Doppler frequency estimation with wavelets and neural networks

Steven E. Noel, Harold H. Szu, and Yogesh J. Gohel

Naval Surface Warfare Center Dahlgren Division, Dahlgren, VA 22448

ABSTRACT

In this paper we apply the continuous wavelet transform, along with multilayer feedforward neural networks, to the estimation of time-dependent radar doppler frequency. The wavelet transform employs the real-valued Morlet wavelet, which is well matched to the doppler signals of interest. The neural networks are trained with the Levenberg-Marquardt rule, which is much faster than purely gradient-descent learning algorithms such as backpropagation. We also apply Donoho's wavelet denoising with the novel super-Haar wavelet to improve performance for noisy signals. The techniques are applied to the problem of radar proximity fuzing.

Keywords: Continuous wavelet transform, radar signal processing, feedforward neural networks, wavelet denoising, Morlet wavelet, super-Haar wavelet.

1. INTRODUCTION

Radar systems detect objects by emitting electromagnetic energy, then processing the signal from the reflected radiation. A fundamental type of information provided by radar is the change in frequency of the echo signal relative to the emitted signal. The frequency shift is proportional to the relative velocity between the object and the radar system. This is the well-known doppler effect. In our work, we are particularly interested in the estimation of doppler frequency for detecting the proximity of targets for fuzing. Such proximity fuzes were one of the first uses of radar,[1] and they continue to have widespread military application today.

The doppler frequency varies over time as the relative velocity between the radar and target changes. This can be analyzed through time-frequency or time-scale representations, for example wavelet transforms. In this paper we apply the continuous wavelet transform to the estimation of time-dependent doppler frequency. The continuous wavelet transform, unlike the discrete wavelet transform, is not restricted to dyadic time scales, and therefore offers more flexibility in the analysis. We perform the transform with the real-valued Morlet wavelet,[2] which is well matched to the doppler signals of interest. This wavelet transform is similar to a time-dependent Fourier transform with a Gabor window,[3] except the window bandwidth changes in proportion to the center frequency of the wavelet rather than remaining fixed.

In radar proximity fuzing, the doppler signals for certain types of targets can be very noisy. To improve frequency estimation performance for such noisy signals, we first apply Donoho's wavelet denoising algorithm.[4] The algorithm computes the discrete wavelet transform of the noisy signal, adaptively shrinks the wavelet coefficients towards zero, then computes the inverse transform to provide a non-parametric estimation of the uncorrupted signal. We compute the discrete transforms with the super-Haar[5] wavelet, which is the superposition of Haar wavelets. The super-Haar wavelet provides good denoising performance for the doppler signals of interest.

The continuous wavelet transform of doppler signals provides information about their time-varying frequencies. We extract this information with feedforward multilayer neural networks, known as multilayer perceptrons.[6] After computing the wavelet transform of the denoised doppler signal, we sample the transform coefficients to provide signal features for neural network processing. The networks are trained with the Levenberg-Marquardt rule[7] to provide the

Further author information –
S.E.N. (correspondence): Email: snoel@nswc.navy.mil; Telephone: 540-653-5488; Fax: 540-653-5503
H.H.S.: Email: hszu@nswc.navy.mil; Telephone: 540-653-2655; Fax: 540-653-2687

Y.J.G.: Email:ygohel@nswc.navy.mil; Telephone: 540-653-5488; Fax: 540-653-5503
doppler frequency for a given time interval. This rule is a powerful generalization of gradient descent which employs an approximation of Newton's method. It is much faster than standard gradient descent algorithms such as backpropagation.

In the next section we briefly discuss continuous-wave radar systems, and characterize their doppler signals for the proximity fuzing problem. In Section 3 we describe the wavelet transform of these signals using the real-valued Morlet wavelet. Section 3 also shows how the wavelet denoising algorithm, with the super-Haar wavelet, can provide an estimate of a doppler signal in the presence of noise. In Section 4, we estimate the time-varying frequencies of doppler signals through the application of wavelet denoising, the wavelet transform, and multilayer perceptrons.

2. PROXIMITY FUZING WITH CONTINUOUS-WAVE RADAR

A radar system can measure the position and velocity of an object in space by transmitting electromagnetic energy and receiving the reflected radiation. One type of radar system is pulse radar, which transmits short bursts of electromagnetic energy. Measurement of the time between transmission and reception provides the distance to the target. However, a radar system can operate with a continuous rather than a pulsed transmitted signal, provided the relatively weak reflected signal can be separated from the transmitted one. One way of separating the transmitted and reflected signals is through the change in frequency of the reflected signal caused by the doppler effect.

It is well known in optics and acoustics that if either the source or observer of an oscillating wave is in motion, the oscillation frequency appears to shift. This shift is the doppler effect. In the case of proximity fuzing the source and observer are both located in the fuze, and the doppler effect arises from the relative motion between the fuze and its target. Assume the distance between the fuze and target is R. The total number of radiation wavelengths λ over the transmitted and reflected paths is then $2R/\lambda$. Since one wavelength λ corresponds to an angular phase of 2π, the total phase ϕ is $4\pi R/\lambda$. The rate of change in ϕ with time is the angular doppler frequency ω_d, which is then

$$\omega_d = 2\pi f_d = \frac{d\phi}{dt} = \frac{4\pi}{\lambda}\frac{dR}{dt} = \frac{4\pi v_r}{\lambda}, \tag{1}$$

where f_d is the doppler frequency shift and v_r is the relative velocity of the target with respect to the fuze. The doppler frequency shift f_d then becomes

$$f_d = \frac{2v_r}{\lambda} = \frac{2v_r f_0}{c}, \tag{2}$$

where f_0 is the transmitted frequency and c is the velocity of radiation propagation.

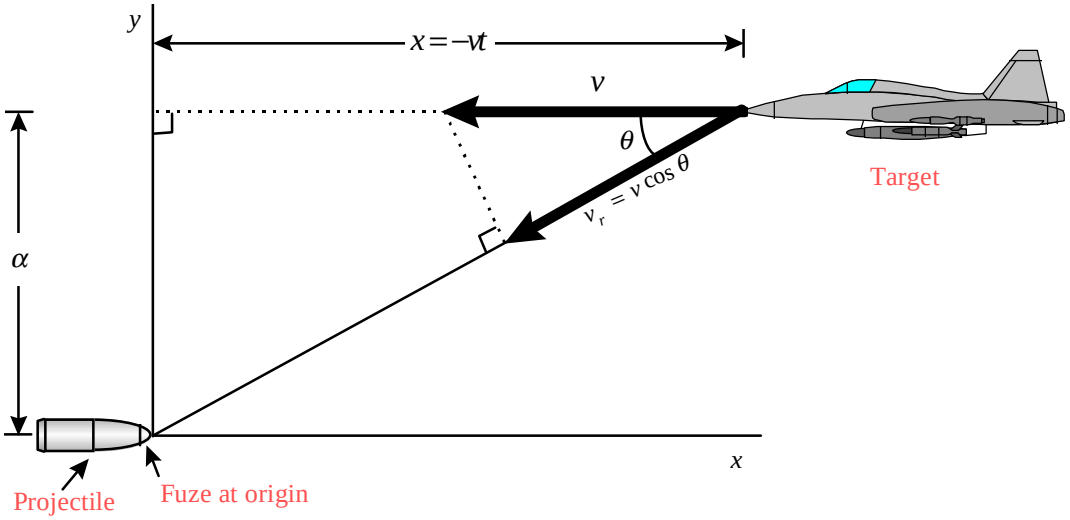

Figure 1. Relative velocity between fuze and target

Figure 1 shows a fuze and target approaching at a constant velocity $\vec{v}$, with the origin fixed at the fuze. The relative velocity v_r can be written as

$$v_r = |\vec{v}| \cos\theta = v \cos\theta , \tag{3}$$

where θ is the angle between $\vec{v}$ and the line joining the fuze and target. The angle θ can be written as

$$\theta = \cot^{-1}\left(\frac{-vt}{\alpha}\right), \tag{4}$$

for time t, with $t = 0$ at the time of closest approach between fuze and target. Here α is the closest-approach distance, that is the distance at $t = 0$, where $\theta = \pi / 2$. The optimal value of θ for fuzing is known to be $\theta_{\text{opt}} = \tan^{-1}\left(v_{\text{frag}} / v\right)$, where v_{frag} is the velocity of the warhead fragments. The fuzing problem is then to estimate the doppler frequency over time, and to fuze when the doppler frequency reaches its optimum value.

3. CONTINUOUS WAVELET TRANSFORM OF DOPPLER SIGNALS

The Fourier transform is the cornerstone of signal processing. However, since it lacks time localization, it is less suited to the processing of doppler signals whose frequencies change over time. The time-dependent Fourier transform localizes time by doing the transform over a window, which shifts in time. However, the width of the window is fixed over the entire transform, which causes problems in the high-frequency limit.[8] In contrast, a wavelet transform has a window whose bandwidth varies in proportion to the center frequency of the wavelet. It performs time-scale processing rather than time-frequency processing. The wavelet transform provides the local scale of the signal over time, which for doppler signals is the local period or inverse of frequency.

The continuous wavelet transform[2] $F_w(a,b)$ of a signal $f(t)$ is given by

$$F_w(a,b) = a^{-1/2} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt, \qquad a > 0 , \tag{5}$$

where a and b are scale and shift parameters, respectively. A necessary and sufficient condition for Eq. (5) to be invertible is that $\psi(t)$ satisfies the wavelet admissibility condition $\int_{-\infty}^{\infty} |\Psi(\omega)|^2 |\omega|^{-1} d\omega < \infty$, where $\Psi(\omega)$ is the Fourier transform of $\psi(t)$. If $\psi(t)$ has reasonable smoothness and decay at infinity, which is usually the case, the admissibility condition can be written as $\int_{-\infty}^{\infty} \psi(t) dt = 0$. Under certain conditions, it is possible to reconstruct $f(t)$ from samples of $F_w(a,b)$ taken on a hyperbolic lattice. The collection of wavelet functions $\psi\left(\frac{t-b}{a}\right)$ over this lattice is said to constitute a frame. A frame, in contrast to a basis, is an overcomplete set. Such a redundant representation allows more flexibility in the choice of inputs to neural networks. In particular, we are not constrained to the dyadic scales characteristic of the discrete wavelet transform.

We choose for $\psi(t)$ the real part of the Morlet wavelet,[9] which is

$$\psi(t) = \text{Re}\left(e^{-i\omega_0 t} e^{-t^2/2}\right) = \cos\left(\omega_0 t\right) e^{-t^2/2} , \tag{6}$$

with $\omega_0 = \pi\sqrt{2 / \ln 2} = 5.336$. The real Morlet wavelet is a Gaussian-modulated sinusoid, which is well suited to processing sinusoidal doppler signals. The wavelet transform with the real Morlet is similar to the time-dependent cosine Fourier transform with a Gabor[8] (Gaussian-shaped) window, given by

$$F(\omega, b) = \int_{-\infty}^{\infty} f(t) \cos\left(\omega t\right) e^{-(t-b)^2/2} dt . \tag{7}$$

For comparison, we can write Eq. (5) as

$$F_w(\omega', b) = \int_{-\infty}^{\infty} f(t) \cos\left[\omega'(t-b)\right] e^{-\left(\frac{t-b}{\omega_0/\omega'}\right)^2/2} dt , \tag{8}$$

where $\omega' = \omega_0 / a$. For the time-dependent Fourier transform in Eq. (7), the width of the window $\exp\left[-(t-b)^2/2\right]$ remains fixed. However, for the wavelet transform in Eq. (8) the width of the window $\exp\left[-\left((t-b)/a\right)^2/2\right]$ varies inversely with the frequency ω'. Thus the frequency bandwidth of the wavelet window varies in proportion to ω'. Also, the cosine term $\cos\left[\omega'(t-b)\right]$ for the wavelet transform shifts in time along with the window, through the shift parameter b. In contrast, for the time-dependent Fourier transform only the window shifts in time, and the cosine term remains fixed.

To improve performance for noisy doppler signals, we apply Donoho's wavelet denoising algorithm[4] before computing the continuous wavelet transform. The algorithm first does the discrete wavelet transform with Mallat's fast pyramid algorithm.[10] The pyramid algorithm computes the transform for J dyadic levels of scale, resulting in vectors of detail and smooth wavelet coefficients $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{J-1}, \mathbf{d}_J, \mathbf{s}_J$. The algorithm then shrinks the detail coefficients for scales $j \le J-1$ to obtain $\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2, \ldots, \tilde{\mathbf{d}}_{J-1}$. Here $\tilde{\mathbf{d}}_j = \delta_{\lambda_j \sigma_j}(\mathbf{d}_j)$, where $\delta_{\lambda\sigma}(x)$ is a nonlinear threshold function given by $\delta_{\lambda\sigma}(x) = 0$ if $|x| \le \lambda\sigma$, $\delta_{\lambda\sigma}(x) = \text{sign}(x)(|x| - \lambda\sigma)$ if $|x| > \lambda\sigma$. The function $\delta_{\lambda\sigma}(x)$ is parameterized by a threshold λ and an estimate of the standard deviation of the noise σ. We use a universal threshold[4] $\lambda_j = \sqrt{2\log(N)}$, where N is the number of data samples. For σ we use the median absolute deviation, which is a robust estimation of standard deviation. Finally, the denoising algorithm computes the inverse discrete wavelet transform using the new coefficients $\tilde{\mathbf{d}}_1, \ldots, \tilde{\mathbf{d}}_{J-1}, \mathbf{d}_J, \mathbf{s}_J$. This results in a non-parametric estimate of the signal without the noise.

For the discrete wavelet transform in the denoising algorithm, we apply a super-Haar wavelet,[5] which is a linear superposition of shifted Haar wavelets. The super-Haar scaling function $\phi(t)$ is given by

$$\phi(t) = \sum_k s_k \phi_H(t-k), \tag{9}$$

where s_k are integer coefficients and $\phi_H(t)$ is the Haar scaling function,[11] given by $\phi_H(t) = 1, t \in [0,1)$ and $\phi_H(t) = 0, t \notin [0,1)$. We apply the super-Haar with $s_k = [1,2,2,1]$.

Figure 2 shows pure, noisy, and denoised versions of a simulated doppler signal. The closest-approach distance α is such that the change in frequency is nearly linear over time. We assume that the sinusoid amplitude is constant over time, which is appropriate over the short time intervals applicable to fuzing. Figures 3 and 4 show the continuous wavelet transforms of the pure, noisy, and denoised doppler signals. The wavelet transforms show the increase in local signal scale over time. In this case the increasing signal scale is the increasing period of the modulated sinusoid. The time-scale structure is somewhat visually apparent in the transform of the noisy signal. However, when samples of the noisy transform are used as neural network inputs for frequency estimation, the high-frequency fluctuations result in poor performance. These fluctuations are largely removed by the wavelet denoising, resulting in much improved performance.

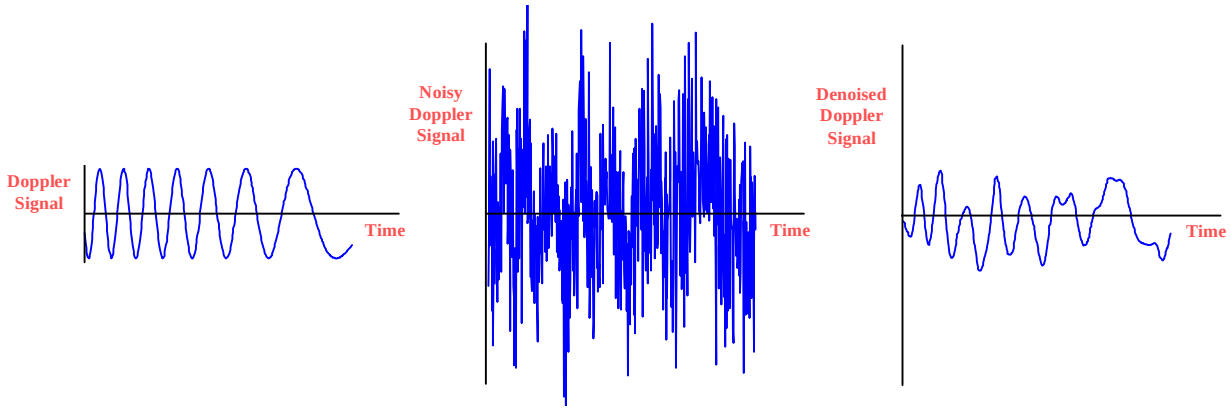

Figure 2. Pure, noisy, and denoised doppler signals

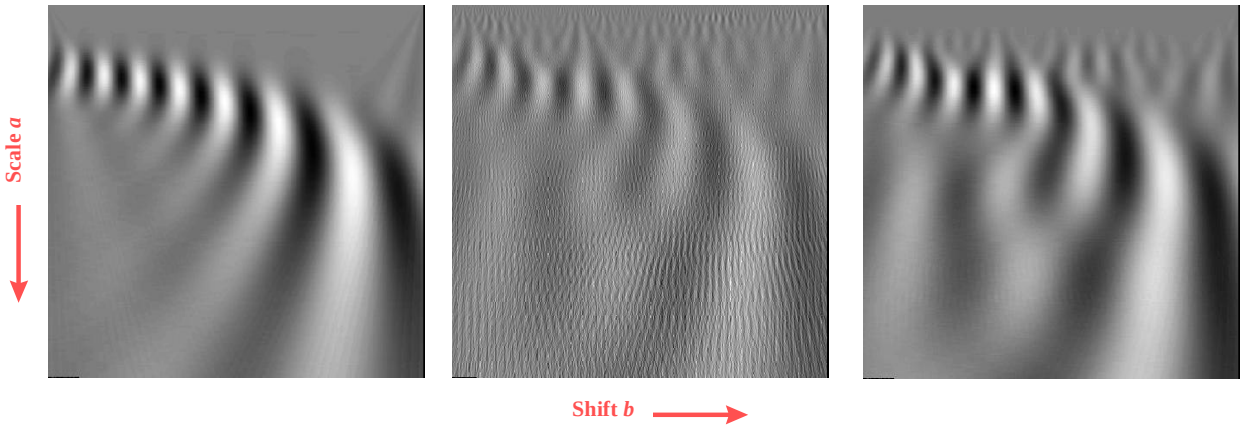

Figure 3. Continuous wavelet transform of pure, noisy, and denoised doppler signals (image plot)

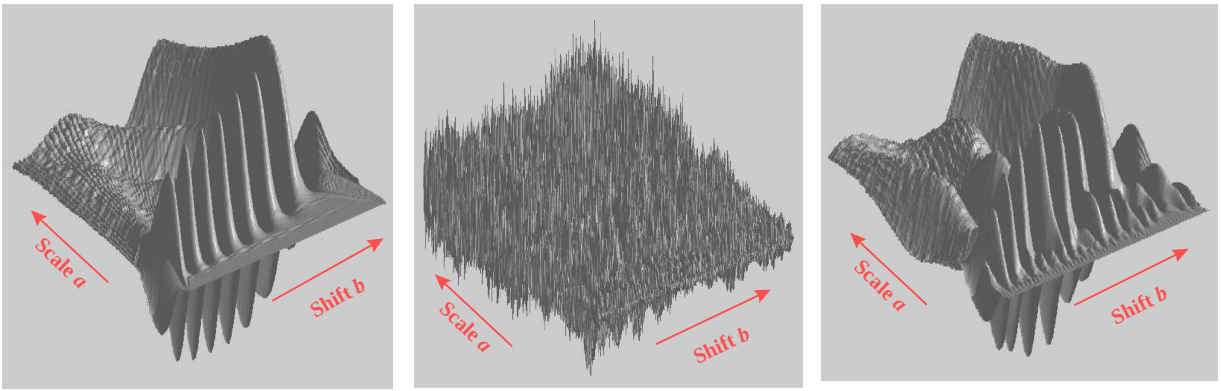

Figure 4. Continuous wavelet transform of pure, noisy, and denoised doppler signals (surface plot)

4. DOPPLER FREQUENCY ESTIMATION WITH WAVELETS AND NEURAL NETWORKS

The continuous wavelet transform correlates a doppler signal with time-localized wavelets at various scales and shifts. It gives the change in local signal scale over time, which in this case is the doppler period or inverse frequency. We use samples of the wavelet transform as inputs to neural networks which are trained to estimate the time-dependent frequency of noisy doppler signals. In this sense the wavelet transform samples can be considered signal features. Moreover, the sampled continuous wavelet transform constitutes a frame rather than a basis. Such a redundant representation allows more flexibility in the selection of signal features. In terms of the most efficient signal representation, these features should be orthogonal. However, such a representation in which the features are completely independent is less robust with respect to noise immunity and fault tolerance. The search for the best representation is therefore a tradeoff between redundancy and robustness.[12]

Figure 4 shows the neural network architecture we employ for doppler frequency estimation. The network is comprised of 3 layers of artificial neurons: an input layer, a middle or hidden layer, and an output layer. Signals flow forward through the network, that is from input layer to hidden layer to output layer. Such an architecture is known as a multilayer feedforward network or multilayer perceptron. The input neuron layer performs no processing, it merely provides means for coupling the input vectors to the hidden layer. The neurons in the middle layer sum the weighted network inputs, along with an internal bias for each neuron, then apply the nonlinear sigmoidal activation function $\sigma(v_j) = \tanh(v_j/2) = (1-e^{-v_j})/(1+e^{-v_j})$, where v_j is the weighted sum for neuron j. This sigmoidal nonlinearity limits the neuron outputs to (-1,1). The single output neuron computes the weighted sum of the outputs of the hidden neurons, along with its internal bias, without applying the sigmoidal function. This architecture is known to be a universal function approximator,[6] that is it can represent an arbitrary function arbitrarily well, given a sufficient number of

neurons in the hidden layer. The particular function mapping that the network performs is determined by the values of the weights between neuron layers and the internal neuron biases.

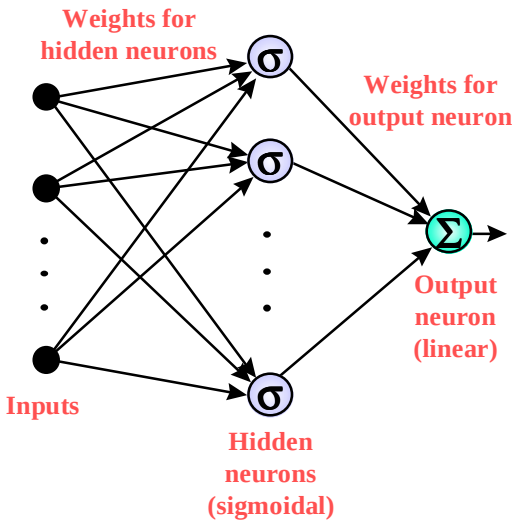

Figure 4. Neural network architecture for doppler frequency estimation

Various learning algorithms exist for computing the network weights and biases for a given problem. The most popular learning algorithm is backwards error propagation,[6] which attempts to minimize the squared error of the network through gradient descent in weight space. We can define the error signal for neuron j as

$$e_j(n) = d_j(n) - y_j(n), \tag{10}$$

where n indexes the training vectors, $d_j(n)$ is the desired response for neuron j, and $y_j(n)$ is the actual output of neuron j. The instantaneous value of the sum of squared errors $\frac{1}{2}e_j^2(n)$ over all neurons in the output layer of the network can then be written as

$$\mathsf{E}(n) = \frac{1}{2}\sum_{j\in C} e_j^2(n), \tag{11}$$

where the set C includes all neurons in the output layer and N is the number of vectors in the training set. The squared error averaged over all training vectors is then

$$\mathsf{E}_{\text{av}} = \frac{1}{N}\sum_{n=1}^{N}\mathsf{E}(n). \tag{12}$$

The average squared error E_{av} constitutes a cost function which is to be minimized. It is minimized approximately by iteratively reducing $\mathsf{E}(n)$ for each training vector. The correction $\Delta w_{ji}(n)$ applied to weight $w_{ji}(n)$ is defined by the delta rule

$$\Delta w_{ji}(n) = -\eta\frac{\partial \mathsf{E}(n)}{\partial w_{ji}(n)}, \tag{13}$$

where η is a parameter that determines the rate of learning. The minus sign in Eq. (13) results in gradient descent in weight space, that is weights are moved in the opposite direction to the error gradient.

We apply a powerful generalization of backwards error propagation known as the Levenberg-Marquardt weight update rule.[7] This rule can be written in matrix notation as

$$\Delta\mathbf{W} = \left(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \mu\mathbf{I}\right)^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{e}, \tag{14}$$

where $\Delta\mathbf{W}$ is the matrix of weight updates, $\mathbf{e}$ is the error vector, and $\mathbf{J}$ is the Jacobian matrix of derivatives of each error to each weight. If the parameter μ is very large, Eq. (14) approximates gradient descent, while if μ is small it becomes

the Gauss-Newton method. The Gauss-Newton method is faster and more accurate near an error minimum, so the idea is to shift towards the Gauss-Newton as quickly as possible. The parameter μ is therefore decreased after each successful step and increased only when a step increases the error. The Levenberg-Marquardt update rule is known to train networks much more quickly than standard backwards error propagation. However, it does require more memory, usually a factor of $C*N$ more, where C is the number of output neurons and N is the number of training vectors.

Figure 5 shows the way in which we sampled the wavelet transform of doppler signals to provide neural network input vectors. Each input vector is composed of 2 parts, with 16 elements for each part. The first 16 elements are the wavelet transform coefficients uniformly sampled in scale, for a single time. The next 16 elements are the transform coefficients sampled at the same scales, but for a single time that is shifted 6 samples ahead of the time of the first part of the input vector. This sampling at 2 different time shifts provides a degree of redundancy that we found greatly improves performance for noisy signals. We trained the networks with samples of transforms of pure doppler signals, sampling only every 4th time shift of the transform. For training outputs, we supplied the known instantaneous frequency of the pure signals for each time shift. Thus the networks were trained to estimate the instantaneous frequency of the doppler signals, given samples of their wavelet transform.

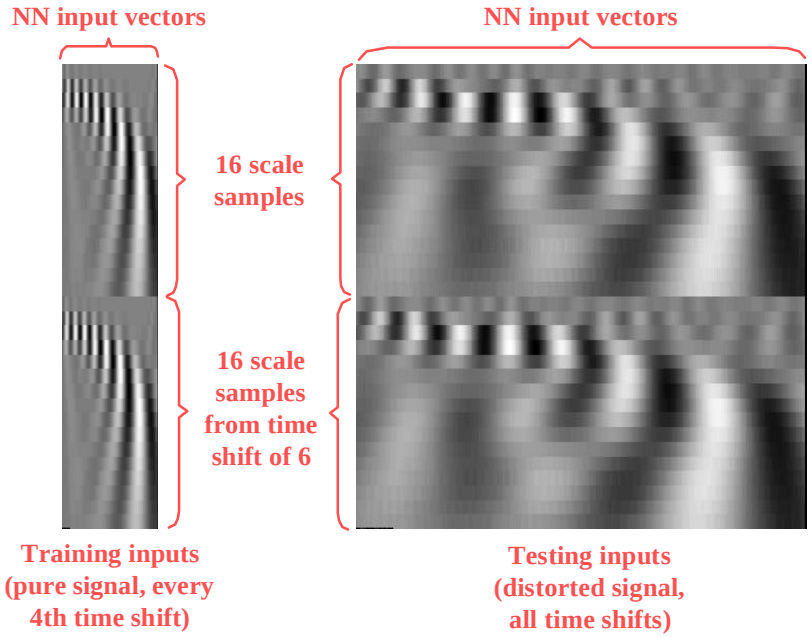

Figure 5. Inputs for neural network training and testing

After training for frequency estimation, we tested the networks with denoised versions of noisy doppler signals. The signal-to-noise ratios were sufficiently low that significant distortion remained after denoising, as for the signal in Figure 2. Figure 6 shows network test results for various signal-to-noise ratios, where the doppler signal frequency decreases nearly linearly over time, corresponding to a relatively large closest-approach distance α. The networks were tested for every time shift of the wavelet transform. Since the networks were trained with only every 4th sample, this shows their ability to generalize to other frequencies. Network performance is relatively good, but degrades with decreasing signal-to-noise ratio as would be expected. Figure 7 shows similar network performance for smaller α, which results in a more pronounced nonlinear change in frequency over time.

5. SUMMARY AND CONCLUSIONS

In this paper we have introduced a technique based on wavelets and neural networks for estimating instantaneous radar doppler frequency. This has direct application to the important military problem of radar proximity fuzing. Our technique first performs the Donoho wavelet denoising algorithm on the doppler signal. This improves performance for low signal-to-noise ratios, though significant distortion remains if the signal-to-noise ratio is sufficiently low. It then

computes the continuous wavelet transform of the denoised signal using the real Morlet wavelet. This provides a time-scale representation which is more appropriate for neural network processing. Samples of the wavelet transform are then used as inputs to multilayer perceptron neural networks. The networks are trained to estimate the instantaneous doppler frequency given the wavelet transform samples as input. They are trained with the Levenberg-Marquardt rule, a powerful generalization of gradient descent training. After training, the networks displayed the ability to generalize, that is to estimate doppler frequencies different than those used in training.

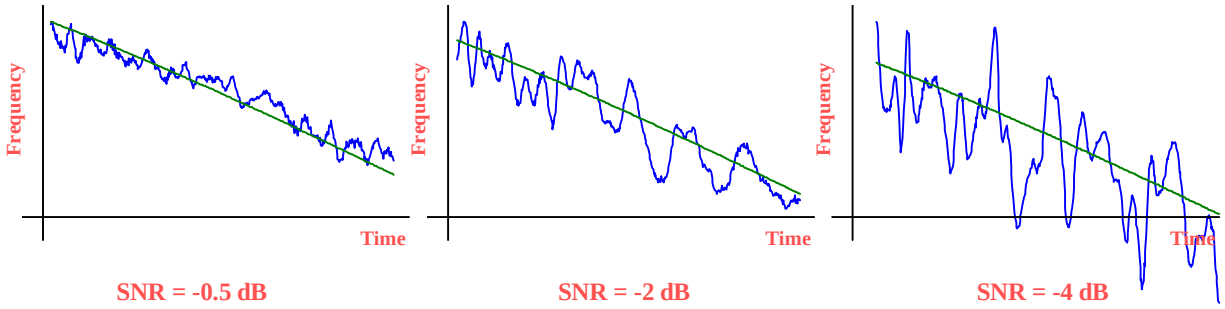

Figure 6. NN/WT estimation of doppler frequency (nearly linear change in frequency)

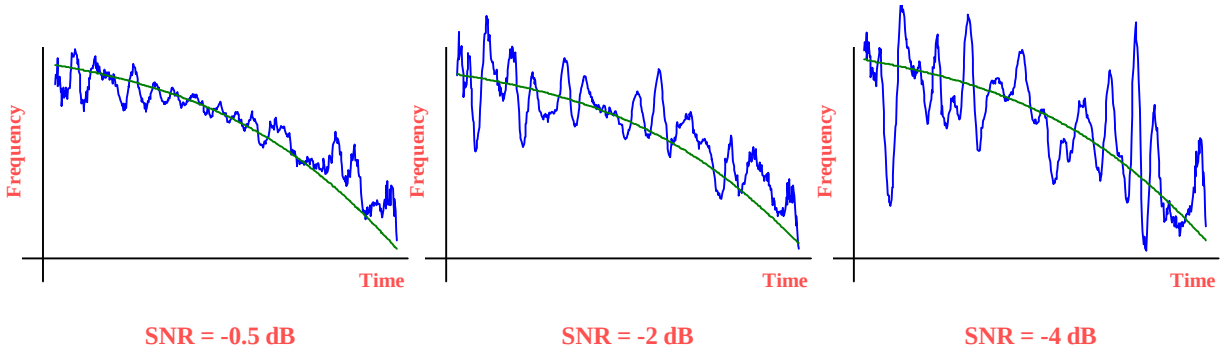

Figure 7. NN/WT estimation of doppler frequency (nonlinear change in frequency)

REFERENCES

1. M. Skolnik, *Introduction to Radar Systems*, 2nd edition, McGraw-Hill, 1980.
2. G. Strang, T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge, 1996.
3. A. Oppenheim, R. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, 1989.
4. D. Donoho, I. Johnstone, "Adapting to Unknown Smoothness via Wavelet Shrinkage," Technical report, Department of Statistics, Stanford University, 1992.
5. H. Szu, J. Garcia, B. Telfer, M. Raghuveer, "Super-Haar Designs of Wavelet Transforms," *Proc. SPIE Wavelet Applications III*, Vol. 2762, pp. 152-164, 1996.
6. S. Haykin, *Neural Networks – A Comprehensive Foundation*, Macmillan, 1994.
7. H. Demuth, M. Beale, *MATLAB Neural Network Toolbox User's Guide*, The MathWorks, 1996.
8. J. Morlet, G. Arens, I. Fourgeau, D. Girad, "Wave Propagation and Sampling Theory," *Geophysics*, Vol. 47, pp. 203-236, 1982.
9. Y. Meyer, Wavelets: Algorithms and Applications, SIAM, 1993.
10. S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. on Patt. Anal. and Mach. Intel.*, Vol. 11, No. 7, pp. 674-693, 1989.
11. A. Haar, "Zur Theory der Orthogonalen Funktionen-Systeme," *Math. Ann.*, Vol. 69, pp. 331-371, 1910.
12. H. Szu, P. Watanapongse, "Invertible Global Fourier and Local Wavelet Transforms versus Non-Invertible Global Principle Component Analysis and Local Independent Component Analysis," preprint.