A Virtual Instrument for Automatic Anemometer Calibration with ANN Based Supervision

F. López Peña & R. J. Duro
Escuela Politécnica Superior, Universidade da Coruña,
Mendizabal s.n., E 15403, Ferrol, SPAIN
Phone: +34 981337400, Fax: +34 981337410
Email: flop@udc.es

Abstract* – *A fully automatic anemometer calibrator intended for performing fast and accurate calibrations has been developed to fulfil the increasing demand and strict requirements of the wind energy industry. Different sensors are connected to a computer where a virtual environment acquires and processes the incoming signals and controls a wind tunnel, allowing the calibration of the anemometer at the pre-selected air speed values. An important part of the resulting complex virtual environment is a supervising system, based on artificial neural networks and able to check and handle the possible malfunctions and deviations within the calibration process.

Keywords* – *Virtual instrument, merging sensors, anemometer, calibration, wind tunnel, neural networks.

INTRODUCTION

During the last decade wind has become a source of energy that is increasingly tapped in many countries. The emergence of this modern industry has required a massive use of anemometers, first to carry out field studies on the optimum placement of the parks and towers, and afterwards to provide data to the control unit of the wind energy tower. There is no other application where the accuracy on wind speed measurement plays such an important role [1]. This very fact has made the quasi-nonexistent current anemometer calibration standards completely obsolete and induced an upsurge of research aimed at obtaining new calibration instruments and procedures [2][3].

In this line, a fully automatic anemometer calibrator intended for performing fast and accurate calibrations has been developed by our group in order to fulfill the increasing demand and strict requirements of the wind energy industry. The calibrator is structured by putting together a hardware

ensemble consisting of an instrumented wind tunnel wired to a PC based data acquisition system, and programmed virtual environment, which has been specifically developed to acquire and process the incoming signals, make the pertinent calculations, and control the wind tunnel power plant. In addition, to supervise the instrument operation we have resorted to neural networks that take into account dynamic aspects of the control and sensor signals [4] [5].

We came into considering the use of artificial neural networks after realizing that the supervision of the calibrator and error handling of the calibration process had become a complex non trivial task. The process supervision and error handling system is conceived to automatically reject possible undesired deviations in sensor values so that confidence in the automatic calibration process is preserved. In addition, it must provide malfunction signals that permit protecting the hardware.

HARDWARE AND SENSORS

A. The wind tunnel

The hardware setup core is a free jet wind tunnel (Fig.1). This wind tunnel has been specially designed and developed in house for anemometer calibration purposes; it produces a 300 mm wide jet with a velocity uniformity of better than 1% and a turbulence level of less than 1%. Its power plant consists of a centrifugal blower driven by an 11 kW asynchronous AC motor. The rotational speed of this AC motor is controlled by an electronic inverter able to work as a variable frequency power supply for the motor. The inverter-motor-blower ensemble is able to feed the free jet with a continuously variable air speed going from 0 to 45 m/s. The frequency supplied at the inverter power output is proportional to a 0-10 V DC input control signal. This signal is connected to one of the data acquisition board analog outputs, allowing the wind tunnel to be driven by the virtual instrument.

B. Sensing

The wind tunnel is instrumented by a set of sensors capable of performing measurements on the different physical quantities needed within the virtual instrument for the anemometer calibration process. Fig. 2 represents a simplified electrical connection scheme of this set-up. All electric and

electronic components are present in this figure except for the AC motor, the inverter and the computer. Following this scheme, the main airflow dynamic pressure is measured by means of a Pitot-static Prandtl type pressure probe by Dwyer, mod. 108019-00A, which is pneumatically connected to a differential pressure transducer Validyne model DP 15-3-20 with a CD 15 demodulator, which is adjusted to measure from 0 to 1000 Pa having a resolution of 0.2 Pa. To calculate the wind speed from the measured dynamic pressure, we must also know the density of air under test conditions. For this purpose we use three sensors measuring three different ambient quantities: pressure, temperature, and humidity. The barometric pressure sensor is a VAISALA mod. PTB100A, its accuracy is of 2 hPa within a range of 800-1060 hPa. The temperature sensor has been designed and implemented in our lab from a National Semiconductor IC temperature sensor model LM35A with a 10mV/ºC output in the 0-100 ºC range, and an amplifier based on the INA125 instrument operational amplifier, the final accuracy is of 0.2 ºC. The electronic hygrometer is a RENSE mod. HX-732-M-02, it can measure from 0 to 100% of RH with an accuracy of 2% RH. All of these sensors, as well as the anemometer to be calibrated, are connected to the data acquisition board installed in the computer trough a proprietary connection box. This box has some filters for the incoming signals as well as the power supplies for the different sensors used. The barometric pressure sensor and the power supply are physically placed inside the connection box. All sensors used have been chosen so as to produce an electrical voltage output proportional to the measured physical quantity.

The above mentioned transducers are periodically calibrated. These calibrations are performed in house and not on the single transducers but on the complete measurement chain taken as a whole, including the transducer itself, filters and signal conditioning, acquisition card, computer, and virtual instrument. In addition, prior to any anemometer calibration session, the differential pressure measuring block is calibrated in three points and the accuracy of the readings obtained for the three ambient quantities are verified by comparing them to the values obtained from reference instruments. If this preliminary test reveals that any of the verified quantities is out of specification, the corresponding measurement chain is calibrated before going ahead with the session.

A. The measuring subsystem

Both, the measuring and control subsystems are implemented within the virtual instrument developed using the National Instrument's LabVIEW data acquisition-programming environment. The measuring subsystem acquires and processes the signals coming from the anemometer and from all the sensors. It is able to handle different types of anemometer output signals, such as voltage, current or frequency signals. The system also permits calibrating anemometers with other types of outputs, but then the output values should be registered by hand. All the measuring procedures that follow are in accordance with the International Energy Agency recommendations [6].

As the wind speeds considered are in the low subsonic range, the air velocity v is calculated from the dynamic pressure Δp measured with the Pitot probe by using the incompressible flow relation:

$$v = \sqrt{\frac{2k_C \Delta p}{C_h \rho}} \qquad (1)$$

In this relation two empirical correction factors are included; k_C is derived from wind tunnel calibration results, it relates the Pitot probe position to the anemometer position; C_h is the head coefficient of the Pitot probe, for our probe C_h=1. Following the Measnet recommendations [7], the correction factor k_C was obtained by using a second Pitot probe at the anemometer location, a value of k_C=1.01 was achieved. The air density ρ is calculated from the measured ambient pressure P_a and temperature T_a as:

$$\rho = \frac{P_a k_\rho}{R_g T_a} \qquad (2)$$

Where R_g is the air gas constant and it has a value of R_g = 287.1 J/kg K. The factor k_ρ corrects the air density for relative humidity ϕ, it is obtained from the following semi-empirical relation:

$$k_\rho = \left(1 - 0.378\left(\frac{\phi P_w}{P_a}\right)\right) \qquad (3)$$

P_w is the vapor pressure at ambient temperature T_a, it was obtained by fitting the curves of a relative humidity chart by means of a third order polynomial:

$$P_w = 0.051T_a^{\,3} + 8.89T_a^{\,2} + 48.3T_a + 604 \qquad (4)$$

This set of four algebraic equations is solved inside the virtual instrument to calculate, in real time, the air speed velocity from the signals acquired by the ambient temperature, barometric pressure and differential pressure sensors, as well as by the electronic hygrometer. The measurement subsystem is set-up to perform the acquisition loop and make all calculations at a rate of 512 samples per second. The output value of the air speed V is the average of the values calculated during one second corrected for flow disturbances induced by the anemometer under calibration. Therefore, and according to expressions 1 and 2, this output value is:

$$V = k_f \frac{1}{512}\sum_{k=1}^{512} v_k = k_f \sum_{k=1}^{512}\sqrt{\frac{2k_c \Delta p_k R_g T_{ak}}{C_h P_{ak} k_\rho}} \qquad (5)$$

Where k_f is a global correction factor taking into account that the flow disturbances induced by the anemometer in the bounded air stream generated by the calibration wind tunnel differ from those produced in open air. The value of this factor depends on anemometer size and geometry; it is obtained for each type of anemometer by cross checking with a calibration made in a large wind tunnel available in our lab. For several different types of anemometers calibrated to this date, the value of this factor ranged from 0.98 to 1.

The anemometer's significant output may be a voltage, intensity, or a periodic signal. When measuring voltage or intensity, only standard conditioning signals are used at the input of this particular data acquisition channel, and then the system reads the signal directly. Most often the anemometer signals are periodic -sinusoidal or TTL types are quite common- being their frequency the significant measuring parameter. In these cases the system measures the frequency by performing a FFT over the incoming signal. For this purpose, a 2 second wide Hanning type window is applied in order to minimize the effects of a limited acquisition time and to be able to measure frequencies as low as 3 Hz. As the acquisition process is set to acquire data in blocks of 1 second, two consecutive blocks are used to obtain a two second wide window and thus achieve a 0.5 Hz resolution. After this, the power spectrum is calculated, and finally the frequency is obtained from the position of the maximum

of the power spectrum and then correcting its value by performing a weighted average with the frequency values of its two adjacent lines in the spectrum; this is a standard procedure to improve the initial resolution given by the FFT. If the anemometers' signals were pure, it should not be necessary to process them in the frequency domain, as different interpolation and/or zero crossing algorithms could achieve better results in the time domain. However, quite often these signals are contaminated by noise and, in some cases, vibrations induced by wind or other mechanical sources can modulate the periodic output signal while it is generated. Thus, the FFT was our choice as it will provide good results for any type of anemometer under calibration regardless of the appearance of this sort of disturbances.

B. Uncertainties

The uncertainty principles outlined in ISO [8] and NIST [9] guidelines reveal that in the measurement of airspeed we have two types of uncertainties: type A evaluation of standard uncertainty and type B evaluation of standard uncertainty. The total combined uncertainty can be evaluated as:

$$u_c^2(V) = u_{c,B}^2(V) + s_A^2(V) \qquad (6)$$

The term coming from type A evaluation of standard uncertainty, s_A, relates to the uncertainty attached to the mean value of a sampled process. It can be evaluated as:

$$s^2{}_A(V) = \frac{1}{n}\frac{1}{n-1}\sum_{k=1}^{n}\left(V_k - \overline{V}\right)^2 \quad (7)$$

Where n is the number of samples, typically 30, used to average the air speed values V_k to achieve its final validated value $\overline{V}$. As mentioned in the previous section, each of these V_k is obtained as an average, during one second, of a higher frequency scan process (512 Hz). We are considering now a second averaging which is performed by the system during n seconds after a given stabilization time. This averaging in conjunction with the high stability of the system ensures a reduction of the noise and makes the term obtained by type A evaluation of standard uncertainty negligible as compared to the one by type B evaluation of standard uncertainty.

Each of the coefficients used for calculating the air velocity have an associated uncertainty. All these uncertainties will contribute to the velocity's uncertainty evaluated as from type B. Assuming that all

different error sources are uncorrelated, the expression that relates the combined uncertainty of category B is reduced to:

$$u_{c,B}^{\ 2} = \sum c_i^{\ 2} u^2(x_i) = c_f^{\ 2} u_f^{\ 2} + c_c^{\ 2} u_c^{\ 2} + c_{\Delta p}^{\ 2} u_{\Delta p}^{\ 2} + c_{Ta}^{\ 2} u_{Ta}^{\ 2} + c_{Pa}^{\ 2} u_{Pa}^{\ 2} + c_{Ch}^{\ 2} u_{Ch}^{\ 2} + c_\rho^{\ 2} u_\rho^{\ 2} \quad (8)$$

Where the c_i are the influence coefficients ($c_i=\partial V/\partial x_i$), the x_i represent the different parameters within the V expression (5), and the $u(x_i)$ are the standard uncertainties of these parameters. As it was mentioned earlier, in the worst case the value of the k_f correction factor is k_f=0.98. Statistical analysis of the data from which the value was obtained shows that its standard uncertainty is u_f=0.002. For the correction factor k_c a standard uncertainty of half the difference between the corrected and uncorrected value is taken, thus u_c=0.005. The verification of the differential pressure and of the ambient pressure and temperature made before any calibration session leads to error values of 0.5 Pa, 0.4 mm Hg, and 0.2 K respectively. Assuming a rectangular distribution of errors, the corresponding standard uncertainties are of $u_{\Delta p}$=0.29 Pa, u_{Pa}=0.25 mm Hg, and u_{Ta}=0.12 K. With these values, and after deriving the influence coefficients, the uncertainty finally achieved from expression 8 is displayed versus air speed in Figure 3.

Notice that the assumption of non correlation of error sources holds in all cases but for the k_ρ coefficient, which correlates with temperature T_a as it results from equations 3 and 4. Nevertheless, as their corresponding influence coefficients are of opposite sign, taking this assumption will only result in a slight overestimation of the overall uncertainty.

C. The control subsystem

The control system is built around a PID controller using the calculated airflow velocity as control parameter. There was not enough information to model the complex blower-AC motor-inverter power plant, therefore we modeled it by measuring and analyzing its response under different input signals. The analysis of the amplitude, phase and coherence of the responses to step and chirp signals revealed that the inverter-motor-blower plant behaves as a high order system but with a dominant second order transfer function. Starting from the general form of this type of functions, the transfer function was finally estimated by analyzing the system response under small step input signals. The PID controller was built over the calculated transfer function and was later adjusted by using a Ziegler-Nichols tuning

method as described in [10]. After its implementation some conflicts were found between the actuation of this controller and the built-in one of the inverter. For large changes in wind speed, it was found that the programmable ramp for the charge output of the inverter interacts with the implemented controller producing an unwanted overshoot in the PID response. To avoid this trouble the control signal to the inverter was passed through a second ramp built in our controller and having a slightly smoother slope than the one of the inverter. In addition, for very small corrections of wind speed the two controllers arrived in some cases to an opposite phase oscillatory mode in such a way that the cross-induced oscillation reached a permanent state. To avoid of this trouble, once the desired value of the air speed was achieved within 0.5% range, our PID controller stopped, leaving the control to the inverter. In these situations the developed supervision system is supervising the inverter behavior and tells the control system to take over by activating the PID before any undesired deviation happens or, in other cases, informs that the conditions for correct measurements are achieved.

THE VIRTUAL INSTRUMENT

The main processes within the virtual instrument are organized as shown in the simplified flow chart of Fig. 4. The main loop is executed for each value of the reference velocity v_r stored in an array. The system acquires data continuously while the PID regulates the control signal of the inverter (*u control*) through an analog output channel of the acquisition board. Once the difference between the true and the required air speed is within a given tolerance margin the neural network produces a ready to measure signal and the PID stops to let the inverter's controller actuate over the AC motor speed. The system then checks that the measured air speed value stays within the tolerance margin during the required stabilizing and averaging time. If this is the case the anemometer output signal and the air speed values are averaged and stored, in any other case the PID is started and the process is repeated. The whole procedure is repeated for each calibration point until the last required value of the reference velocity v_r is achieved.

During calibration, the process can be stopped at any time by the user or by the system when any malfunction is detected. Possible errors or malfunctions are handled by two different error managing systems, one of these is rule based and the other is the neural network based system described later.

They supervise the possible overflow of any of the input channels as well as the correct response of the AC motor to the PID control signals, in addition they detect any problems related to the control of the system in terms of failures in connections, power turned off, by analyzing the time evolution of inputs and outputs of the control system.

The virtual instrument was designed bearing in mind that the final user should be able to perform fast and accurate anemometer calibrations with a minimum of knowledge and training. Therefore the instrument user interface is divided into several modules linked in a simple and intuitive pop-up menu.

ERROR SUPERVISION AND HANDLING.

In order to obtain a calibration system that can be run reliably by inexperienced users it was very important to include a flexible error supervision and handling subsystem in charge of monitoring the operation of the control system and sensors and providing information on whether the system was working correctly or there was some type of failure. The main objective of this subsystem was to prevent damages to the calibrator hardware and to ensure that the sequences of measurements obtained by the system verified the appropriate conditions of values and stability for the calibration points to be reliable.

The problem in designing such a system is to clearly define some elusive thresholds and patterns based on the temporal evolution of the signals provided by the sensors. These thresholds and patterns are related to the intervals of the variables for which calibrations are valid, the degree of stability of these signals, the patterns associated to different types of failures and distinguishing them from normality, etc.

In general, extracting knowledge from an expert on the final decision to be made is relatively easy and reliable. However, extracting from the expert the appropriate thresholds and time windows to encode this decision into a set of rules, in other words, to make the expert clearly and algorithmically explain how the decision is made, is much more difficult, especially when more than one signal is simultaneously involved in the decision process. This has led us to choose artificial neural networks as the supporting structure for the error supervision and handling subsystem and we have used it as an

expert to develop the final rule based system. Consequently, in this work we have tried to put together the best of both worlds. We make use of the ANN based supervisor as an expert modeler, this is, we model the expert decisions through an ANN which is later used to extract the rule based thresholds and temporal patterns required. In the final system we can run the ANN or the rule based system extracted from it, whatever is more convenient.

A. Approach

The task of the supervision module is twofold, on one hand, to determine that the system is stable in the state required for a calibration point to be taken and, on the other, to decide when a malfunction occurs that will render the data obtained unusable and, consequently, it is not worth taking a calibration point. Both tasks imply performing time related processing that involves determining, on one hand, the temporal window (contiguous set of time instants) or the set of non contiguous individual time instants we are going to consider and, on the other, the threshold values that delimit normality for all of the dimensions involved (number of time instants times the number of signals considered times the dimension of each signal). Thus, this detection must be carried out through comparison to a normality template that is related to the temporal evolution of the measurement values. As in most systems that work with real signals, normality is not represented by a well defined pattern, but rather by a set of, sometimes very complex, and fuzzy, boundary conditions in the dimensionality induced by the signal (in the sense of Takens and Mañe [11, 12]) that determine intervals where the signals are valid and intervals where they are not, and one must perform a rather complicated thresholding process when defining these high dimensional volumes.

In what follows we present results of using two types of ANNs to model the expert´s knowledge. In the first case, for simplicity and to establish a reference performance, we considered a traditional multilayer perceptron using the standard backpropagation algorithm for training. To model time dependent processes using the classical multilayer perceptron architecture it is necessary to provide the network with time windows of the input signals as parallel inputs, thus introducing two additional design decisions, namely, size of the window and points that make it up. In the second case, we tried to avoid these extra design decisions and developed an ANN architecture based on trainable delays

that is able to autonomously determine the window size and what points make it up straight from the input signal. In the results we will show that the second structure provides the same level of performance using a much simpler network in terms of nodes and connections and, what is more important, the designer does not need to define the input windows to the network.

In both cases we considered hand labeled operational charts as training and test sets. These charts include values for the reference air-speed, the measured air-speed and the inverter's control voltage during normal operation of the system. While these data were being recorded different malfunctions were forced on the system so that they would be reflected on the training and test sets at different points during the calibration procedure. Once these data were recorded, a human supervisor labeled by hand when it was considered that the system was stable enough to measure as the first output signal and when a malfunction occurred as the second one. These labels were taken as the target values for the network so that it would emulate the decision of the human.

B. Multilayer Perceptron ANN

In order to design the Multilayer Perceptron (MLP), as has already been commented, it is necessary to consider a temporal window for each one of the three signals we have available as inputs. It was empirically determined that a 21 instant long window of temporal values was required for each signal in order to be able to adequately model the human expert´s knowledge, leading to a huge network. Take into account that 21 instants of time using points from 3 signals leads to 63 inputs. To process these inputs we required two hidden layers with 50 nodes and the two outputs (one denoting if the signal is stable and measurements can be taken, the other one denoting there is a malfunction) leading to a grand total of 5750 weight values and very long training times so as to avoid local minima. To reduce the size of the networks using MLPs the only choice was to try to reduce the number of inputs so as to make processing easier. This again implies that the designer must somehow determine which points out of the 21 per signal in the window are really important and which may be ignored, either through a tedious testing process or by looking at how the previous network makes use of the inputs (what input weights are relevant and what input weights are almost zero). Using this procedure we reduced the number of inputs to one third of the original set. We have in fact selected an unevenly

sampled window of the values of the signals. The network employed was a two hidden layer multilayer perceptron with delayed inputs. The delays were 3, 6, 9, 12, 15, and 20 instants of time into the past and, consequently, we had six delayed inputs plus the current value per signal. This temporal window was appropriate for the system to make use of the sequence of data to determine if stabilization had been achieved and to rule out spurious values.

The final network consisted of 7 inputs and two 35 node hidden layers in addition to the 2 output nodes, leading to 1540 weights. After training on the operational charts commented above, it was inserted in the control structure of the calibration system and the results obtained demonstrated that it could determine with very good reliability when the system was ready to measure (through the consideration of the stability of the signals) and when there were problems that required readjusting the control parameters of the system or even shutting it down to prevent failures. In other words, the network had obtained a good definition of the high dimensional normality boundary for the signal. Figure 5 displays the results for a typical run (not seen in training) of the calibration procedure where some failures were forced. When the network's output-1 is high it is because the network has detected some type of failure and alerts the user that it is not safe to measure and that there may be a problem with the hardware. When the network's output-2 is high it indicates that the different voltages have been stable long enough for a calibration point to be measured unless output-1 says otherwise. During this run two hardware errors were forced, in the first case the Validyne differential pressure transducer was turned off and the network correctly detected the problem. In the second case the blower was turned off and the detection of this failure was also valid. In addition, whenever the set points for calibration were reached and maintained with enough stability for a sufficiently long period of time, the network indicated so consistently.

We must say that things work fairly well, but the network employed was handcrafted to fit this type of window and is obviously not very well suited for the description of more arbitrary time dependent phenomena unless a perfectly good window may be found and an adequate preprocessing algorithm that always selects this window for presentation to the network is implemented.

C. Trainable Delay Based ANN

The windowing approach presents many well known drawbacks, and to avoid them we require networks that are capable of carrying out temporal reasoning processes and which can learn the appropriate state space representation of the mechanisms involved straight from the signal, without any previous processing or windowing. We need the network to determine by itself what temporal instants are relevant for the decision in addition to determining the thresholds that define the normality region.

To address this problem we have considered a different type of network which consists of several layers of neurons connected as a Multiple Layer Perceptron (MLP). The only difference with traditional MLPs is that the synapses are represented by a delay term in addition to the classical weight term. That is, now the synaptic connections between neurons are described by a pair of values, (W,τ), where W is the weight describing the ability of the synapse to transmit information and τ is a delay, which in a certain sense provides an indication of the length of the synapse. The longer it is it will take more time for information to traverse it and reach the target neuron.

We have developed a training procedure for this type of networks, Discrete Time Backpropagation, a description of which may be found in [5]. This algorithm permits training the network through variations of synaptic delays and weights, in effect changing the length of the synapses and their transmission capacity in order to adapt to the problem in hand. From the point of view of time dependent signals, we obtain the window size and the points to be sampled within the window automatically through the training of the delay terms. In fact, the resulting network only has three inputs corresponding to the three signals as they come. It is through the delays in the different synaptic connections the network has learnt that it selects what values from which signals it will consider in order to make the final decision. Now the designer does not need to specify temporal windows, just what signals are going to be considered in the monitoring process. In addition, the resulting networks are much smaller (3 inputs, two 14 node hidden layers and 2 outputs, leading to 266 weight values). Figure 6 displays some results obtained with this network. It is quite evident that the results are comparable to those of the previous networks and they completely model the decisions of the expert. It is very important to note here that the expert provided information only on whether the system was stable and on the abnormal conditions that occurred. The expert never defined any thresholds or

temporal windows, the network obtained those in a completely autonomous manner. In fact, in some instances it generalized even better than the human expert, predicting breakdowns long before the human expert had enough data to become aware of them.

D. Rule-based supervision module

A rule-based system was constructed and set up in parallel with the last ANN supervision system. We started with a quite simple traditional ruled system and we kept enlarging it by analyzing the situations where this system was not providing an error action or warning message when the ANN based one was. In other words, we used the ANN based system as a model of the human expert to develop the rule based system. This way we have ended up with a robust traditional rule based error handling system that has been developed during normal operation of the calibrator without the risk of damaging the anemometer or the instrument, or obtaining mistakes in the calibrations as a consequence of an error or malfunction. When working in parallel both the ANN based and the traditional systems usually react simultaneously, but there are some situations were the ANN based system can predict the error before it happens anticipating the other one. Nevertheless, once the traditional system is completely set up it turns out to be more convenient when the calibration system is modified because in this case only some of its parameters need to be tuned while the ANN based one requires a new training process. In fact, one of the lines of research we are following is on how to obtain an adaptive ANN architecture that will be able to adapt rapidly to changes in the calibration system without having to completely retrain.

CONCLUSIONS

A virtual instrument to automatically perform fast and accurate anemometer calibrations has been developed. The instrument uses a set of different sensors and instrumentation, controls a specifically designed wind tunnel and performs all needed data processing and calculations to allow a non expert user to carry out the calibrations with remarkable accuracy. An artificial neural network based structure supervises operation and permits protecting the hardware and rejecting faulty measurements.

This neural network based approach has proven to be very reliable in the determination of malfunctions, often predicting them before the human expert or a traditional ruled system had become aware of a problem. It has been also used as a model of the human expert during all the trials carried out for the development of a ruled base system.

ACKNOWLEDGEMENTS

We are grateful for the collaboration of Texas Controls S. L. of Spain and for their support in the implementation of the system. This research was also partially funded by FEDER through project N. 1FD97-1700.

REFERENCES.

[1] A. Albers, H. Klug, "Wenn Prognose und Realität nicht übereinstimmen: Verifizierung". "Energieproduktion von Windparks". ERNEUERBARE ENERGIEN 8 (1998) 4, S. 34.

[2] T. Lockhart, "Relative uncertainty in wind tunnel calibrations". *Amer. Meteor. Soc.* Tenth Symposium on Meteor. Obs. and Instrumentation, 1998.

[3] National Engineering Laboratory, "Recommendations on the Use and Calibration of Cup Anemometers". *National Wind Turbine Centre*, 139/95, Scotland, 1995

[4] R.J. Duro and J. Santos, "Discrete Time Backpropagation for Training Synaptic Delay Based Artificial Neural Networks", *IEEE Transactions on Neural Networks,* Vol.10, No. 4, pp. 779-789, 1999.

[5] E.B. Kosmatopoulos, M.M. Polycarpou, M.A. Christodoulou, and P.A. Ioannou, "High-Order Neural Network Structures for Identification of Dynamical Systems", *IEEE Transactions on Neural Networks,* V. 6, N. 2, pp. 422-431, 1995.

[6] R. S. Hunter (Ed.). Wind Speed Measurement and Use of Cup Anemometry. In: Recommended Practices for Wind Turbine Testing, Issue 11. International Energy Agency, (IEA), 1999.

[7] Network of European Measuring Institutes, MEASNET. Cup Anemometer Calibration Procedure. Version 1, 1997.

[8] International Standards Organization, (ISO). Guide to the Expression of Uncertainty in Measurement. 1st Edition, 1992.

[9] National Institute of Standards and Technology, (NIST). Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results. NIST Technical note 1297, 1994.

[10] K.J. Åström and T. Hägglund, "PID Controllers: Theory, Design, and Tuning", 2nd edition. *Instrument Society of America*, 1995.

[11] F. Takens. On the numerical determination of the dimension of an attractor. In: D. Rand and L. S. Young (Eds.). Dynamical Systems and Turbulence, Warwick 1980 Lecture Notes in Mathematics, Vol. 898. Springer Verlag, 1981, pp 366-381

[12] R. Mañe. On the dimension of the compact invariant sets of certain non-linear maps. In: D. Rand and L. S. Young (Eds.). Dynamical Systems and Turbulence, Warwick 1980 Lecture Notes in Mathematics, Vol. 898. Springer Verlag, 1981, pp 230

LIST OF FIGURE CAPTIONS

Fig. 1. - Calibration wind tunnel.

Fig. 2. – Sensor electrical connection diagram.

Fig. 3. – Achieved accuracy versus air speed.

Fig. 4.- Main flow chart for the virtual instrument.

Fig. 5.- Voltage signals that are input to the network (top) and outputs of the network (bottom) for a typical run of the system where two failures were forced. ANN output 1 represents failures and ANN output 2 indicates when it is safe to measure.

Fig. 6.- Results obtained with the second neural network used. Induced failures are of the same type as those of Figure 5.

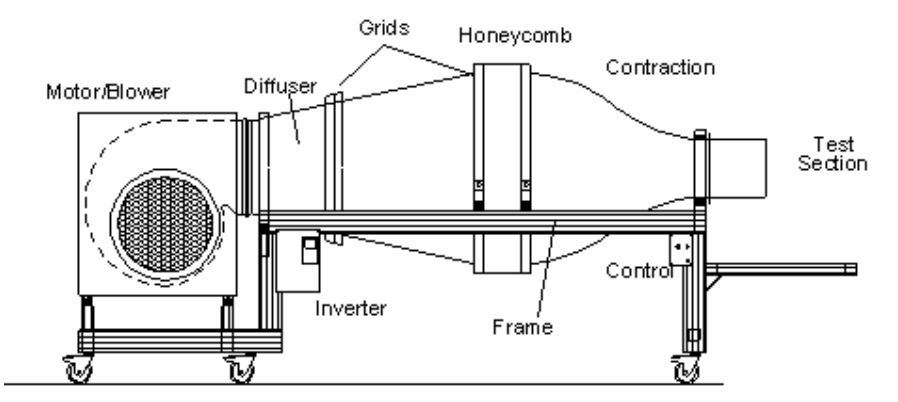

Fig. 1. - Calibration wind tunnel.

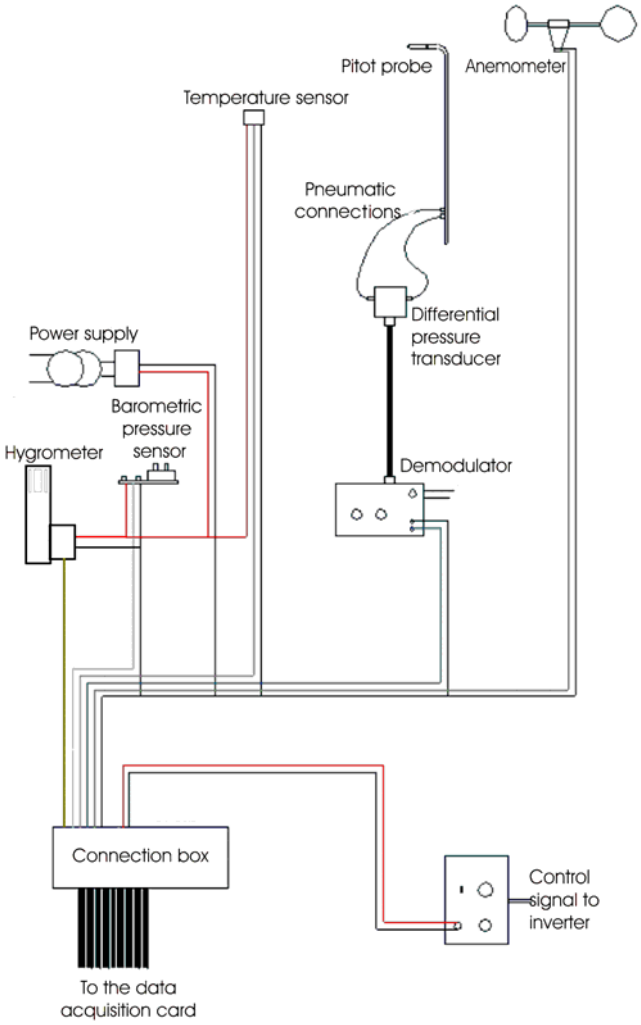

Fig. 2. – Sensor electrical connection scheme.

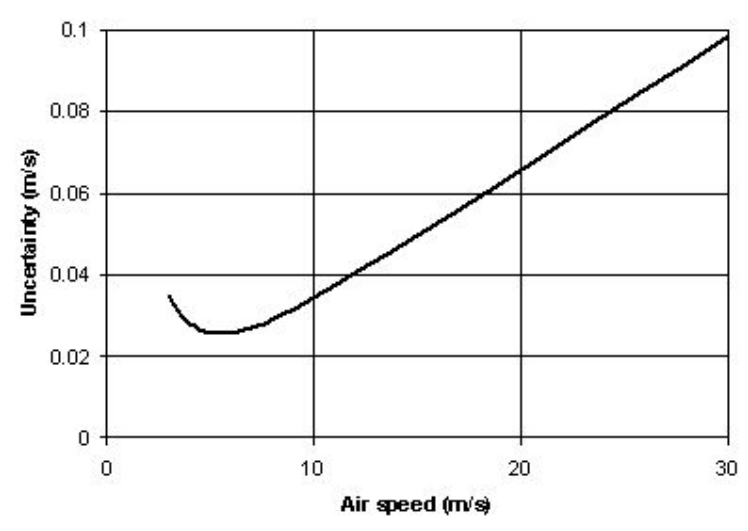

Fig. 3. – Achieved uncertainty versus air speed.

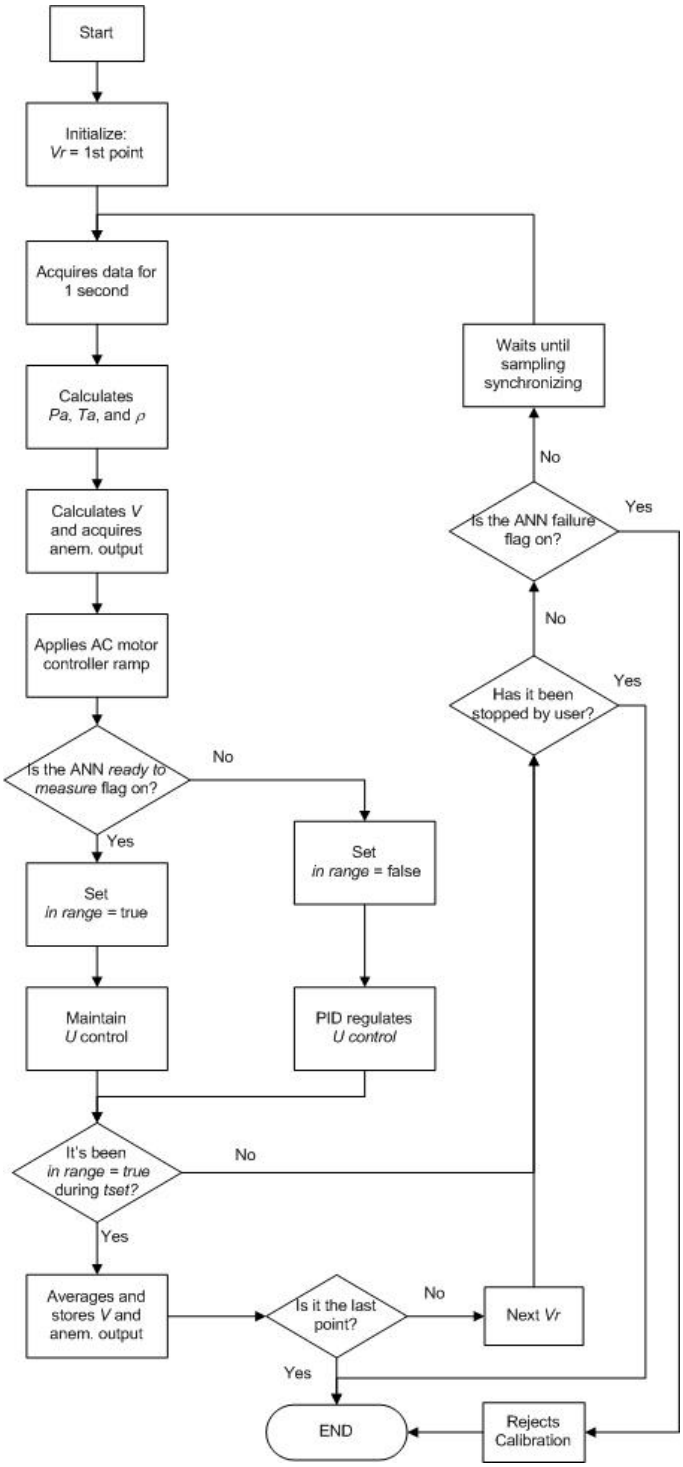

Fig. 4.- Main flow chart for the virtual instrument.

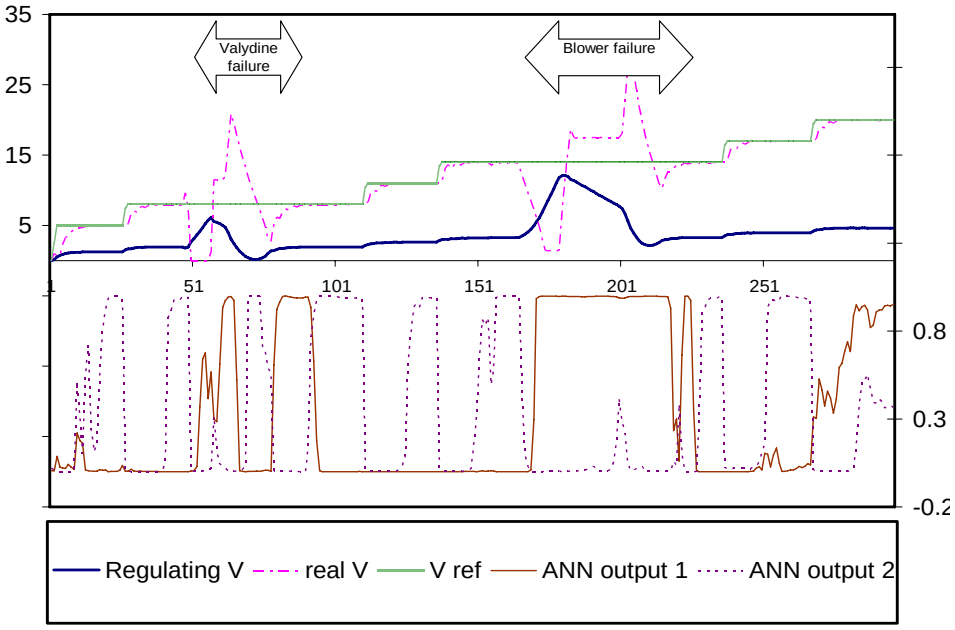

Figure 5.- Voltage signals that are input to the network (top) and outputs of the network (bottom) for a typical run of the system where two failures were forced. ANN output 1 represents failures and ANN output 2 indicates when it is safe to measure.

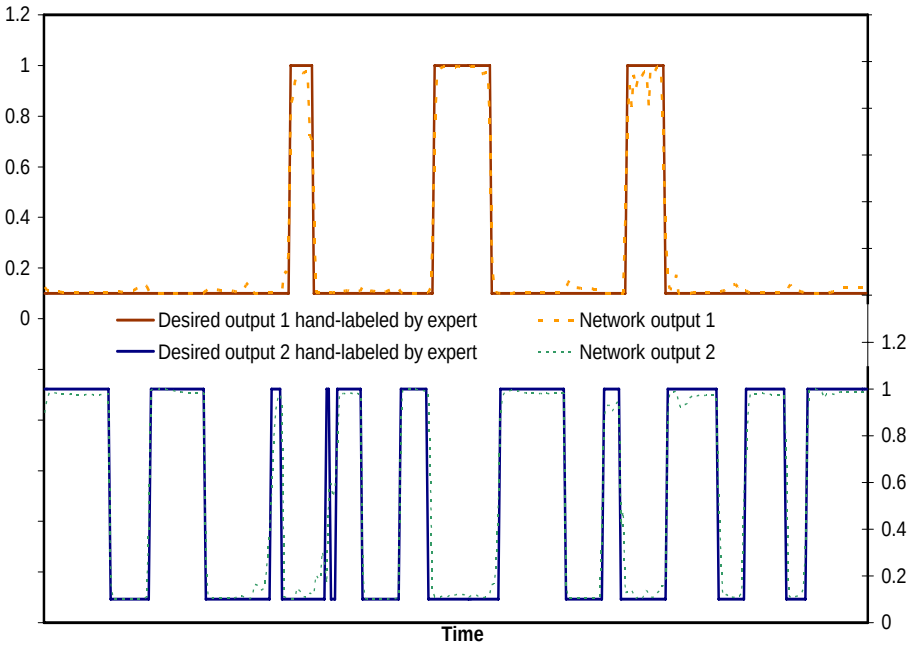

Figure 6.- Results obtained with the second neural network used. Induced failures are of the same type as those of Figure 5.