Meas. Sci. Technol. **11** (2000) 784–794. Printed in the UK PII: S0957-0233(00)11654-6

Neural network data analysis for laser-induced thermal acoustics

Stefan Schlamp†, Hans G Hornung† and Eric B Cummings‡

† Graduate Aeronautical Laboratories at the California Institute of Technology, Pasadena, CA 91125, USA
‡ Sandia National Laboratories, Livermore, CA 94551, USA

E-mail: stefan@galcit.caltech.edu, hans@galcit.caltech.edu and labsmith@home.com (Eric B Cummings)

Received 4 February 2000, in final form and accepted for publication 5 April 2000

Abstract. A general, analytical closed-form solution for laser-induced thermal acoustic (LITA) signals using homodyne or heterodyne detection and using electrostrictive and thermal gratings is derived. A one-hidden-layer feed-forward neural network is trained using back-propagation learning and a steepest descent learning rule to extract the speed of sound and flow velocity from a heterodyne LITA signal. The effect of the network size on the performance is demonstrated. The accuracy is determined with a second set of LITA signals that were not used during the training phase. The accuracy is found to be better than that of a conventional frequency decomposition technique while being computationally as efficient. This data analysis method is robust with respect to noise, numerically stable and fast enough for real-time data analysis.

Keywords: four-wave mixing, velocimetry, thermometry, transient grating, scattering, non-intrusive, neural network, optimal filtering, regression

1. Introduction

While other laser-diagnostic techniques such as particle image velocimetry and laser Doppler anemometry are commercially available as packages of systems, this is not true for laser-induced thermal acoustics (LITA). One of the reasons is the complexity of the optical set-up and alignment for LITA. Secondly, the characteristic advantages of LITA (short test times, high signal levels and non-intrusiveness) are significant only for a small number of specialized applications. Finally, the data analysis can be cumbersome and requires user input, interaction and expertise to ensure the integrity of the results. The latter point is the focus of this paper. We seek a method that performs the data analysis accurately, computationally efficiently, robustly and autonomously. We propose and demonstrate the use of a feed-forward neural network for this task.

LITA (or laser-induced grating thermometry) is a four-wave mixing technique that has been used successfully for remote, non-intrusive and instantaneous measurements of the speed of sound [1], the thermal diffusivity [1] and the flow velocity of gases [2–5]. If the gas composition is known, the temperature can be obtained from the speed of sound [2, 6–8].

Two coherent intersecting pulsed laser beams (excitation beams) create by thermalization and/or electrostriction a density and consequently a refractive index grating in the sample volume that evolves over time. A third, continuous laser beam (the interrogation beam), directed at the Bragg angle onto the sample volume, is scattered into a coherent signal beam whose intensity depends on the instantaneous modulation depth of the refractive index grating.

Two detection approaches can be used. In homodyne detection, only the signal beam intensity is recorded over time. In heterodyne detection, the superposition of the signal beam and a reference beam is recorded. Since the signal beam has a Doppler shift proportional to the component of the fluid velocity along the direction normal to the grating, the latter approach makes this Doppler shift visible in the recorded signal.

Given the time-resolved heterodyne or homodyne signal, one of three methods is currently employed to obtain the speed of sound and, in the case of heterodyne detection, the fluid velocity.

(i) One can use a technique whereby a signal obtained from a theoretical model is used for a nonlinear least-squares fit to the experimental data. The speed of sound and flow velocity are floating parameters during the fitting. This technique requires a theoretical model. References [9, 10] provide such a model. The theory in those references assumes, however, homodyne detection.
(ii) The Pronys method ([7, 11]) is a simpler version of the fitting technique mentioned above. A linear combination of damped complex potentials is fitted to the data. No theoretical method is needed.
(iii) From the location of the peaks in the power spectrum one can infer the Brillouin frequency and the Doppler frequency. The speed of sound and the flow velocity can be deduced.

Previous research (e.g. [1, 9]) has shown that LITA can also be used to measure the thermal diffusivity. The thermal diffusivity governs the exponential decay of the LITA signal. Hence, the thermal diffusivity can be calculated from the decay time constant of the LITA signal. However, the last two methods can only extract frequencies from the data. Only the first method is capable of extracting signal parameters other than frequencies from signals. Although the full fitting technique is computationally expensive ($\mathcal{O}(n^3)$, where n is the number of data points in a signal), it is more accurate than the frequency decomposition technique since it represents an optimal filter for the noisy data under the assumption that the theoretical model is a correct representation of the experimental signal. The frequency decomposition technique is computationally cheap ($\mathcal{O}(n \log n)$) and can be performed in real time at driver laser frequencies of $\mathcal{O}(10\,\text{Hz})$.

Artificial neural networks are heavily used for all sorts of classification problems (e.g. [12, 13]), robot control [14, 15], speech recognition [16, 17] and image processing [18] but less extensively for data analysis in engineering problems. This is mostly due to a lack of familiarity of the engineering community with the concepts of neural networks. Secondly, the lack of analytical tools for *a priori* prediction of the network performance, optimal learning algorithms and amount of training necessary and for guidance in the design of the network architecture is another source of dissatisfaction.

We use only very basic neural network concepts to achieve the results presented in section 5. This demonstrates that even simple neural network implementations can yield very satisfactory results. Excessive empirical trial and error with the network architecture and a learning scheme can potentially improve the network performance and accelerate the training process but this is not necessary in order to arrive at satisfactory results. The disadvantage of the neural network implementation presented in this paper is the requirement to train the neural network prior to its use. Without advanced numerical schemes this can take a considerable amount of time. It should be pointed out that the trained network performs the data analysis fully autonomously. The neural network outputs are direct functions of the network inputs so that numerical instability and poor convergence behaviour do not pose problems.

In section 2 we will present the theoretical framework for LITA using either heterodyne or homodyne detection. It is an extension of the work presented in [9, 10]. The solution will be used to create a set of LITA traces which are used to train a neural network and to test its performance. In section 3 we will present a summary of the basic theory of feed-forward neural networks and the back-propagation learning rule. This section should provide just enough information for the reader who is unfamiliar with neural networks to follow this paper. For more background information, the interested reader is directed to [19, 20, 29]. Section 4 shows how the theory of feed-forward networks can be implemented for the LITA data analysis. Section 5 gives some results of the theoretical derivation from section 2 and will show the performance of the neural network data analysis.

2. LITA theory

The electrical field of the scattered LITA signal beam in Fourier space is [10]

$$\frac{E_s(\vec{q}, \vec{R}, t)}{P_0(t)} = -\frac{k_s^2\omega^2}{4\pi R}\chi(f_0)\exp[\mathrm{i}(\vec{k}_s\cdot\vec{R} - f_0 t)] \circ \Re(A_{P1}\Phi_{P1}^{(d,0)} + A_{P2}\Phi_{P2}^{(d,0)} + A_T\Phi_T^{(d,0)} + A_D\Phi_D^{(d,0)}). \quad (1)$$

$A_{P1,P2,T,D}$ are the relative amplitudes of the acoustic waves, the thermal grating and finite driving-time terms. $\Phi_{P1,P2,T,D}$ contain the temporal and spatial profiles of these terms. $\circ$ represents a temporal convolution. Equation (1) contains the effects of finite beam sizes, single-rate thermalization and electrostriction. For a more detailed explanation of the terms in equation (1), the reader is directed to [9, 10].

We superimpose a reference beam with the same Gaussian geometry and direction as the signal beam of the form

$$E_{ref} = E_r + E_r^* \quad (2a)$$

$$E_r = \frac{A}{2}\exp[\mathrm{i}\vec{k}_s\cdot\vec{R} - \mathrm{i}(f_0 - \Delta f_{ref})t + \mathrm{i}\tilde{\phi}]\exp\left(-\left|\frac{\hat{e}_0\otimes\vec{r}}{\sigma}\right|^2\right) \quad (2b)$$

to the signal beam. f_0 is the interrogation beam frequency, σ its Gaussian half-width and k_s its wavevector magnitude. $\otimes$ and * denote the vector cross product and the complex conjugate, respectively. $\tilde{\phi}$ is used to model a phase shift between the reference beam and the signal beam. In the final result (equation (6)), $\tilde{\phi}$ produces a phase shift between the Brillouin frequency and the Doppler frequency component in the signal (see figure 4 later). This effect was observed experimentally in [4], where $\tilde{\phi}$ took random values for every signal. This is caused by small time-varying perturbations (e.g. vibrations) in the optical set-up. The frequency of the reference beam is assumed to be shifted by Δf_{ref} from that of the interrogation beam. In experiments, this frequency shift could be introduced by a Bragg cell in the beam path. Its purpose is to improve the accuracy for low speed velocity measurements and to remove the direction ambiguity from the velocity measurements.

For the time being, we do not specify the (temporally constant) intensity of the reference beam relative to the signal beam and use the prefactor A to keep equation (2) general. Furthermore, we will absorb any multiplicative constants that will show up along the way into A. For $A = 0$, i.e. zero reference beam intensity, we expect to recover the result for homodyne detection.

Since the Fourier transform is a linear operation, we can superimpose the Fourier transform of equation (2a) directly upon equation (1). The Fourier transform of equation (2a) is:

$$E_{ref}(\vec{q}, t) = A_{ref}\Phi_{ref}\exp[\mathrm{i}(\vec{k}_s\cdot R - f_0 t)] \quad (3a)$$

where

$$A_{ref} = A\exp(\mathrm{i}\tilde{\phi}) \quad (3b)$$

$$\Phi_{ref} = \Sigma_{ref}\Psi_{ref} \quad (3c)$$

$$\Psi_{ref} = \exp\left(-\frac{\sigma_y^2}{4}(q_y - q_\psi)^2 - \frac{\sigma_z^2}{4}q_z^2\right) \quad (3d)$$

$$\Sigma_{ref} = \exp(\mathrm{i}\Delta f_{ref} t) \quad (3e)$$

$$\sigma_y = \frac{\sigma}{\sin\psi} \qquad \sigma_z = \sigma \tag{3f}$$

and q_ψ is the phase-matched scattering or grating vector. Note that, in equation (3*d*), we neglected a second lobe centred at $q_y = -q_\psi$ as well as any variations in the x-direction. The latter is justified by the fact that, for small excitation beam crossing angles, the spatial extent of the grating will be much larger in the x-direction than it is in the y- and z-directions.

Now we can superimpose the signal and reference beams by including equation (3*a*) in equation (1) as follows:

$$\begin{aligned}\frac{E_s(\vec{q},\vec{R},t)}{P_0(t)} &= -\frac{k_s^2\omega^2}{4\pi R}\chi(f_0)\exp[\mathrm{i}(\vec{k}_s\cdot\vec{R}-f_0t)]\\ &\circ\Re(A_{P1}\Phi_{P1}^{(d,0)}+A_{P2}\Phi_{P2}^{(d,0)}+A_T\Phi_T^{(d,0)}+A_D\Phi_D^{(d,0)}\\ &+A_{ref}\Phi_{ref}).\end{aligned}\tag{4}$$

Detectors measure the intensity of the electrical field, i.e. the square of the modulus of equation (4). Also, at this point we assume that the excitation laser pulse is short compared with all other time scales and that we can approximate it by a Dirac delta function. This simplifies the temporal convolution into a simple multiplication. Hence, the signal intensity using heterodyne detection is then

$$\begin{aligned}\mathcal{L}_{het} \propto\ &(A_{P1}\Phi_{P1}^{(d,0)}+A_{P2}\Phi_{P2}^{(d,0)}+A_T\Phi_T^{(d,0)}\\ &+A_D\Phi_D^{(d,0)}+A_{ref}\Phi_{ref})\\ &\times(A_{P1}^*\Phi_{P1}^{(d,0)*}+A_{P2}^*\Phi_{P2}^{(d,0)*}+A_T\Phi_T^{(d,0)*}\\ &+A_D\Phi_D^{(d,0)*}+A_{ref}^*\Phi_{ref}^*).\end{aligned}\tag{5}$$

Finally, we have to integrate equation (5) over the detector area. In the limit of a small detector, we can multiply equation (5) by the detector area. In the limit of a large detector we can use infinite spatial integrals of equation (5). In the latter case, the result is

$$\begin{aligned}\mathcal{L}_{het} \propto\ &\exp\left[-\frac{8\sigma_y^2}{Y^2(Y^2+2\sigma_y^2)}\left(\frac{c_st}{2}\right)^2\right]\\ &\times[(P_1+P_2)(T^*+D^*)+(P_1^*+P_2^*)(T+D)]\\ &+\exp\left(-\frac{8\sigma_y^2}{Y^2(Y^2+2\sigma_y^2)}(c_st)^2\right)(P_1P_2^*+P_1^*P_2)\\ &+\exp\left[-\frac{8\sigma_y^2}{(Y^2+2\sigma_y^2)(Y^2+\sigma_y^2)}\left(\frac{\overline{\eta}+vt}{2}\right)^2\right]\\ &\times[(T+D)R^*+(T^*+D^*)R]\\ &+\exp\left[-\frac{8\sigma_y^2}{(Y^2+2\sigma_y^2)(Y^2+\sigma_y^2)}\right.\\ &\left.\times\left(\frac{\overline{\eta}+(v+c_s)t}{2}\right)^2\right](P_1R^*+P_1^*R)\\ &+\exp\left[-\frac{8\sigma_y^2}{(Y^2+2\sigma_y^2)(Y^2+\sigma_y^2)}\right.\\ &\left.\times\left(\frac{\overline{\eta}+(v-c_s)t}{2}\right)^2\right](P_2R^*+P_2^*R)+P_1P_1^*\\ &+P_2P_2^*+TT^*+TD^*+T^*D+DD^*+RR^*\end{aligned}\tag{6}$$

where $P_1 = A_{P1}\Sigma_{P1}$, $T^* = A_T^*\Sigma_T^*$, $R = A_{ref}\Sigma_{ref}$, etc. The term RR^* at the very end of equation (6) represents the constant reference beam intensity in the form of a dc offset. We see that, for $A = 0$ ($A_{ref} = 0$), the solution collapses onto the solution for homodyne detection [10]. Since we do not attempt to find an expression for the *absolute* LITA signal intensity, we, as in equation (5), neglect multiplicative constants.

3. Neural network formulation

Multilayer feed-forward networks were first studied by Rosenblatt [21] in the late 1950s but, owing to the absence of a training algorithm for multilayer networks, interest subsided until the reporting of the back-propagation learning rule in 1986 [22]. Back-propagation has actually been discovered independently at least three other times [23–26]. Reference [23] refers to work done, on a related problem, in the early 1950s [27].

The network we are considering (figure 1) has n input units x_i, $i = 0,\ldots,n$, h units z_j in the hidden layer, $j = 1,\ldots,h$, and m output units y_k, $k = 1,\ldots,m$. Each unit is connected to every unit in the next higher layer. A weight is assigned to each such connection. A normalized, time-discretized LITA signal $\mathcal{L}(t_i)$ will be used as input. So, n will be chosen to be the number of points in the signal trace.

The values z_j of the units in the hidden layer are determined by the values of the input units, the weights w_{ji} (from input unit i to hidden unit j) and an activation function $\sigma(\cdot)$ by

$$z_j = \sigma\left(\sum_{i=0}^{n} w_{ji}x_i\right). \tag{7}$$

Note that the index counts from zero to n and we define $x_0 = -1$ and call it a bias unit. Its significance lies in its mathematical and algorithmic convenience. It allows an affine transformation of the inputs (i.e. one involving a linear combination of inputs $a_1x_1+a_2x_2+\ldots$ plus an offset a_0) to be treated as a linear combination; thus, all weights, including a_0, may be treated uniformly, rather than requiring separate treatment for a_0.

Similarly, the values of the output units are given by

$$y_k = \sigma\left(\sum_{j=0}^{h} v_{kj}z_j\right) \tag{8}$$

where $\sigma(\cdot)$ is the same activation function as before, $z_0 = -1$ and v_{kj} is the weight from hidden unit j to output unit k. The only requirements for the activation function $\sigma(\cdot)$ are that it is nonlinear, differentiable and bounded. Hidden layers do not expand the network's capabilities if the activation function is linear. This is because any linear combination of linear functions is again only a linear function. The requirement that $\sigma(\cdot)$ be differentiable is due to the back-propagation learning rule. The boundedness of $\sigma(\cdot)$ is not a strict requirement but it is helpful for avoiding overflows. We use

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{9}$$

but other choices such as $\sigma(x) = \tanh(x)$ are possible.

Each output unit represents one parameter that we want to filter from the LITA signal in the input. By the choice of the activation function (equation (9)), the output units can only

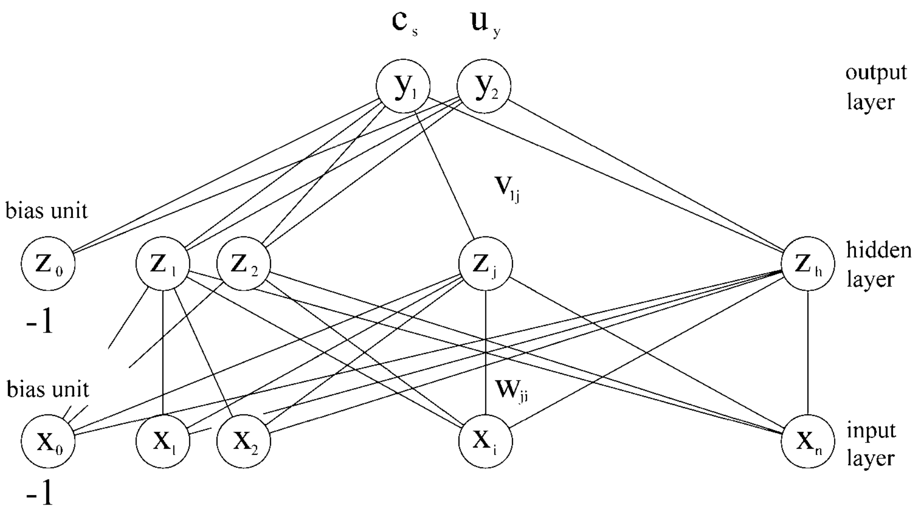

Figure 1. The layout of a one-hidden-layer feed-forward neural network.

have values in the range 0–1 and we must therefore scale the outputs to fall in the range of the target values, i.e. the speed of sound and flow velocity (see equations (16*a*) and (16*b*)).

We see that, given the proper weights w_{ji} and v_{kj} and a LITA signal as input, the y_k's can easily be found. The problem is that of how to find the correct weights that perform the filtering correctly. This process is referred to as training of the neural network.

Assume that we have a number $\mu = 1, \ldots, N$ of LITA signals (the training set) with known correct output values η_k^μ (called target values) but the network with incorrect weights returns values at the output units y_k^μ. One possibility for defining an error measure is by writing

$$E = \frac{1}{2}\sum_{\mu=1}^{N}\sum_{k=1}^{m}(y_k^\mu - \eta_k^\mu)^2. \tag{10}$$

This represents the sum of the squares of all individual errors. E is zero if and only if $y_k^\mu = \eta_k^\mu$ for all k and μ. By using equations (8) and (9) in equation (10) we can differentiate with respect to the weights v_{kj} and obtain

$$\frac{\partial E}{\partial v_{kj}} = \sum_{\mu=1}^{N}(y_k^\mu - \eta_k^\mu)y_k^\mu(1 - y_k^\mu)z_j^\mu. \tag{11}$$

The choice of the activation function in equation (9) allows us to express $\sigma'(x) = \mathrm{d}\sigma/\mathrm{d}x$ by $\sigma(x)$ itself,

$$\sigma'(x) = \frac{\mathrm{d}\sigma(x)}{\mathrm{d}x} = \sigma(x)(1 - \sigma(x)). \tag{12}$$

This has the advantage that we do not have to compute σ' during the training.

We can continue, use equation (7) in equation (11) and differentiate with respect to the weights w_{ji}. The result is

$$\frac{\partial E}{\partial w_{ji}} = \sum_{\mu=1}^{N} x_i^\mu \sum_{l=1}^{m} v_{lj} z_j^\mu(1 - z_j^\mu)(y_l^\mu - \eta_l^\mu)y_l^\mu(1 - y_l^\mu). \tag{13}$$

This gives us all the tools we need. By updating the weights according to

$$\Delta w_{ji} = -\eta\frac{\partial E}{\partial w_{ji}}$$
$$\Delta v_{kj} = -\eta\frac{\partial E}{\partial v_{kj}} \tag{14}$$

where η is called the learning rate, the error measure E can be reduced iteratively, provided that η is sufficiently small. Equation (14) represents the method of steepest descent. More sophisticated updating rules than the one shown in equation (14) can be used, These give faster convergence, e.g. by introducing a 'momentum' term [19]:

$$\Delta w_{ji}(t+1) = -\eta\frac{\partial E}{\partial w_{ji}} + \alpha\Delta w_{ji}(t)$$
$$\Delta v_{kj}(t+1) = -\eta\frac{\partial E}{\partial v_{kj}} + \alpha\Delta v_{kj}(t). \tag{15}$$

α must be between 0 and 1. Commonly a value of 0.9 is chosen.

Some authors [19, 30–33] have proposed an adaptive scheme of adjusting the parameters α and η during the training to further improve the convergence behaviour. In most adaptive schemes η is increased by a small additive constant if the cost function E decreases monotonically over a number of iterations. An increase of E during the training, on the other hand, normally indicates that the minimization algorithm overshot the minimum and a reduction in step size is appropriate. Hence, if E increases over one training iteration, η is decreased geometrically (i.e. multiplied by a constant between zero and unity).

As stated in section 4, however, we change the training set slightly after every iteration to prevent the network from over-training. This introduces noise which in turn prevents us from using such an easy adaptive scheme. Owing to the high number of connections (50 000+) we cannot use a more efficient (but memory demanding) minimization scheme, such as the Levenberg–Marquardt algorithm.

Hence, we know what values to use for n, m and x_i, how to calculate y_k and how to find appropriate weights.

The number of hidden units required cannot be precisely determined *a priori* but has to be found empirically. It can be shown [34, 35] that, given a sufficient number of hidden units, a one-hidden-layer feed-forward network is capable of approximating *any* continuous function to arbitrary accuracy. From the derivation in section 2 we can conclude that such a continuous function exists.

Two kinds of errors can be defined. The 'bias' is the part of the error which is due to deficiencies in the network architecture, i.e. an insufficient number of layers or of hidden units. If h is too large on the other hand, the network will learn the task 'too' well, meaning that it will specialize on the training set but will perform poorly on data that were not used in the training phase. This is referred to as over-training. The 'variance' is the part of the error that is due to the fact that the training set does not cover the entire space of inputs.

4. The set-up

If, for a given application, the ranges of target values for c_s and u_y are known ($\tilde{\phi} = 0$–2π always), we can create a training set by using uniformly distributed random values for these parameters in equation (6). Figure 2 shows a typical trace as it was used in a training set. As only a preprocessing step, all signals are normalized with respect to a (positive) peak value of unity. The training set has to consist of a sufficient number of traces in order to cover the parameter space well (i.e. to reduce the variance) and to avoid over-training of the network. In general the number of training signals should be much larger than the number of units in the hidden layer. To avoid having an excessively large training set, the authors chose to vary the training set over the course of the training. After each updating of the weights, one randomly picked trace from the training set was replaced by a new trace with new random values for u_y and c_s. This procedure ensures that good coverage of the input parameter space is obtained while limiting the size of the training set at any one time. In addition, this scheme introduces noise into the minimization procedure. This helps to prevent the minimization from converging to a local minimum rather than to the global minimum.

As a test case, we consider atmospheric air at rest which is accelerated isentropically to $M = 0.9$. Hence, we would expect to measure speeds of sound in the range $c_s = 320$–345 m s^{-1} and flow velocities $u_y = 0$–288 m s^{-1}. The phase shift $\tilde{\phi}$ is a random variable out of the range 0–2π. The experimental parameters in equation (6) such as laser wavelengths, the excitation beam crossing angle and beam half-widths are set to typical values. Instead of using c_s and u_y directly as values for η_k^μ, these values have to be scaled to fall into the range 0–1. This is necessary because the range of outputs of the neural network is limited by the choice of the activation function to be 0–1, where 0 and 1 are approached asymptotically. By using the scaling

$$\eta_1^\mu = \frac{1}{2}\left(\frac{c_s^\mu - 320}{345 - 320} + \frac{1}{2}\right) = 0.25\text{–}0.75 \qquad (16a)$$

$$\eta_2^\mu = \frac{1}{2}\left(\frac{u_y^\mu}{288 - 0} + \frac{1}{2}\right) = 0.25\text{–}0.75 \qquad (16b)$$

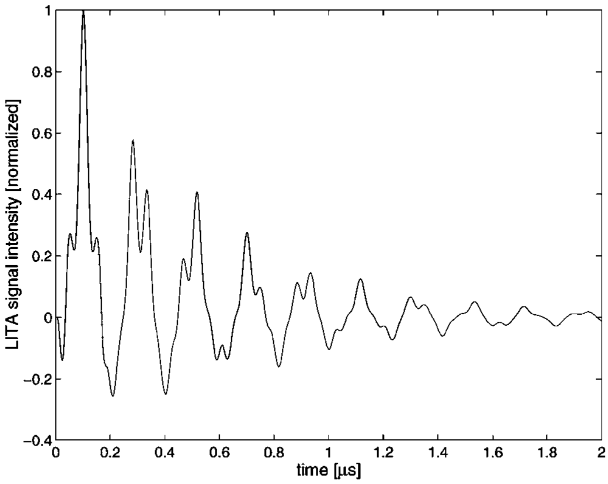

Figure 2. A typical theoretical trace as used in the training and test set for $c_s = 340$ m s^{-1} and $u_y = 100$ m s^{-1}.

we retain comparable network sensitivities over the full range of speeds of sound and flow velocities. Furthermore, since the error measure in equation (10) minimizes the sum of the total errors rather than the sum of the relative errors, the difference between these two error measures is reduced. It would be possible to adjust equation (10) such that the relative errors are minimized but, as we will show later, minimizing for absolute errors avoids problems with the Fourier limit at very low flow speeds.

The training was performed over up to 1.5 million iterations with a fixed learning rate of $\eta = 0.0075$ and $\alpha = 0.9$. The training was stopped when E stopped decreasing. The number of hidden units was set to the values $h = 5$, 10, 20 and 50. The weights w_{ji} and v_{kj} were set to small random values initially. The training set consisted of $N = 250$ traces with random values for c_s, u_y and $\tilde{\phi}$ out of the ranges specified above. Each trace consists of 1000 data points, i.e. $n = 1000$. After each iteration, a random trace from the training set is replaced by a new trace with random u_y, c_s and $\tilde{\phi}$ to reduce over-training. A validation set of 250 traces which are not used for the training is also created. The validation set remains unchanged during the training. It is used to check for over-training.

5. Results

First, look at some results of the expanded theoretical model from section 2. Figures 3 and 4 show heterodyne LITA signals from thermal gratings calculated from equation (6). The speed of sound is represented by the Brillouin frequency (the high-frequency component in signals) and the flow velocity is represented by its Doppler shift (the low-frequency component). At $M = 1$, these frequencies match. All traces depicted have $M = 0.11$ and $c_s = 345$ m s^{-1}.

Figure 3 shows the influence of the reference beam intensity on the signal. The reference beam intensity increases by a factor of ten between each plot in figures 3(b)–(f). The signal shape does not change even for stronger reference beams than that in figure 3(f).

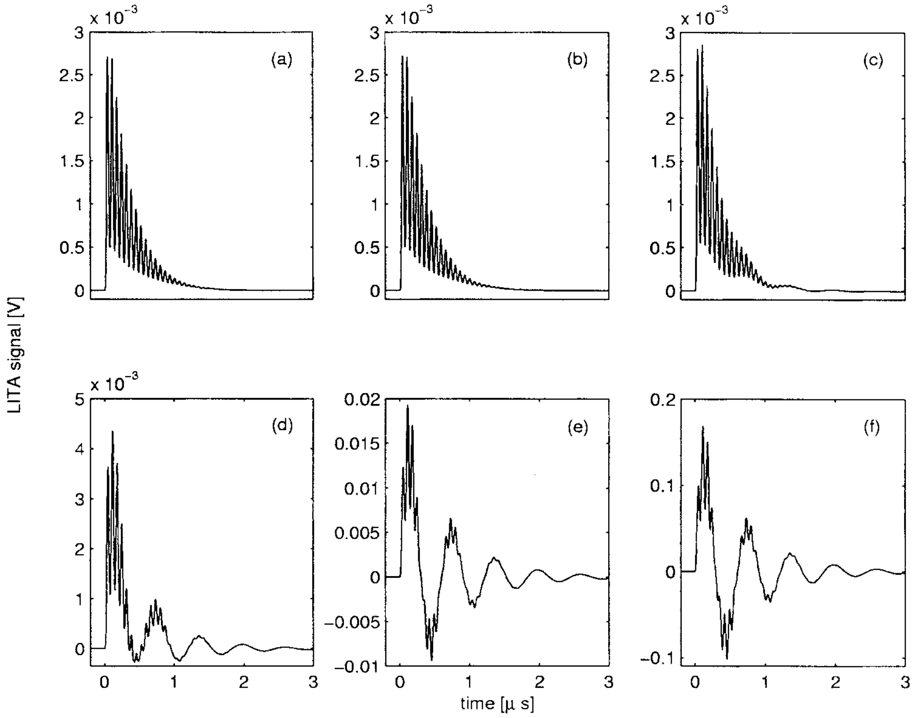

Figure 3. Theoretical heterodyne LITA signals for atmospheric air at $M = 0.11$ from equation (6) for various reference beam intensities: (*a*) $A = 0$ (a homodyne signal), (*b*) $A = 0.0001$, (*c*) $A = 0.001$, (*d*) $A = 0.01$, (*e*) $A = 0.1$ and (*f*) $A = 1$.

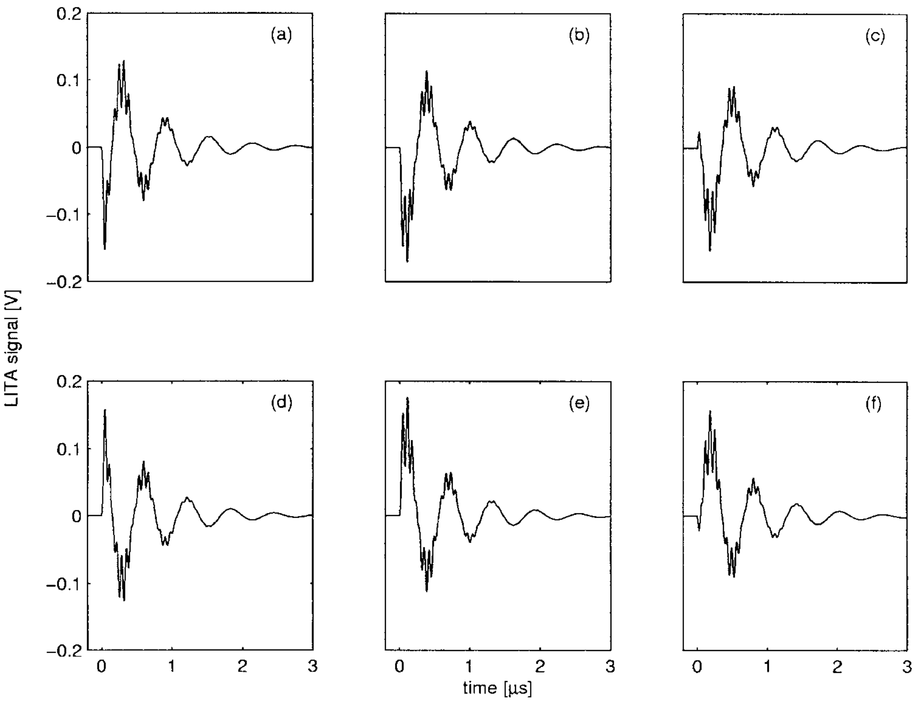

Figure 4. Theoretical traces in the limit of a strong reference beam and for the same flow conditions as those in figure 3 but with different phase shifts between the Brillouin frequency and the Doppler shift: (*a*) $\tilde{\phi} = 0°$, (*b*) $\tilde{\phi} = 60°$, (*c*) $\tilde{\phi} = 120°$, (*d*) $\tilde{\phi} = 180°$, (*e*) $\tilde{\phi} = 240°$ and (*f*) $\tilde{\phi} = 300°$.

Even though it seems that the signal becomes stronger, one has to remember that the dc offset caused by the reference beam has been subtracted from figures 3 and 4. The offset grows like A^2 whereas the signal excluding the offset grows linearly with A. Hence, the signal amplitude relative to the dc offset is actually decreases on going from figure 3(*b*) to figure 3(*f*). All traces used for training and validation of the neural network were chosen to be in the limit of a strong reference beam. Figure 4 shows the same trace with varying phase shifts $\tilde{\phi}$. All signals would correspond to the same

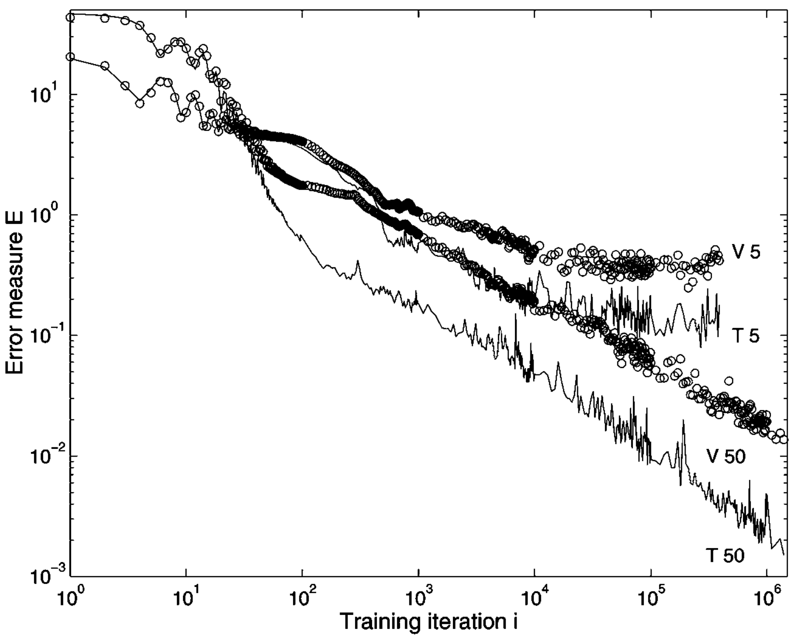

Figure 5. The error measure E from equation (10) during the training phase calculated using the training set (full curves, T50 and T5) and the validation set (symbols, V50 and V5) for 50 hidden units (V50 and T50) and five hidden units (V5 and T5). Only 100 data points per decade are plotted.

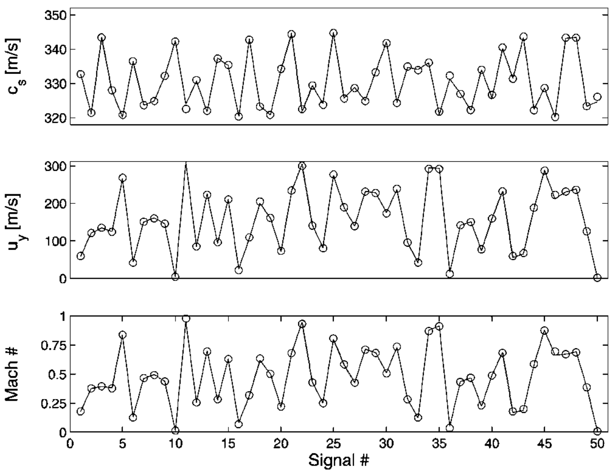

Figure 6. Direct comparisons between neural network outputs (symbols) and target values (lines) for the speed of sound (top), flow velocity (centre) and the derived quantity of the Mach number (bottom) with $h = 50$.

flow velocity, speed of sound, etc and we expect the neural network to be uninfluenced by different values of $\tilde{\phi}$.

Figure 5 shows how the error measure E given by equation (10) decreases during the training. Depicted are the cases with the most ($h = 50$) and the least hidden units ($h = 5$). The full curves show E calculated using the training set; the symbols plot E for the validation set. The difference between the two curves indicates the amount by which the network has 'specialized' for the training set. If the top curve were to level off while E for the training set continued to decrease, the performance limit of the network would be reached and any additional training would only represent over-training. For example, we see in figure 5 that the two curves for $h = 50$ move in parallel and that E has not reached an asymptotic value by the time the training is stopped. Continued training could improve the performance further. For the case $h = 5$, in contrast, the errors both for the training set and for the validation set have asymptotically reached minima. The curves for $h = 10$ and 20 are not plotted in figure 5 but we can summarize that the minimum value of E decreases with increasing h.

Figure 6 gives a direct comparison between the neural network outputs and the target values for a subsample of the validation set. We see very good agreement even for the derived Mach number ($M = u_y/c_s$). To check the robustness of the approach for analysis of neural network data with respect to noise, we added varying degrees of Gaussian noise to the validation set (see examples in figure 7). Figure 8 shows the RMS error of the neural network output for c_s and u_y versus the signal-to-noise ratio (SNR). The errors increase slowly with the SNR. The more hidden units the lower the error levels. In the limit of zero noise, the errors for the speed of sound are 0.25 m s^{-1} ($h = 50$) and 1 m s^{-1} ($h = 5$). For the velocity measurements, the performance also depends strongly on the number of hidden units. The errors in the same limit are 2 m s^{-1} ($h = 50$) and 20 m s^{-1} ($h = 5$). For very low SNRs, the errors are large and independent of the number of hidden units. They correspond to mostly random network outputs.

Besides the bias and the variance there is a third error source for this particular application. The Doppler shift is proportional to the flow velocity. However, the signal lifetime is limited by diffusion and the finite size of the laser beams. Hence, at very low frequencies there will be only a fraction of a cycle within a signal, making accurate frequency measurements impossible. This so-called Fourier limit represents a theoretical limit to all data analysis techniques. Figure 9 shows the uncertainty of the neural network output for c_s and u_y versus the flow velocity u_y. Note that the

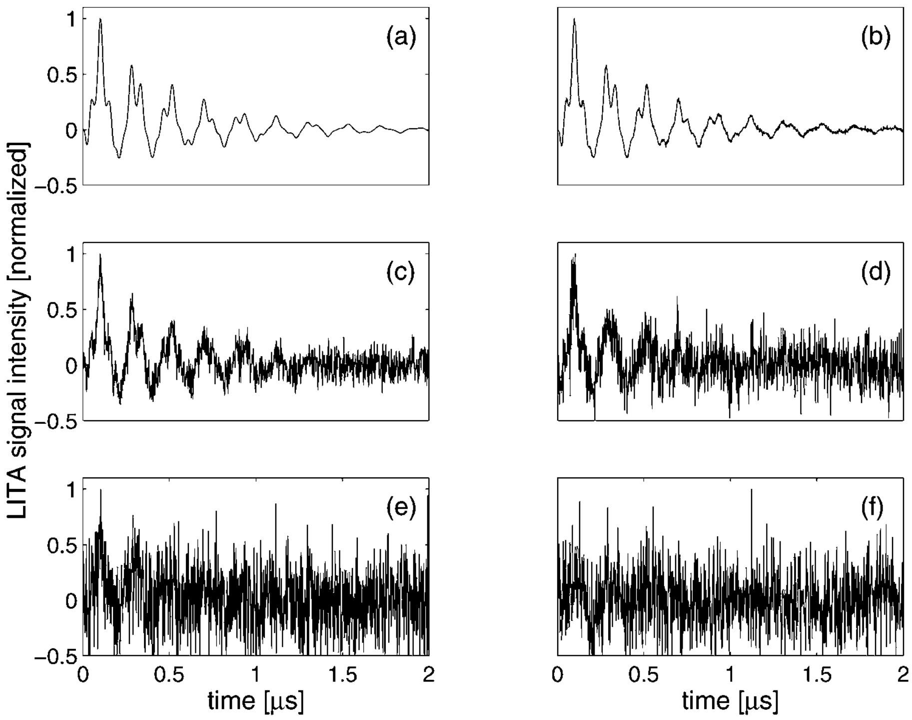

Figure 7. LITA traces with various SNRs: (*a*) ∞ (no noise), (*b*) 100, (*c*) 10, (*d*) 5, (*e*) 2 and (*f*) 1.

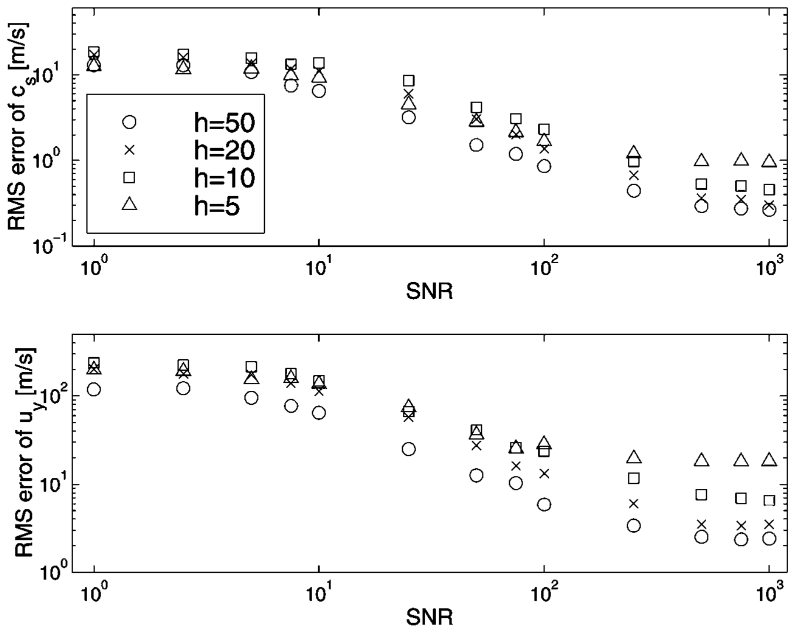

Figure 8. RMS errors for c_s (top) and u_y (bottom) as functions of the signal-to-noise ratio (SNR).

absolute uncertainty of u_y is almost constant except for low flow velocities. It increases by one order of magnitude for flow speeds below 30 m s^{-1}. For flow velocities close to zero, the errors become independent of the number of hidden units.

In figure 10 c_s–u_y combinations, covering the whole range of parameters on which the network was trained, are used to create signals that are used as inputs to the neural network with 50 hidden units. Correct (input) values are plotted as circles. The actual network outputs are plotted as crosses. The errors are very small for most of the parameter space. Only for small flow velocities and in some other regions do we observe noticeable errors.

6. Discussion and conclusions

A general expression for LITA signals from thermal or electrostrictive gratings and using homodyne or heterodyne detection has been derived and found to be in good agreement with experiments [4]. The shape of heterodyne LITA signals

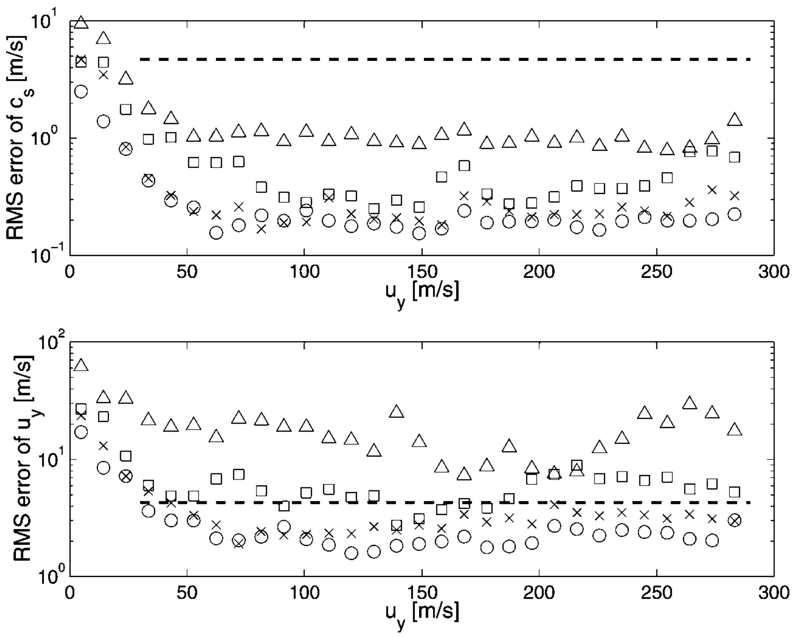

Figure 9. RMS errors of c_s and u_y as functions of u_y. The symbols are the same as in figure 8. The broken lines represent uncertainties obtained using a frequency decomposition technique on the same data.

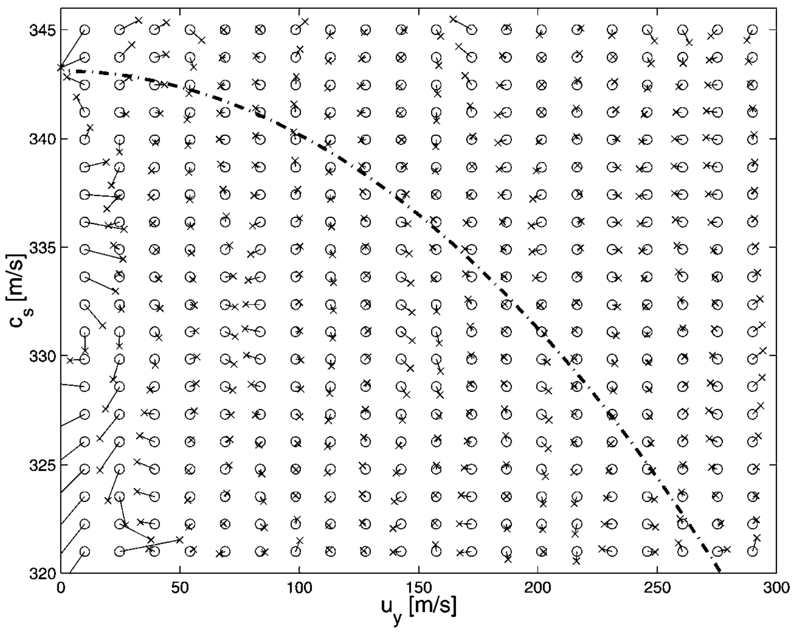

Figure 10. Neural network outputs versus correct values over the whole of the array of u_y–c_s combinations for which the neural network was trained. The broken line represents the isentropic expansion of air ($T_t = 293$ K) to $M = 0.9$.

approaches a limit for strong reference beams. Experiments have shown that there is a phase shift between the oscillations at the Brillouin frequency and the Doppler frequency. This phase shift is due to vibrations in the optical components, temperature variations and other unpredictable effects. It varies randomly from signal to signal. The theory presented takes this effect into account.

We implemented a one-hidden-layer feed-forward neural network algorithm for the data analysis. Its accuracy was very good with the exception of the regime of flow speeds below 50 m s^{-1}. This is well before the Fourier limit should become significant. In fact, experimental results with the fitting technique give much better results in this velocity regime [4]. Also, this theoretical limit should affect only the velocity measurements, not the sound speed result. Using more than 50 hidden units could possibly mitigate this problem. In addition, we see in figure 5 that E is still decreasing for the case of $h = 50$ when the training is stopped. This means that the accuracy could be further improved by prolonging the training phase. An optimized learning rule replacing equations (15) will reduce the number of training iterations by increasing the rate of convergence of

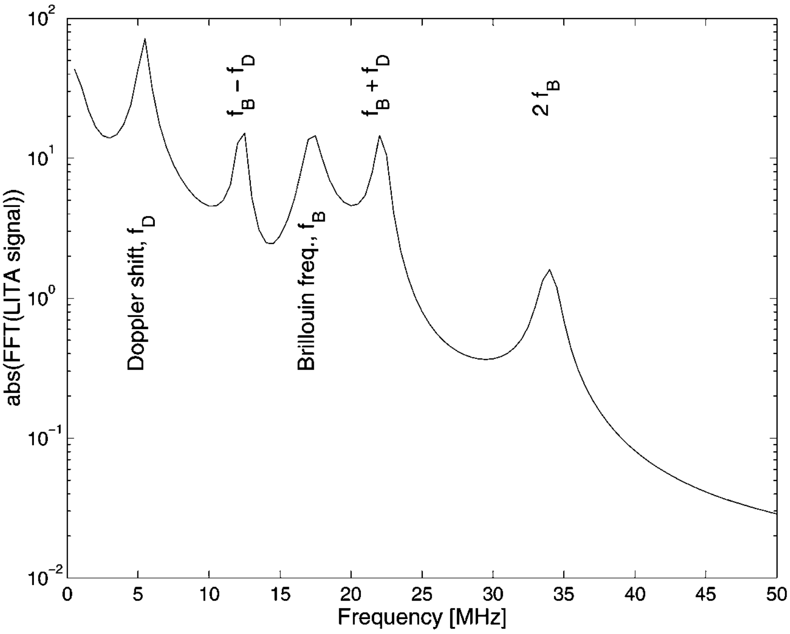

Figure 11. Frequency decomposition of the LITA signal from figure 2.

E. The fact that the errors for low SNRs and very low flow velocities are nearly independent of the number of hidden units suggests that we face a theoretical limit that we cannot overcome by increasing h.

The error for the flow velocity is fairly constant over the range of u_y. This means that the percentage errors are large at low flow velocities. This, however, is not due to the neural network but is governed by the Fourier limit which no data analysis method can escape. We showed that the neural network is robust with respect to noise. The performance worsens gradually in the presence of noise in the data. The phase between the Brillouin frequency and the Doppler shift $\tilde{\phi}$ has, as required, no influence on the data analysis.

This indicates that, internally, the network performs a frequency decomposition. It not only looks for the location of the peaks in the spectrum but uses all available information in the processing. This would be equivalent to applying the fitting technique to the FFT of the experimental data. Figure 11 shows the frequency decomposition of the signal plotted in figure 3. We see that, besides the two fundamental frequencies, the spectrum also contains some mixtures and harmonics of those two. For different flow velocities, the arrangement of the peaks will vary. Without prior knowledge it is non-trivial to determine which peaks correspond to the Brillouin frequency and the Doppler shift. The neural network is apparently capable of learning this task. Furthermore, note that, in the case of figure 11 ($u_y = 100$ m s^{-1}), the peak for the Doppler shift is at 5.5 MHz. The resolution of the spectrum is 0.5 MHz which corresponds to flow or sound speeds of 10 m s^{-1}. This uncertainty does not include the effect that nearby peaks might have for different values of u_y and the effect of noise. If we use the peak at twice the Brillouin frequency for determination of the speed of sound, the uncertainty is halved. Similarly, by using the distance between the peaks corresponding to $f_B - f_D$ and $f_B + f_D$ in the spectrum (the second and fourth peaks in figure 11), we can halve the uncertainty for the flow velocity. Using the same input data as those for the neural network, the broken lines in figure 9 mark the average uncertainty levels obtained using such a FFT peak detection scheme. The neural network performance with $h = 50$ is significantly better than that of the frequency decomposition technique.

Once the proper weights w_{ji} and v_{kj} have been found in the training phase, the neural network scheme is computationally very cheap. It requires only approximately $\mathcal{O}(n \times h + h \times m)$ operations to obtain a parameter estimate from a given input. The Levenberg–Marquardt scheme, in comparison, requires the inversion of the Hessian matrix in every iteration ($\mathcal{O}(n^3)$) in addition to other calculations. Besides being computationally expensive, it also tends to be unstable if the Hessian matrix is near singular.

We conclude that the accuracy of the neural network method presented lies in between those of the pure frequency decomposition technique and the nonlinear fitting technique. The computational cost is comparable to that of the fast frequency decomposition technique. An additional advantage of the neural network technique is its robustness. In particular, the Levenberg–Marquardt fitting scheme described in [9] is numerically very unstable. Lacking good initial estimates for the fitting parameters, it often does not converge to the correct solution. A combination of the neural network technique with the Levenberg–Marquardt scheme, whereby the neural network outputs are used as initial guesses for the Levenberg–Marquardt scheme, could be used if very accurate results are required. With good initial guesses, the Levenberg–Marquardt scheme will be more stable and converge faster. Although the neural network was used only to extract c_s and u_y, additional units in the output layer could be added, e.g. to extract the thermal diffusivity or the phase shift.

It must be pointed out that the training takes a considerable amount of time. This, however, can be done in advance. In the actual experiment the data analysis can then be performed in real time at a rate of thousands of signals per second, allowing the possibility of real-time data analysis even for multi-point measurements. Currently, the LITA data analysis requires user expertise and input which is unacceptable for a user-friendly and packaged LITA system. Either on its own or in combination with the Levenberg–Marquardt algorithm, the neural network approach can provide significant advantages for this application.

Acknowledgments

Sam Roweis (Gatsby Computational Neuroscience Unit, University College, London) and Erik Winfree (California Institute of Technology) provided the core source code for the neural network implementation. This work was supported by Advanced Projects Research, Inc and by NASA Langley Research Center under NASA contract NAS1-99016.

References

[1] Butenhoff T J 1995 Measurement of the thermal-diffusivity and speed of sound of hydrothermal solutions via the laser-induced grating technique *Int. J. Thermophys.* **16** 1–9

[2] Stampanoni-Panariello A, Hemmerling B and Hubschmid W 1998 Temperature measurements in gases using laser-induced electrostrictive gratings *Appl. Phys.* B **67** 125–30

[3] Walker D J W, Williams R B and Ewart P 1998 Thermal grating velocimetry *Opt. Lett.* **23** 1316–18

[4] Schlamp S, Cummings E B and Sobota Th H 2000 LITA velocimetry using heterodyne detection *Opt. Lett.* **25** 224–6

[5] Schlamp S and Allen-Bradley E 2000 Homodyne detection laser-induced thermal acoustics velocimetry AIAA paper 2000-0376

[6] Brown M S and Roberts W L 1999 Single-point thermometry in high-pressure, sooting, premixed combustion environments *J. Propulsion Power* **15** 119–27

[7] Hart R C, Balla R J and Herring G C 1999 Nonresonant referenced laser-induced thermal acoustics thermometry in air *Appl. Opt.* **38** 577–84

[8] Baker P F, Grinstead J H and Miles R B 1999 Single-pulse temperature measurement in supersonic air flow with predissociated laser-induced thermal gratings *Opt. Commun.* **168** 177–82

[9] Cummings E B, Leyva I A and Hornung H G 1995 Laser-induced thermal acoustics (LITA) signals from finite beams *Appl. Opt.* **34** 3290–302

[10] Schlamp S, Cummings E B and Hornung H G 1999 Beam misalignments and fluid velocities in laser-induced thermal acoustics (LITA) *Appl. Opt.* **38** 5724–33

[11] Marple S L 1987 *Digital Spectral Analysis with Applications* (Englewood Cliffs, NJ: Prentice-Hall)

[12] Scott D M and Waterland R L 1995 Identification of plastic waste using spectroscopy and neural networks *Polym. Eng. Sci.* **35** 1011–15

[13] Balfoort H W *et al* 1992 Automatic identification of algae—neural network analysis of flow cytometric data *J. Plankton Res.* **14** 575–89

[14] Safaric R, Jezernik K and Pec M 1998 Neural network control for direct-drive robot mechanisms *Engng Applications Artificial Intelligence* **11** 735–45

[15] Woon L S *et al* 1999 Adaptive neural network control of coordinated manipulators *J. Robotic Systems* **16** 195–211

[16] Linggard R, Myers D J and Nightingale C (eds) 1992 *Neural Networks for Vision, Speech, and Natural Language* (London: Chapman & Hall)

[17] Giles C L and Gori M (eds) 1998 *Adaptive Processing of Sequences and Data Structures: International Summer School on Neural Networks* (Berlin: Springer)

[18] Leondes C T (ed) 1998 *Neural Network Systems, Techniques, and Applications* (San Diego, CA: Academic)

[19] Hertz J, Krogh A and Palmer R G 1991 *Introduction to the Theory of Neural Computation* (New York: Addison-Wesley)

[20] Bishop C M 1995 *Neural Networks for Pattern Recognition* (Oxford: Oxford University Press)

[21] Rosenblatt F 1962 *Principles of Neurodynamics* (New York: Spartan)

[22] Rumelhart D E, Hinton G E and Williams R J 1986 Learning representations by back-propagating errors *Nature* **323** 533–6

[23] Bryson A E and Ho Y C 1969 *Applied Optimal Control* (New York: Blaisdell)

[24] Werbos P J 1974 Beyond regression: new tools for prediction and analysis in the behavioral sciences *PhD Thesis* Harvard University

[25] Werbos P J 1994 *The Roots of Backpropagation* (New York: Wiley)

[26] Parker D B 1985 Learning logic *Technical report* TR-47 Center for Computational Research in Economics and Management Science, Massachusetts Institute of Technology, Cambridge, MA

[27] Robbins H and Monro S 1951 A stochastic approximation method *Ann. Math. Statist.* **22** 400–7

[28] Masters T 1993 *Practical Neural Network Recipes in C++* (London: Academic)

[29] Haykin S 1999 *Neural Networks—A Comprehensive Foundation* (Upper Saddle River: Prentice-Hall)

[30] Cater J P 1987 Successfully using peak learning rates of 10 (and greater) in back-propagation networks with the heuristic learning algorithm *IEEE 1st Int. Conf. on Neural Networks (San Diego 1987)* vol II, ed M Caudill and C Butler (New York: IEEE) pp 645–51

[31] Franzini M A 1987 Speech recognition with back propagation *Proc. 9th Annual Conf. of the IEEE Engineering in Medicine and Biology Society (Boston 1987)* (New York: IEEE) pp 1702–3

[32] Vogl T P, Mangis J K, Rigler A K, Zink W T and Alkon D L 1988 Accelerating the convergence of the back-propagation method *Biol. Cybernetics* **59** 257–63

[33] Jacobs R A 1988 Increased rates of convergence through learning rate adaptation *Neural Networks* **1** 295–307

[34] Cybenko G 1989 Approximation by superpositions of a sigmoidal function *Math. Control Signals Systems* **2** 337–41

[35] Hornik K, Stinchcombe M and White H 1989 Multilayer feedforward networks are universal approximators *Neural Networks* **2** 359–66