Theodoridis, S., Ed. Prentice Hall, Englewood Cliffs, NJ, USA, 512–553.

[10] Hornik, K., Stinchcombe, M., and White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2, 359–366.

[11] Khairnar, D. G., Merchant, S. N., and Desai, U. B. 2007. Improving performance in pulse radar detection using radial basis function network. *IET Radar Sonar and Navigation*, 1, 1 (Feb. 2007), 8–17.

[12] Kumar, P., Merchant, S. N., and Desai, U. B. 2004. Improving performance in pulse radar detection using Bayesian regularization for neural network training. *Digital Signal Processing*, 14 (July 2004), 438–448.

[13] Kwan, K. H., and Lee, C. K. 1993. A neural network approach to pulse radar detection. *IEEE Transactions on Aerospace and Electronic Systems*, 29, 1 (Jan. 1993), 9–21.

[14] Leung H., and Haykin, S. 1991. The complex back propagation algorithm. *IEEE Transactions on Signal Processing*, 39, 9 (Sept. 1991), 2101–2104.

[15] Lewis, B. L., and Kretschmer, F. F. 1982. Linear frequency modulation derived polyphase pulse compression codes. *IEEE Transactions on Aerospace and Electronic Systems*, 18, 5 (Sept. 1982), 637–641.

[16] Nguyen, D. and Widrow, B. 1990. Improving learning speed of 2-layer neural network by choosing initial values of adaptive weights. In *Proceedings International Joint Conference on Neural Networks (Washington D. C., USA, June 1990)*, 21–26.

[17] Rao K. D., and Sridhar, G. 1995. Improving performance in pulse radar detection using neural networks. *IEEE*

Transactions on Aerospace and Electronic Systems, 31, 3
(July 1995), 1193–1198.

- [18] Reddy Y. M. 2007. Performance Optimization of High Resolution Radar Signatures using Radial Basis Function Neural Networks. PhD thesis, University College of Engineering, Osmania University, Hyderabad, India.
- [19] Reddy, Y. M., Pasha, I. A., and Vathsal, S. 2006. Design of radial basis neural network filter for pulse compression and

sidelobe suppression in a high resolution radar. In Proceedings International Radar Symposium (Krakaw, Poland, May 2006), 1–46.

- [20] Skolnik, M. I. 1980. Introduction to Radar Systems. McGraw Hill, NY, USA.