## Annex D
Evaluation of alternative or complementary processing methods for the needle-bar signals

**Index**

1. **WAVELET FILTERING AND RECONSTRUCTION USING “AUTOSIGNAL”**.......................... 2  
   1.1. OBJECTIVE................................................................................................................. 2  
   1.2. BRIEF OVERVIEW OF THE WAVELET TRANSFORM ...................................................... 2  
   1.3. DECOMPOSITION OF VOID AND SIMULATED PENETRATION SIGNALS USING WAVELETS .............................................. 5  
   1.4. CONCLUSIONS........................................................................................................... 8  
2. **WARPING AND PARTIAL NULLING OF THE SIGNALS**.................................................. 9  
   2.1. OBJECTIVE.................................................................................................................. 9  
   2.2. TIME - WARPING ...................................................................................................... 9  
   2.3. NULLING OF SIGNAL PORTIONS................................................................................... 9  
   2.4. RESULTS ..................................................................................................................... 10  
3. **PRELIMINARY EVALUATION OF THE USE OF ARTIFICIAL NEURAL NETWORKS TO CORRECT RESULTS**............................................................................................ 12  
   3.1. OBJECTIVE................................................................................................................ 12  
   3.2. BRIEF OVERVIEW OF ARTIFICIAL NEURAL NETWORKS........................................... 13  
   3.3. ANN TOPOLOGY USED ............................................................................................. 16  
   3.4. QUALITATIVE ANALYSIS OF THE METHOD’S EFFICIENCY ........................................ 19  
4. **REFERENCES**............................................................................................................. 27
1. Wavelet filtering and reconstruction using “AutoSignal”

1.1. Objective

The thought of using wavelets to decompose the needle-bar force signals derives from the nature of the wavelet transform.

With the wavelet transform, a signal is decomposed as a sum of base functions of finite length, as opposed to the Fourier transform, where signals are decomposed into a sum of sine functions. This base function, known as wavelet, is, in principle, a better way of decomposing needle-bar force signals. The wavelet should be able to approximate the transitory burst produced by the needle penetration and withdrawal better than the sine functions used in the Fourier transform.

In this way, decomposition, filtering and reconstruction of the signals, eliminating uninteresting components, would be possible with less error.

Another possibility of performing feature extraction would be directly on the wavelet spectrum. The wavelet transform presents the major advantage over the Fourier transform of delivering not only frequency, but also time information. This means that not only is it possible to detect which frequency components are contained in a signal, but also when they occur.

A basic description of the wavelet transform and its properties, based on [2], is given in the next section. A more thorough analysis is outside the scope of this work.

To test the performance and usability of the wavelet transform, a trial version of the “AutoSignal” program (©1999, AISN Software, Inc.) was used. AutoSignal is a package with extensive signal processing tools, among which the Wavelet Transform and Wavelet-based filtering and reconstruction are included.

1.2. Brief overview of the Wavelet transform

The wavelet transform expands a signal into a sum of wavelets.

The term wavelet means “small wave”. “Small” is used in the sense that the wave is of finite length (“compactly supported”). A mother wavelet is used as a prototype, and its time-scaling and translation results in the wavelets used to actually decompose the signal.

Many functions can be used as mother wavelets. Besides the finite length, they have to possess an additional set of properties to qualify as a mother wavelet. An example of a wavelet, the widely used “Morlet Wavelet”, is given in Figure 1.
The mother wavelets possess adjustable parameters that can be used to tune the mother wavelet for the signal to be decomposed. In the case of the Morlet function, the parameter is the wave number. The wavelet presented in Figure 1 will result in the function depicted in Figure 2 when its wave number is increased from 6 to 12.

To compute the wavelet transform (more precisely, the continuous wavelet transform, CWT), the mother wavelet is scaled (dilated or compressed in time) and translated (shifted) on the time-scale. As in every transform, a correlation between the signal and the scaled and translated wavelet is computed. It is defined as follows [2]

\[
CWT_x^\psi (\tau, s) = \Psi_x^\psi (\tau, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} x(t) \psi^\ast \left( \frac{t-\tau}{s} \right) dt
\]  

(1)

where

- \( CWT \): Continuous Wavelet Transform
- \( \psi \): Transforming function
- \( \tau \): Translation factor
- \( s \): Scale factor

Figure 1: Morlet wavelet, wave number=6

Figure 2: Morlet wavelet, wave number=12
The wavelet transform has a bidimensional argument: The scale \( s \), with which frequency information is extracted, and the translation \( \tau \), that introduces time information. Although named “continuous”, for practical purposes, in which the transform is calculated by a computer, a finite set of scales and translations is used. The correlation between the signal and the set of wavelets is obtained by integrating the multiplication of the wavelet and the signal over all time. The factor \( \frac{1}{\sqrt{s}} \) is used for energy normalization.

The presence of time information is an advantage of the wavelet, especially suited for non-stationary signals. Other transforms are also able to deliver time information. The Short-Time Fourier Transform (STFT), for example, is a method for obtaining frequency content over time. It is obtained by windowing a time signal over short periods of time and calculating an FFT for every portion of the signal obtained in this way.

The STFT, however, presents a resolution problem that can impair its use in some situations. Narrowing the time window causes a loss of resolution in the frequency domain, which can cause energy of neighbouring frequency components to overlap.

Unlike the STFT, where a constant time window is used and thus a constant frequency resolution is obtained, the wavelet transform presents a significant advantage by being a multiresolution analysis (MRA). In fact, the scaled and translated wavelets represent a window function by themselves, and since they are scaled to different widths, the resolution is not constant over the whole transform domain.

The CWT presents a good time resolution with poor frequency resolution at high frequencies, and a poor time resolution with good frequency resolution at low frequency. This property makes sense on signals that possess high frequency components for short durations and low frequency components for long durations, as is the case in most signals of practical applications.

For a better understanding of the wavelet transform, a signal has been composed as a sequence of three sine waves of 10, 30 and 20 Hz (Figure 3).

![Figure 3: Signal composed for CWT demonstration](image)
The CWT can be represented in several ways as a 3-D graph (Figure 4 uses a shaded graph), or as a 2-dimensional plot known as a contour graph (Figure 5).

In the contour graph, the value of the CWT is represented as a colour on the time-frequency plane. Both graphs deliver the same information. The contour graph is easier to read on a printed document, whilst the 3-dimensional representation is more advantageous for observation on a computer where it can be interactively turned and zoomed.

![Figure 4: Shaded 3-D graph of the CWT (Morlet, wave number 8) of signal depicted in Figure 3](image)

![Figure 5: Contour plot of the CWT (Morlet, wave number 8) of signal depicted in Figure 3](image)

In both representations the sine waves that compose the signal can be localised in time; an FFT would indicate the presence of the three frequency components, but it would not be possible to localise them.

1.3. Decomposition of void and simulated penetration signals using wavelets

After a first approach to the fundamentals of the CWT, an analysis of both the simulated penetration signals as well as the void signals was made.
The primary objective was to tune a wavelet so that the components of penetration and inertial forces could be separated in the CWT. This would then allow the elimination of the uninteresting signal components and reconstruct a signal containing only penetration and withdrawal forces.

A total separation has been found not to be possible. The fundamental issues that hinder the separation are twofold:

> Although the penetration burst is only present on part of the sewing cycle, it is linked to the inertial force component, having the same periodic nature and the same fundamental frequency. Inherently, the spectrum presents energy at the same fundamental frequency and some harmonics, regardless of the transforming function. Elimination of these components unavoidably distorts the signal;

> The void signal contains discontinuities resulting from mechanical imperfections that generate a similar spectral content as the penetration signal; elimination of these components would be equivalent to eliminating the penetration signal itself.

Figure 6 shows the spectral content of a void signal acquired at about 600 s.p.m (10 Hz) transformed with a Morlet wavelet, wave number 8.

![Figure 6: CWT of a void signal acquired at about 600 s.p.m (10 Hz) transformed with a Morlet wavelet, wave number 8](image)

The plot shows an important component at the base frequency of 10 Hz and some energy at 20 Hz. This might lead to the conclusion that eliminating component 1 and 2 would be enough to eliminate all energy of the void signal. But a closer look at the 25-80 Hz range shows that some residual energy is still left on other frequency bands:
The better time resolution at higher frequencies shows that the residual energy present at higher frequency bands is related to the discontinuities of the signal, since it is localised.

It cannot be eliminated from a real signal without eliminating useful information (->the penetration signal).

Figure 8 shows the wavelet transform of a SinVar4 signal with a 10 Hz periodicity, computed with the same mother wavelet as the spectra presented previously.
It is quite clear that the signal contains a significant amount of energy at the fundamental frequency and at the component 2 of the spectrum. This frequency band corresponds to the frequency band of the void signal, and a total separation is thus not possible. Nevertheless, using a Gauss-Derivative mother wavelet of order 2, a more optimistic observation is possible:

![Continuous Wavelet Time-Frequency Spectrum](image)

Figure 9: CWT of a SinVar4 signal computed with a Gauss Derivative mother wavelet of order 2

Regrettably, the same effect of “energy transfer” to higher frequencies is also present in the decomposition of the void signal, and the overall result of the filtering process may not improve; only a more exhaustive study can show if this approach is useful.

### 1.4. Conclusions

In a first and quick analysis, the wavelet transform has not been found to substantially improve the filtering process when compared with the FFT-based filtering method.

However, it has been observed that the choice of the wavelet and the adjustment parameter significantly affects the result of the wavelet transform. A more thorough study may lead to ways of separating the useful signal more effectively.

In addition, only mother wavelets pre-defined within the AutoSignal software have been used, but other functions can be used as a transforming function, as long as they possess certain properties. The choice of a more adequate mother wavelet to match the penetration signal may lead to better results.
2. Warping and partial nulling of the signals

2.1. Objective

This section describes two techniques used to modify the signal by eliminating or nulling the portions of the signal outside the penetration phases. This procedure was tried to learn if it would be useful in the separation of the penetration and withdrawal components.

2.2. Time-warping

In this method, the portions of the signal that lie within the penetration phases are extracted as subsets that are subsequently concatenated to produce the final result. Figure 10 shows the result of this operation on a SinCons signal.

![Figure 10: Original and warped SinCons signal](image)

Figure 10 illustrates the outcome of time-warping on a void signal.

![Figure 11: Original and warped void signal](image)

The warped signals’ sample frequency is adjusted to maintain the same fundamental frequency of the signal. This produces the effect of “stretching” the penetration portion of the signal over the whole cycle.

2.3. Nulling of signal portions

In this case, the signal portions outside the penetration phase are nulled. Figure 12 shows the result of this operation on a void signal.

The procedure leaves the penetration signals generated by Anfil intact, since they are intrinsically nulled outside the penetration phases.
Both of the techniques described have not led to useful results. In fact, separation of signal components is more difficult after this pre-processing procedure.

The effect of warping the penetration signal is the transfer of signal energy to lower frequency bands. This can be observed in Fig, where the components 1 to 30 extracted of a warped SinVar4 signal can be compared with the harmonics of the original.

As described in chapter IV, the penetration signal's energy is shifted to and concentrated in lower frequency bands when the penetration bursts' duration is increased. This is the effect that time-warping has on the signal, regarding that sample frequency is adjusted to maintain the same fundamental frequency of the signal, and thus the penetration phase is artificially longer than in the original signal.

Figure 13 also shows that most of the signal's energy is contained in components 0 to 3 after warping. This is the frequency band that is being used as stopband after first studies of spectral filtering in this application.

On the other hand, changes in the void signal's spectrum do not bring any significant benefit to the filtering operation. Fig shows the first 30 components of the original, warped and nulled void signal. The graph is
zoomed in the y-scale to show the contribution of components 3 to 30 to the signal’s energy.

Figure 14: Components 0 to 30 of original, warped and nulled void signal

Both warping and nulling produce sharp transitions of the remaining void signal. These will appear as energy spread over higher frequencies. Unsurprisingly, energy in the lower frequency components has slightly decreased, as shown in Figure 15. They are still significant and have to be filtered in the same way as in the original signal.

Figure 15: Components 0 to 5 of original, warped and nulled void signal

The results presented enable therefore the conclusion that neither the warping nor the nulling method aid in the separation of the penetration signal.
3. Preliminary evaluation of the use of Artificial Neural Networks to correct results

3.1. Objective

After seeking an alternative for the filtering process, the idea of using artificial neural networks (ANN) to correct the results after filtering, arose.

ANN’s are networks of processing elements that can be trained to approximate virtually any mathematical function. They are used when an analytical representation of the function is not possible to obtain or when a function is to be defined from noisy or inaccurate data.

The purpose of studying ANN’s in the context of this work is to train a network to approximate the following function:

\[
[y_1; y_2; y_3] = f(x_1, x_2, x_3)
\]  

(2)

where

\[y_n:\] Feature value extracted from signal in phase \(n\) after filtering

\[x_n:\] Original feature value for phase \(n\)

It is actually the inverse of the function described in eq.(2) that is sought for. The idea is to filter the signal, extract the feature values and correct them using the ANN.

The existence of such a function is justifiable by the properties of the penetration signals. As stated in chapter IV-2.4.4, “the convolution of a base signal with a square wave has the effect of repeating the base function’s spectral information over the whole frequency spectrum”. This means that it may be conceivable to obtain a function to invert the process. Given the complexity of the penetration signal’s spectrum, such a function is very difficult or even impossible to be obtained analytically, but a neural network may be trained to approximate it. In this perspective, the network should also be able to be trained to correct values obtained by other filtering stopbands, as for instance the 0-1-2-3 stopband, that delivers the lowest residual values of inertial forces. This possibility would represent a major advantage and will also be studied.

The Anfil program created to generate simulated penetration signals is an excellent tool to generate training data for such a neural network. Different signal models with varying amplitude ratios can be used to generate a great amount of training data.

Since the field of ANN’s is a research subject on its own, the author has limited its study on the basics of ANN’s and on the capabilities of commercially available software. A further study is outside the scope of this work.
3.2. Brief Overview of artificial neural networks

Although the biological NN’s are much more complex than ANN’s, originally the design of ANN’s was inspired in their biological counterpart, resulting in some analogies between them.

An ANN is a network of simple processing units (nodes or neurons). These units are connected by unidirectional connections (synapses), which carry numeric data. The processing units operate on their local data (in cases in which the node has a local memory) and on the inputs they receive via the connections. The strength of the connections, usually called weights, are adapted in a training (learning or teaching) phase such that the network’s output matches a desired response to an input.

ANN’s are a mathematical technique like any other, with the distinction that its design has been motivated by the biological NN.

Many different topologies for ANN’s have been developed. The simplest is the layered feedforward network, often called “multilayer perceptron” that has been used in the context of this work. A simplified diagram of an example structure is given in Figure 16.

![Diagram of a three layer feedforward ANN](image)

Figure 16: Example structure of a three layer feedforward ANN
This network has three layers: the output, input and an additional hidden layer. In a layered network, normally each node is connected with every node in its neighbouring layer. There are, however, many alternatives to this connection scheme.

The network is a feedforward network because the numerical information is carried over its connections unidirectionally (from input to output). Other topologies can use bidirectional connections or loops.

Every connection has a certain weight associated to it. The total input to the node $j$ of the network is computed on the node as

$$i_j = \sum_{k=1}^{n} w_{kj} \cdot o_k$$  \hspace{1cm} (3)

where

- $i_j$: Total input to node $j$ (activation value)
- $w_{kj}$: Weight of connection between node $k$ and node $j$
- $o_k$: Output of node $k$
- $n$: Total number of nodes connected to node $j$

The value computed in eq.(3) is called the activation value of node $j$.

The output of node $j$ is a function of its activation value:

$$o_j = f(i_j)$$  \hspace{1cm} (4)

This function is called activation function or transfer function. Many activation functions are used in ANN's. The most widely used is the sigmoid:

$$o_j = \frac{1}{1+e^{-i_j}}$$  \hspace{1cm} (5)

The use of a smooth non-linear function is necessary for backpropagation, the learning method commonly used with multi-layer perceptrons [6]. A learning requires the following components:

- A measure of performance, in order to quantify the performance of the training method and stop it if performance is deteriorating;
- A rule for weight adjustment that assures an improvement of performance during the training phase.
In backpropagation, the performance and weight adjustment are based on an error measure. The error is the difference between the current network output and the desired response, which means that the output must be known. This type of learning is said to be supervised. The principle of this learning method is depicted in Figure 17.

![Figure 17: Principle of supervised learning](image)

In the context of this work, the multilayer perceptron is the obvious first choice for a first approach to the desired function approximation. It is readily available in commercial packages, has a good learning performance and is believed to be capable of approximating arbitrary functions[6].

To test the effectiveness of ANN’s in correcting the extracted feature values, an evaluation version of the Neurosolutions application of NeuroDimension, Inc, was used.

This package supplies an extensive set of tools for ANN development. The user can design the structure of an ANN and then train and test it. Moreover, the program provides an uncomplicated ANN construction “wizard” with which it is possible to set up a typical ANN for a specific application in a very quick and uncomplicated way, hiding the complex details of ANN’s. This wizard has been used in this work.

After the development phase, a module of the NeuroSolutions program generates C code that can be integrated into other applications. This is an important advantage in the context of this work, since it makes an integration of a correction algorithm in the AST program possible.

The trial version of NeuroSolutions does not allow the user to output numerical data in any way. This means that in the first experiments the evaluation of the effectiveness of ANN’s for the purposes in hand was only possible in a qualitative way.
3.3. ANN Topology used

In this work, the ANN construction wizard (called “NeuralExpert” within NeuroSolutions) has been used to try the function approximation application. This wizard leads the user through the following steps until it generates the topology and all tools for its training, test and evaluation:

1. Problem type specification
   The user selects the type of problem to be solved. This can be classification, function approximation, prediction or clustering.

2. Input Data
   The user indicates the path to the file where its input data is located. This is the data the network will be trained with. In the present work, these are sets of feature values extracted after filtering, each of the three phases supplying one value.

3. Desired Data
   In this step, the path to the file containing the desired response of the network is indicated. In this work, these are the feature values extracted directly from the original (simulated) penetration signal.

4. Generalization protection
   The user indicates a percentage of the input and desired data that should be put apart to be used as cross-validation data. Cross-validation data is data that is not part of the training data and with which the performance of the network is evaluated. This serves to stop the training process when a sufficient performance is obtained. Failing to do so, the network may be overfitted or overtrained. This means that it specializes on the training set, and is not general enough to perform equally well on data it has not seen yet.

5. Genetic Optimisation
   It is possible to optimise the network using genetic algorithms, a technique that will not be analysed nor used in this work.

6. Network complexity
   In this step the user may choose a low, medium or high complexity for the network. This results in a different number of hidden layers. Usually, a low complexity is used in a first approach and is increased if the results are unsatisfactory.

After this set-up process, the software generates a neural network and auxiliary tools that result in the screen shown in Figure 18.
This screen shows the structure of the ANN and all the “probes” inserted in the network to view the values at specific nodes, among other elements. By default, probes are set on the Error node and at the outputs.

The software displays graphs representing the output of the network and the desired values of both the training and cross-validation data set. These graphs allow the user to observe the training process and verify the convergence of the output to the desired value in the process.

The network structure generated by the application by default for function approximation is a multilayer perceptron with an input, an output and a hidden layer, which represents the “low complexity” option in the software. The layers are called “axons” within the application. Some attempts to use medium or high-complexity networks (that add new hidden layers) curiously resulted in a poorer performance, so that the low complexity option was maintained through the remaining experiences.

The hidden layer uses a non-linear activation function based on the hyperbolic tangent, whilst the output layer has a biased linear transfer. The representation of the network that can be seen in Figure 18 also includes an error criterion element at the output that implements a quadratic error function (L2 cost function). This error is fed back through the backpropagation plane and is used to adjust the weights at the forward propagation plane by means of a learning rule (momentum learning by default).

It is possible to modify the network after the wizard has generated it. This possibility has been used to change the input axon to a special type of...
axon that possesses a delay line ("TDNN axon"). This axon has a memory structure that allows the present input sample to be presented to the network together with delayed input samples. Two parameters define this process: the delay and the number of delay lines. A TDNN axon with a delay of 2 samples and 3 delay lines would present, at input sample $n$, the values $x(n), x(n-2), x(n-4), x(n-5)$. The use of the TDNN axon in this work will be described later.

Training is performed in epochs. This means that the training set is repeatedly presented to the network. It generates an output for the input value presented and the error between the current and desired output provides the basis for the adjustment of the network’s weights.

After the training phase, neural networks have to be fed with data that fits in the same range as the data used for training. Data that is outside this range will cause errors. To illustrate this, a network has been trained to approximate the function

$$y = x^2 + 1$$

(6)

The network is trained with a data set generated in a spreadsheet program for $x \in [-100,100]$. The last 20% of this data are set apart for cross-validation. Figure 19 shows the output of the network on the training and of the cross-validation data after 1000 epochs of training.

![Figure 19: Network trained to approximate a quadratic function](image)

As can be seen, the output of the network approximates the function quite well in the training set. The cross-validation set, however, lies outside
the input range of the training set, and the network performs poorly in this range.
To overcome this issue, it is possible to normalize the input data by shifting and scaling it.

3.4. Qualitative analysis of the method’s efficiency

The first approach to training the ANN was carried out before the optimisation of the FFT-filtering algorithm presented in chapter IV-2.6. A 0-1-2-3C stopband was used to extract the peak values and a data set with all possible signal models and amplitude combinations was used.

Many experiments were made, with different data sets. The original feature values (peaks) of the most representative data set is depicted in Figure 20.

![Figure 20: Data set used for ANN training in preliminary experiments](image)

The qualitative analysis of the results on different test sets didn’t show much success in the correction of feature values.

After this first approach, the use of ANN’s was provisionally abandoned. The optimisation of the FFT-filtering procedure was performed and the use of ANN’s was tried again using the optimised 0-1-3 stopband.

This time, the SinVar4 signals were used in the same way as they had been used for filtering optimisation. The same data sets were used to train the network, containing 192 stitches: SinVar4 signal in 4 amplitude combinations, 3 stitches per combination.

In the first experiment, the ANN was trained with a data set extracted directly from the penetration signal, without the void signal. Figures and show the convergence of the network for peak and power values on the training set.
As can be observed, the network converges quite well for peak values but its performance is poorer when trying to extract power values.
After this training phase, several test sets were composed out of different signal models.

The test set based on the SinVar4 model showed a similar performance as the training set; this fact is an important result, meaning that the network is able to generalize its result to other sets than the training set.

It is also very interesting to observe that the network performs reasonably well for other signal models. The next figures show its output and desired values for different signal models.

Figure 23 shows that the output of the network is able to follow the values extracted from the SinCons signal quite effectively. But it is also possible to observe that the output value still depends on the values extracted from the other phases. This seems to be an effect that the network is unable to correct completely and only a quantitative analysis can show if the improvement in performance is interesting.

The next figures show the performance of the network on values extracted from the SinVar and SinVar7 models.
Figure 24: Output and Desired values for the SinVar test set in 1 amplitude combination, peaks/valley

Figure 25: Output and Desired values for the SinVar7 test set in 4 amplitude combinations, peaks/valley
It can be seen that the performance obtained for the SinVar model seems to be very interesting, but it becomes poorer for the SinVar7 signal. The detection of singular values is an important aspect of this analysis. The network has therefore been tested with a sample based on the SinVar5 model. The result is depicted in Figure 26.

![Figure 26: Output and Desired values for the SinVar5 test set in 4 amplitude combinations, peaks/valley](image)

The graph shows that in certain situations, the correction of values in phase 3 is done with some error. Furthermore, the singular value seems to be detected in the period following the one in which it occurred. This happens especially in periods in which the peak values are more unbalanced between phases. It has to be taken into consideration that the SinVar5 model generates, at the singular value, an amplitude ratios that the network has not been trained with.

An attempt to reduce these errors has been made by using a TDNN axon with three delay lines at the input of the neural network. The network is thus trained using the current input and 3 past inputs, and may in this way model the measurement distortion produced by a previous value more adequately. The result with this network seems to be better than with the previous architecture and it performs equally well on other signal models. Another solution to this issue may be to train the network with more than the 4 amplitude combinations that have been used. This would allow the network to adapt itself to input values that are more unbalanced.

Finally, the network was tested with a data set obtained from a SinVar4 signal mixed with a void signal. The SinVar4 signal was scaled to produce
a signal-to-noise ratio of 1:2. Naturally, a lower performance is expected in this case, since this data set includes an effect that the network has not been trained with. Figure 27 shows the result obtained with this test set.

![Figure 27: Output and Desired values for a test set in which a SinVar4 in 4 amplitude combinations was mixed with a low-speed void signal, peak values](image)

The last experiment in this preliminary analysis was to train the network with a data set that was obtained on basis of a penetration signal mixed with a low-speed void signal. The network’s output converged well on the training set, and two test sets were generated with a low and a high-speed void signal. In the high-speed test signal, the same signal-to-noise ratio was maintained, but the data set was normalised to the data range of the low-speed signal.

Figure 28 and Figure 29 show the outcome of this experiment on the two test sets:
Figure 28: Output and Desired values for a test set generated with low-speed void signal, peak values

Figure 29: Output and Desired values for the normalised test set generated with high-speed void signal, peak values
The effectiveness of the correction method appears to be quite interesting in this situation, as shown in the figures, but only a quantitative analysis of these results, in different situations, can validate this processing method.
4. References


[7] Donald R. Tveten, The Pattern Recognition Basis of Artificial Intelligence, procurer na IEEE, não sei se é este o livro onde vem o doc de que estou a retirar os elementos (“Backprop”)