Remote Sensing Operations

Fouad Badran*, Carlos Mejia, Sylvie Thiria* ** and Michel Crepon****

* CEDRIC, Conservatoire National des Arts et Métiers

292 rue Saint Martin - 75141 PARIS CEDEX 03 FRANCE

** Laboratoire d'Oceanographie et de Climatologie,Université de Paris 6, PARIS, FRANCE

1.Introduction

The term "Remote Sensing" refers to the measuring of physical parameters from a distance. Transfer functions are mainly used to relate measured quantities to significant physical parameters under study. In many cases transfer functions cannot be determined from theoretical considerations and have to be estimated from a data set. As an example oceanographers and meteorologists expect to measure the sea surface wind by using spaceborne radar. Theory predicts some relationship between the backscatter signal of the radar and the wind vector but the complete problem is too complicated to solve.

Neural Networks (NN) offer interesting possibilities for solving problems involving transfer functions. NN are adaptative, thus providing an easy way of modelling a large variety of physical phenomena [Thiria, 93].

In what follows, attention is focused on the retrieval of wind vectors from scatterometer measures by the use of Neural Networks. The method can easily be extended to a large class of problems involving the computation of empirical transfer functions. The matter is of interest since the European satellite ERS1 with the AMI (Advanced Microwave Imager which can function in scatterometer mode) was launched in 1991 and the American scatterometer NSCAT (NASA SCATterometer) should be launched in 1996.

2.The Geophysical model

The wind and the latent and sensible heat fluxes are the major forcing of the ocean circulation. As the spatial distribution of these parameters plays an essential role in driving the large scale circulation, a good knowledge of their space variability is of fundamental importance to understand and model ocean circulation. Scatterometers will provide measurements of the wind vector on a grid mesh of 50×50 km with a time and space coverage dramatically improved as compared to conventional means of observation. For the first time oceanographers can expect to obtain an adequate description of the forcing of ocean circulation which mainly depends on the wind-stress vector for the surface layers and on the wind-stress curl for the large scale and deep motions.

Scatterometers are active microwave radar which accurately measure the ratio of transmitted versus backscattered power signal, a ratio usually called the normalized radar cross section (σ^0) of the ocean surface.The physics of the interaction of the radar beam with a rough sea surface is poorly understood. Such effects as wave breaking, modulation by long waves, and the effects of rain, make the problem complex.

In fact empirical models have been proposed by geophysicists: they result from experimental research which has shown the strong dependence existing at the ocean surface between the σ^0 and the wind. An empirical relation can be determined by a functional equation of the form (Figure 1):

$$\sigma^0 = G(\chi,\theta,\varepsilon,v)$$

where σ^0 is expressed in dB and represents the ratio between the radar signal emitted and received .

G is a function of:

- the angle of incidence θ, measured on the vertical plane, between the incident radiation and the normal to the sea surface,
- the azimuth angle χ, measured on the horizontal plane, between the antenna and the wind direction,
- the radiation polarization ε, and
- the wind speed *v*.

Since it is found that the σ^0 varies harmonically with χ, it is possible to compute the wind direction by using several antennas pointing in different orientations with respect to the satellite track (Price, 1976 - Freilich and Chelton, 1986).

A model developed by A. Long [Long, 1986] gives an expression for σ^0 which is approximated by a Fourier series of the form:

$$\sigma^0 = U \cdot \left(\frac{1 + b_1 \cdot \cos(\chi) + b_2 \cdot \cos(2\chi)}{1 + b_1 + b_2} \right)$$

with $U = A \cdot v^{\gamma}$

The parameters A and γ only depend on the incidence angle θ and b_1 and b_2 are a function of both the wind speed v and the incidence angle θ.The different parameters used in this model are determined experimentally.

The computation of the wind vector requires the inversion of the above formula.

In the absence of noise introduced by the sensor or by the geophysical model, the three σ_0 given by the three antennas $(\sigma_1, \sigma_2, \sigma_3)$ at a given point, enable the computation of the wind speed and the wind direction in a unique way, except at certain peculiar points where ambiguities exist [Fig. 2(a)].

In the σ^0 space, the graph of the above function [Fig. 2(a)] is a triple cone-like surface with singularities corresponding to ambiguities in the wind direction (Cavanié and Offiler, 1988 - Roquet and Ratier, 1988). The directrix is a Lissajous curve [Fig. 2(b)] which is a function of the wind direction whereas a coordinate along the generatrix is a function of the wind speed. At a constant wind speed, the Lissajous curve implies that two directions differing by 150° are possible for some $(\sigma_1, \sigma_2, \sigma_3)$ measurements [Fig. 2(b)].

The problem is therefore how to retrieve wind vectors using the observed measurements $(\sigma_1, \sigma_2, \sigma_3)$.

The determination of the wind vector may be decomposed into two different problems which are not of the same order of difficulty. Due to the Lissajous ambiguities, computing the wind speed is easier than computing the wind direction. Thus the whole problem can be decomposed into two sub-problems leading to the determination of two distinct transfer functions. The first one is a singlevalued function which permits the computation of the wind speed while the wind direction determination is a multivalued function. The aim of this preliminary study is to prove that neural networks can efficiently solve such problems.

Before real data were available, we tested the method on simulated data computed from meteorological models. The swaths of the scatterometer ERS1 were created by simulating a satellite flying on wind fields (Figure 1). The backscatter values $(\sigma_1, \sigma_2, \sigma_3)$ given by the three antennas were calculated using Long's model [Long, 86]. Noise was then added to the σ's in order to simulate the errors made by the scatterometer (a gaussian noise with zero average and of standard deviation 9.5% for both lateral antennas and 8.7% for the central antenna was added to each measurement). As the actual noise of the ERS1 scatterometer is about 5%, our simulations have been run in a very noisy environment. Adding noise to the calculated data increases the ambiguities, given that a point $(\sigma_1, \sigma_2, \sigma_3)$ may lie between two Lissajous branches (Figure 2).

The data set was extracted from 22 simulated maps representing 22 days of satellite observation over the southern Atlantic ocean, regularly distributed over September 1986; 9 more days were used for test (5041 wind vectors).

3.Neural networks

In the following we use a MultiLayer Perceptron (MLP) [Rumelhart,86] trained using the backpropagation algorithm. It enables, in general, to learn a function f : $^n \rightarrow ^p$ using the association between m vectors ($\mathbf{x}_1,\ldots,\mathbf{x}_m$) (the input data), each $\mathbf{x}_k$ in ^{n}, and m corresponding vector ($\mathbf{y}_1,\ldots,\mathbf{y}_m$) (the output data), each $\mathbf{y}_k$ in ^{p}.

A given MLP architecture defines a family $\boldsymbol{F}$ of highly non linear functions. The approximation of a multidimensional transfer function $\mathbf{T}$ ($^n \rightarrow$) with a MLP can be done by using specific architectures (e.g particular families $\boldsymbol{F}$) whose first level has n cells and the output layer one cell. Once the architecture is fixed, the best approximation of $\mathbf{T}$ (the Least Mean Square approximation) has to be found in the related family $\boldsymbol{F}$. The computed output values for input x are expected to approximate $\mathbf{T}(\mathbf{x})$.

By restricting $\mathbf{T}$ to be continuous on a compact subset, $\mathbf{T}$ can be approximately represented by a k-layer MLP ($k \geq 3$) in the sense of the uniform convergence [Cybenko, 1989]. Moreover, the set of k-layer MLP's provides a powerful basis of functions and theoretical results can often be experimentally extended to non-continuous mappings $\mathbf{T}$.

In this section we present a methodology based on the latter MLP properties in order to get additional information on the unknown transfer function. When T is a complex multi-valued transfer function, a network with just one output will not be able to produce the possible different values together with their likelihoods. However, if we discretized the range of values in bins, and assign one output unit per bin, a MLP will be able to produce a posteriori probabilities for each bin. Moreover, statistics on the outputs of such an MLP will discover the nature (single-valued or multi-valued) of the unknown transfer function.

Assume that **T** takes its values in [*a*, *b*], which is discretized in *N* intervals. The MLP is defined as before except for the output layer. This layer is made of *N* cells, where cell *i* stands for the values of **T**(**x**) belonging to:

$$I_i = \left[a + \frac{b-a}{N}(i-1),\ a + \frac{b-a}{N} i\right] \qquad 1 \le i \le N.$$

An input pattern **x** is associated with class *i* if **T**(**x**) belongs to I_i. Approximating **T**(**x**) is now equivalent to learning the association {(**x**,*i*), **x** in ^{n}}.

In order to provide a numerical value approximating **T**(**x**), we identify I_i by means of its centroid m_i

$$m_i = a + \frac{b-a}{N}\left(i - \frac{1}{2}\right).$$

We now have to define the desired output. For a given input **x**, the desired output vector **y** = ($y_1, y_2, \ldots, y_N$) is defined by y_i = +1 if the value of **T**(**x**) is in interval I_i and y_i = –1 if it is not.

For a given weight matrix **W**, the MLP is now a function:

F(·,**W**) = (F_1(·,**W**), ..., F_N(·,**W**)) from (^{n} → ^{N})

N being the number of intervals used to discretize **T**.

The MLP is now used as a classifier system. Theoretical results [see appendix] show that **F**(**x**,**W***) (**W*** are the optimum weights) gives the "best" approximation in the family $\boldsymbol{F}$, related to the MLP architecture, of Bayes discriminant functions. So the output values can be used as likelihood coefficients; this allows us to rank the multiple solutions. Each cell *i* of the

output layer associates the value m_i with the related coefficient of likelihood $F_i(\mathbf{x},\mathbf{W^*})$. Each input pattern **x** is thus associated to the output activation curve which is the graphical representation of the relation $(i, F_i(\mathbf{x},\mathbf{W^*}))$ (Figure 3). In this figure the *x*-axis represents the index *i* of the output cells and the *y*-axis the output values $F_i(\mathbf{x},\mathbf{W^*})$. A peak $(i, F_i(\mathbf{x},\mathbf{W^*}))$ of this curve is a local maximum which can be characterized by cells $i-1, i, i+1$. The peak *i* is thus associated to a coefficient of likelihood $F_i(\mathbf{x},\mathbf{W^*})$ and to a numerical value q_i which is the weighted average:

$$q_i = \frac{\sum_{j=i-1}^{i+1} F_j(\mathbf{x},\mathbf{W^*}) m_j}{\sum_{j=i-1}^{i+1} F_j(\mathbf{x},\mathbf{W^*})} \quad . \qquad (3)$$

For each pattern **x**, owing to the number of significant peaks, the MLP gives **L** possible values of $\mathbf{T}(\mathbf{x})$ which are the q_{i_k} (k= 1, .. **L**). These values correspond to the **L** peaks having the largest likelihood coefficients These peaks are ranked by decreasing order of $F_i(\boldsymbol{x},\boldsymbol{W^*})$ which correspond to a decreasing order of their conditional probabilities $p(I_{i_1} |\mathbf{x}) \geq \ldots \geq p(I_{i\mathbf{L}} |\mathbf{x})$.

Looking at the different curves obtained for each pattern of the learning set enables us to understand the nature of the function studied. Figure 3 gives two characteristic examples of the output curves obtained with the approximation of the two transfer functions we deal with: wind speed and wind direction as function of $(\sigma_1, \sigma_2, \sigma_3)$. These approximations were obtained by training MPL's (see details below). The nature of the phenomenon is related to the number of significant peaks. It clearly appears that the wind speed transfer function [Fig. 3(a)] is singlevalued (most of the related output curves exhibit a single peak). On the contrary the wind direction transfer function is multivalued [Fig. 3(b)]; the existence of several peaks demonstrates the existence of major ambiguities (two peaks or more).

In the following, the approach described above is called "classifier mode" (i.e. by a NN with N outputs), in opposition to the "real mode" which refers to the real regression of T(x) by a NN with one output.

4.Numerical results

First it is noted that the normalized radar cross section (σ^0) function depends strongly on the incidence angle. In fact, the points located on the same parallel to the satellite trajectory are associated with a specific model; different parallels lead to similar models which only differ from the values of the parameters (b_1, b_2, A, γ). Without any loss of generality we chose to study the central trajectory and the two associated transfer functions

$$\mathbf{T_1} : (\sigma_1, \sigma_2, \sigma_3) \rightarrow (v) \text{ and } \mathbf{T_2} : (\sigma_1, \sigma_2, \sigma_3) \rightarrow (\chi)$$

Owing to the space variability of the wind, the measurements at a given point are closely related to those occurring in the neighborhood. Thus the input layer of each MLP consists of 9×3 neurons (σ^0 values relative to the 3 antenna) representing a 3×3 space window related to the central trajectory (Fig. 4). Each architecture has two hidden layers of 25 units, each layer is fully connected to the next. In the "real mode" a single output computes a single value for $\mathbf{T}(\mathbf{x})$, in the "classifier mode" there are 16 outputs (N =16 from 4 to 20 m/s). Tables 1 gives the performances on test set with the accuracy. For the two experiments (real and classifier mode) $\mathbf{T_1}$ is approximated with a good accuracy. However using the "classifier mode" gives a useful insight into function $\mathbf{T_1}$. The output curves exhibit for each input a single significant peak showing that $\mathbf{T_1}$ is single-valued Figure 3a). The same experiments using the same learning data set and the same test set were performed without spatial context. So the input layer has only three inputs coresponding to $(\sigma_1, \sigma_2, \sigma_3)$ and the performances related in the following on the learning set and the test set were decreased by 17%.

Similar experiments were performed to determine $\mathbf{T_2}$. The accuracy of the results is now given in degrees (±20° ,±10°).

At a given wind speed the wind direction is given by a Lissajous curve. In order to be correctly positioned along the curve presented (Figure 2), the wind speed computed above is added onto the input layer as a supplementary information. This new information improves the accuracy of the solution. Thus the input layer of each MLP is now composed of 30 neurons (9×3 correspond to the σ_o values, the last three correspond to the computed wind speed). All the architectures are fully connected and have two layers of 25 units; in the "real mode" they have two outputs which compute the sine and cosine of the desired direction, and in the "classifier

mode" 36 outputs (N = 36, from 0° to 360°). The results are displayed in Table 3 . Table 3 shows that, with the required accuracy of ±20°, two peaks help to recover $\mathbf{T_2}(\mathbf{x})$ with 98%. To select the final wind direction we apply an additional level (ambiguity removal) which is not presented in this paper [Badran & al., 1991].

In order to prove the power of the NN approach, Table 4 compares our final results with those of six classical methods [Chi & Li 88]. It shows that the NN results are very good compared to other techniques, moreover all the classical methods are based on the assumption that a precise analytical function ($(\nu,\chi) \rightarrow \sigma$) exists, the NN method is more general and does not rely on such an assumption.

5.Conclusion

The aim of the present paper was to show that Neural Networks (NN) can model a large class of complex transfer functions and that the nature of the transfer function (singlevalued or multivalued) can be deduced from the numerical results provided by the MLP. For the sake of clarity, we have chosen to deal with a particular example, the decoding of satellite scatterometer data, which highlights the different possibilities of NN. The transfer function from the radar measurements to the wind vector is a very complex multivalued function, therefore several solutions have been obtained with its related likelihood coefficient. The first results obtained using the real data provided by the ERS1 scatterometer confirm the adequacy of the neural approach [Mejia, 94].

Moreover the determination of the nature of the transfer function we present is related to conditional density approximation $p(\mathbf{y} / \mathbf{x})$. The classifier mode can be easily adapted to directely approximate this unknown conditional density. Recent results [Bishop, 94] present a general framework for this problem.

6.References

F. Badran, S. Thiria, and M. Crepon 1991, "Wind ambiguity removal by the use of neural network techniques," *J. Geophys. Res.* **96**(C11), 20 521-20 529.

C. M. Bishop 1994, "Mixture density networks," Neural Computing Research Group Report, NCRG/4288 (Departement of Computer Science, Aston University, Birmingham, UK).

C. Y. Chi and F.K. Li 1969, "A Comparative Study of Several Wind Estimation Algorithms for Spacebornes scatterometers," *IEEE Transactions on Geoscience and Remote Sensing* **26**(2), 19-88.

G. Cybenko 1989, "Approximation by superposition of sigmoïdal function," *Mathematical control of signals and systems*, Vol 2, pp 303-314.

M. H. Freilich and D. B. Chelton 1986, "Wave number spectra of Pacific winds measured by the Seasat scatterometer," *J. Phys. Oceanogr.*, **16**, 741-757.

A. E. Long 1986, "Towards a C-Band Radar Sea Echo Model for the ERS1 Scatterometer," *Proc. First Int. Coll. on Spectral Signatures* (ESA, Les Arcs).

C. Mejia, S. Thiria, F. Badran and M. Crepon 1994, "Determination of the geophysical model function of ERS1 Scatterometer by the use of neural networks," in *IEEE Ocean94 Proceedings* **1**, Brest-Sept. 13-16.

J. C. Price 1976, "The nature of Multiple Solution of Surface Wind Speed Over the Oceans from Scatterometer Measurement," *Remote Sensing of Environement*, **5**, 47-74 (American Elsevier Publishing Compagny).

D. M. Richard and R. P. Lipmann 1991, "Neural Network classifiers estimate Bayesian *a posteriori* probabilities," *Neural computation* **3**(4) 461-483.

D. E. Rumelhart, J.L. McClelland and the PDP Research Group 1986, *Parallel Distributed Processing: explorations in the microstructures of cognition* (MIT Press).

S. Thiria, C. Mejia, F. Badran and M. Crepon 1993, "A neural network approach for modeling nonlinear transfert functions: application for wind retrival from spaceborne scatterometer data," *J. of Geophys. Res.* **98**(C12), 22 827-22 841.

Appendix

CLASSIFIER MODE AND BAYES DISCRIMINANT FUNCTIONS

QMN has been widely used as classifiers. The classical approach is to use Bayes discriminant functions. The Bayes discriminant functions guarantee the minimum error risk of classification. These two approaches are closely related as described here .

The Bayes theory assumes an a-priori knowledge of the probability distribution of the data. Usually the density function is chosen in a given family. In order to give some theoretical insight on the way QMN and Bayes classifier behave we consider a N classes classification problem. The following notations are used :

$C_1, C_2, \ldots, C_N$ are the classes to be learned.

X is the learning set.

x is an element of **X**.

$P(\mathbf{x})$ the density function of **x**.

$p(C_i \mid \mathrm{x})$ the probability of the class C_i given x.

A Bayes decision rule is defined by :

$$\text{decide } C_i \text{ if } p(C_i \mid \boldsymbol{x}) = \sup_j \left(p(C_j \mid \boldsymbol{x}) \right) \quad j \in [1 \ldots N] .$$

Considering the function $g = (g_1, g_2, \ldots, g_N)$ called a Bayes discriminant function, where

$$g_i(\mathrm{x}) = p(C_i \mid \mathbf{x}) - \sum_{i \neq j} p(C_i \mid \mathbf{x}) ;$$

the above rule can be rewritten

$$\textit{decide } C_i \textit{ if } g_i(x) = \sup_j \left(g_j(x) \right) \quad j \in [1 \ldots N] .$$

Let us now focus on the QMN approach. In order to built the QMN classifier one have to defined the desired outputs of the network, that is the coding of the different classes. In most cases the desired outputs are the index of the classes : for a given pattern **x** of class C_i, the desired output is thus $\mathbf{y} = (y_1, \ldots, y_N)$ with $y_i = +1$ for C_i and $y_j = -1$ else.

It can be shown [Gish, 1990; White, 1989] that, given a particular architecture $\mathbf{F}(\cdot,\mathbf{W}) = (f_1, f_2, \ldots, f_N)$ and a large learning set **X**:

$$C(\boldsymbol{W}) = \sum_{\boldsymbol{x} \in \boldsymbol{X}} \left| \left| \boldsymbol{F}(\boldsymbol{x},\boldsymbol{W}) - \boldsymbol{y} \right| \right|^2 \approx \sum_k \int \left[f_k(\boldsymbol{x}) - g_k(\boldsymbol{x}) \right]^2 P(\boldsymbol{x}) \, d\boldsymbol{x} + \text{Constant}$$

Thus the back-propagation algorithm which minimizes $C(\mathbf{W})$ gives a function

$\mathbf{F}(\cdot,\mathbf{W}) = (f_1, f_2, \ldots, f_N)$ which approximates the Bayes discriminant functions $(g_1, g_2, \ldots, g_N)$. The approximation is in the sense of the probability weighted least sqares of the above formula.

The states of the output units $f_k(\mathbf{x})$ are thus linked to the *a posteriori* probability $P(C_k \mid \mathbf{x})$. It is straightforward to show that $f_k(\mathbf{x})$ can also be interpreted as coefficients of likelihood. The accuracy of the approximation depends on the complexity of the *a posteriori* probability function (Annex A), the architecture of the QMN and on the learning set $\mathbf{X}$ [Geman, 1992; Vapnik, 1982; White, 1989].

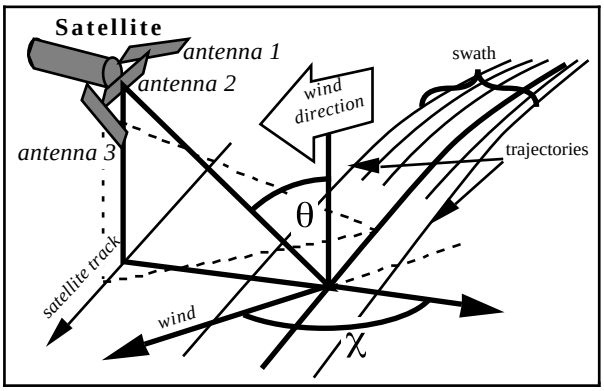


Figure 1: Definition of the different geophysical scales.

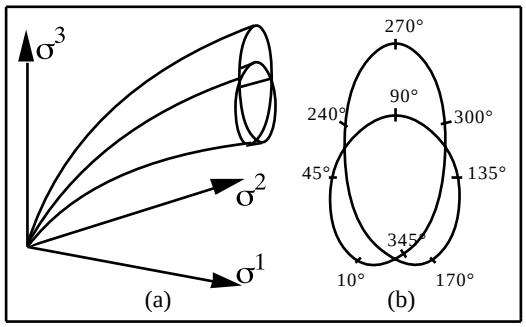
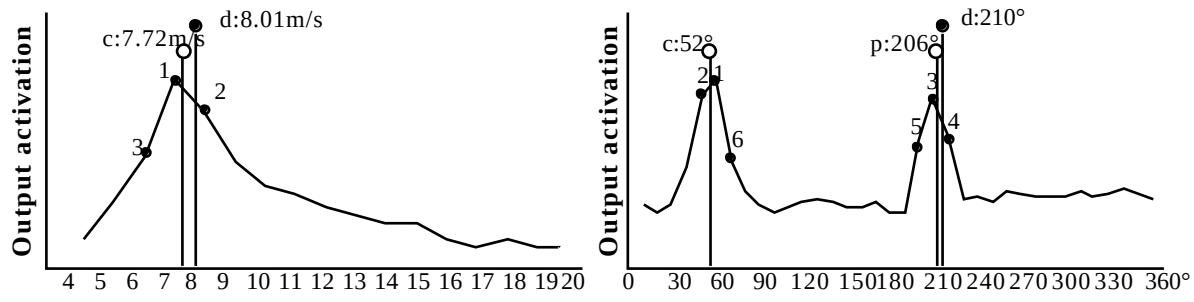


Figure 2: (a) Functional for computing the wind speed and direction as a function of the σ^0 measured by the antennae; (b) Lissajous curve obtained for a section at constant wind speed. The points of a section of the surface are related to the same to the same wind speed for all possible directions of the wind.



a *b*

Figure 3: Characteristic examples of output curves: (a) speed; (b) direction. The letter d represents the real solution on the x axis , c is the first significant peak computed by using formula 3, p the second peak

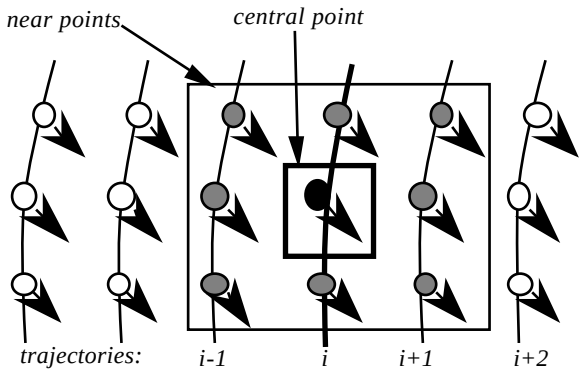


Figure 4: 3×3 space window related to the trajectory i .

	accuracy	*Real mode*	*classifier mode*
±1m/s	PERF	97.4 %	91.3%
± 2 m/s	PERF	99.8 %	97.3%
	RMS	0.29m/s	0.42 m/s
	BIAS	0.04 m/s	0.02 m/s

Table 1: performances on the test set for wind speed retrieval. RMS represent the root mean square error and BIAS the mean of the difference between the desired output and the computed output.

accuracy	learning	test
PERF ±10°	52.9%	51.4%
PERF ±20°	75.6%	74.0%
PERF ±30°	84.9%	82.4%
RMS	2.46 m/s	2.86 m/s

Table 2: wind direction retrieval for the real mode

accuracy	±10°-1p	±10°-2p	±20°-1p	±20°-2p	±20°-3p	±20°-4p
test	67.8%	91.9%	72%	98.4%	99.5%	99.9%
RMS.	4.75m/s	1.03m/s	4.75m/s	0.75m/s	0.61m/s	0.57m/s

Table 3: wind direction retrieval using the **classifier mode.** Each row displays the accuracy and the number of possible solutions allowed for $\mathbf{T_2(x)}$ (number of curve peaks).

Speed	WLSL	ML	LS	WLS	AWLS	L1	LWSS	N.N
Low	0.92	0.66	0.67	0.74	0.69	0.63	1.02	0.49
Middle	0.89	0.85	1.10	1.31	0.89	0.98	0.87	0.53
Height	3.71	3.44	4.11	5.52	3.52	4.06	3.49	1.18

Table 4: E_{rms} (in m/s) for different fixed wind speed. The wind vector error e is defined as follows: e = **V1** – **V2** where **V1** is the true wind vector and **V2** the estimated wind vector, $E_{rms} = E(||e||)$. Performances of the above method compared with those of seven other wind estimation algorithms proposed. All of them are based on weighted least squares minimization: (WLSL) is a weighted least squares algorithm which acts in the log domain, (LS) is the least squares algorithm, (WLS) the weighted least squares, (AWLS) the adjustable least squares, (L1) the L1 norm algorithm, (LWSS) the least wind speed squares algorithm and (ML) the maximum-likelihood algorithm. A description of these algorithms is given in Chong-Yung Chi and Fuk K. Li [Chi & Li, 1988].