We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Data Simulation and Trend Removal Optimization Applied to Electrochemical Noise

Victor Martinez-Luaces and Mauricio Ohanian

Additional information is available at the end of the chapter

Abstract

A well-known technique, electrochemical noise analysis (ENA), measures the potential fluctuations produced by kinetic variations along the electrochemical corrosion process. This practice requires the application of diverse signal processing methods. Therefore, in order to propose and evaluate new methodologies, it is absolutely necessary to simulate signals by computer data generation using different algorithms. In the first approach, data were simulated by superimposing Gaussian noise to nontrivial trend lines. Then, several methods were assessed by using this set of computer-simulated data. These results indicate that a new methodology based on medians of moving intervals and cubic splines interpolation show the best performance. Nevertheless, relative errors are acceptable for the trend but not for noise. In the second approach, we used artificial intelligence for trend removal, combining an interval signal processing with backpropagation neural networks. Finally, a non-Gaussian noise function that simulates non-stationary pits was proposed and all detrending methods were re-evaluated, resulting that when increasing difference between trend and noise, the accuracy of the artificial neural networks (ANNs) was reduced. In addition, when polynomial fitting, moving average removal (MAR) and moving median removal (MMR) were evaluated, MMR yielded best results, though it is not a definitive solution.

Keywords: electrochemical noise, data simulation, signal processing, trend removal methods, noise filtering, artificial neural networks

1. Introduction

To accurately determine the life expectancy of an industrial component, it is necessary to quantify the metal deterioration by environmental influence —for example, construction,

automotive, naval and aeronautic industries, among others. For this reason, it is important to obtain accurate methods to measure and predict deterioration processes [1].

A well-known methodology is ENA, which measures variations in current and/or potential fluctuations, produced by fluctuations in electrochemical process kinetics. An important advantage of this corrosion measure technique is that no external signal perturbations are necessary, as the electrochemical system uses natural techniques for measurement. Additionally, measurement devices are better in terms of affordability in comparison with other techniques [2]. Moreover, ENA provides information in terms of current and potential fluctuations in low amplitude and frequency and presents more complexity in experimental data treatment. The application of ENA technique yields information about the corrosion process mechanisms, kinetics, and morphology [3] from the calculation of the parameters of noise resistance ($R_N = \sigma_V/\sigma_I$), localization index ($LI = \sigma_I/I_{RMS}$), and power spectral density (PSD).

In this process, an electrochemical signal that follows a particular trend is perturbed with a certain noise. In real conditions, noise appears as a result of corrosion reactions, while the trend is due to hydration processes, species dissolution, and thermal cycling, among other processes, which are not necessarily associated to the material deterioration. Therefore, to understand corrosion behavior, it is crucial to detrend the signal function isolating this noise. This procedure is called trend removal [3]. Several methodologies have been applied to implement it, especially statistical procedures [4]. Despite having been proved to be useful in some cases, a really accurate procedure for trend removal is still an open problem.

In fact, it is relatively easy to remove noise from a noisy function by polynomial approximation (ANNs) [5] or genetic programming [6, 7]. However, it is extremely difficult to do the inverse filtering. In fact, as shown in Section 3, trend removal cannot be performed accurately by traditional techniques such as polynomial approximation. If this technique is used, low-level relative errors in trend approximation become unacceptable to high-level ones in case of noise approximation, since errors and noise usually have the same order of magnitude. For this reason, other algorithms and computational techniques—as ANNs, analyzed in Section 4— must be studied in order to improve the traditional detrending methods, proposing new feasible and adaptable trend removal tools for different purposes.

Additionally, it is important to mention that the main disadvantage of ENA technique is the high dispersion observed on experimental results [2]. In a previous work [8], three different factors (electrolyte, frequency simulation, and trend removal) were analyzed with a view to determine causes for the observed high dispersion. The experiments done, modifying these parameters, showed that the trend removal method is the one which best explains this phenomenon.

In order to assess the performance of the different methods, it is a must to know exactly both trend and noise for a particular signal. Then, the error levels obtained using these methods can be statistically compared.

Due to these reasons, trend and noise were both computer simulated, as described in detail in the next section.

2. Data simulation for ENA

2.1. Analytical trend simulation

In order to simulate the artificial trend, nontrivial functions must be used in the process. As mathematical methods can easily be used to approximate them, trigonometric functions, polynomials, and exponentials can be considered as trivial trend lines. This is a relevant aspect that should be taken into account when trend removal methodologies are evaluated.

For this purpose, we used two different curves as trend lines, since they are nontrivial and at the same time their graphics look like the experimental curves obtained in previous research.

The first trend line considered was: $\frac{f(t)=ac^{t/3}}{\Gamma(\frac{t}{3}+1)-b}$ (Curve 1), where $\Gamma(x) = \int\limits_{0}^{+\infty} t^{x-1}e^{-t}dt$ is the Euler's Gamma function. This transcendent function cannot be approximated by elementary methods, and at the same time, its shape is similar to one of our experimental curves, and for this reason it was selected among others studied in a first approach.

A second trend line chosen for evaluating the detrending methods was the Lorentz's function: $f(t) = y_0 + \frac{2A}{\pi}\frac{w}{4(t-x_c)^2+w^2}$ (Curve 2), which also looks similar to one of the experimental curves obtained in previous experiments. Besides, due to the Runge's phenomenon (see the similarity of $f(t)$ with Runge's function), this curve cannot be easily approximated since going to higher degrees polynomial interpolation does not improve accuracy.

2.2. Noise distribution

As in case of trend, the goal with noise is to simulate real conditions as much as possible. In real conditions, noise is unpredictable, and so a random function for noise data generation should be used.

In a first approach, noise data generation was implemented using an inverse Gaussian distribution. To do this, a numeric algorithm was applied [9] by splitting into three sections the Gaussian function according to the derivative level (low, medium, and high) and then applying different algorithms to each section for error minimization. Once done, noise generated function is added to trend function, obtaining a noisy signal adequate for trend removal testing.

3. A first approach about detrending methods

Some theoretical aspects of the trend removal problem were developed by K. Hung Chan and colleagues [10]. The authors presented differences between the most used trend removal methods: first difference method (also named point-to-point method) and least squares. They utilized a linear trend superimposed with white noise and the first difference method gave

power spectral density (PSD) exaggerated at high frequencies and attenuated at low frequencies. On the contrary, the regression residuals method generates results with PSD exaggerated at low frequencies and attenuated in the high-frequency region.

Mansfeld et al. [11] studied the characteristics of detrending in ENA data with an experimental and theoretical approach. Thus, by applying linear fit to the trend of experimental data, they obtained a good concordance between the noise spectra after detrending and impedance spectra. The authors found that MAR, previously proposed by Tan et al. [12], caused erroneous ENA results.

The performance of most used trend removal methods (analogical high-pass (HP) filtering, digital high-pass filtering, MAR, and polynomial fitting) was analyzed by Bertocci et al. [13]. They worked on a simulated data set achieved by superimposing white noise to a linear trend and applied the detrending methods. They studied the box size influence on MAR, concluding that a small box size effectively removes the trend without phase shift. However, MAR also removes the signal information at low frequency. A high-order MAR (greater box size) generates alterations in the signal shape. With the intention of evaluating the produced attenuation, the authors used polynomial detrends of different orders. The components in frequency $1/T$ and $2/T$ window (T being the experimental period) were eliminated when a fifth-order polynomial was used. The best results were obtained when the polynomial methodology was shared with prepossessing windowing. By using the Chebyshev filter (digital processing) of different cutoff frequencies, the authors recovered the white-noise PSD. They concluded that the use of high-pass (HP) analogical filtering is the ideal method to eliminate low-frequency components from the experimental signal. Although filters of cutoff frequency near to $1/T$ with low intrinsic noise were too costly. Besides, the major drawback of analogical filters in real-time processing is related to the oscillations that the components have in abrupt signal variations, that is, when switching on the system.

Ohanian et al. [8] found a great dispersion in the experimental ENA results on low-alloy steel performed in saline solution. The authors attributed the excessive dispersion mainly to the detrending method used. On a simulated data set, they analyzed the polynomial order influence and the aliasing phenomena.

As we mentioned earlier, there are many methodologies for trend removal to isolate ENA. In a first approach, several methodologies were analyzed, since they were studied by well-known authors like Mansfeld [11], Tan [12], and especially Bertocci and Huet [13], who evaluated the most commonly used trend removal methods. Four of the detrending methods from this group are described in detail, as they showed better performance than others that have been assessed by using the computer-simulated data. Three of them can be considered traditional methods, since they appear regularly in papers on studies performed on trend removal technique using ENA. The fourth method was proposed by our research team in order to obtain more accuracy between the simulated noise and the results of the detrending process.

The selected methods are the following:

- Polynomial fitting

This is a well-known technique widely used [14]. Trend is fitted using squares regression, and then noise is obtained as the difference between experimental and predicted data (by the

regression model). In this case, the best results were obtained by using a commercial calculus package (Origin Pro 7 ®). Due to the limitations of the program, a ninth-order polynomial was utilized with the maximum order available.

- MAR-10

In this method, noise is computed using a moving average [15]. More precisely, if m_n represents the following moving average: $m_n = \frac{1}{21} \sum_{p=-10}^{10} x_{n+p}$, then noise is computed as the difference $x_n - m_n$ where x_n denotes the nth simulated value. Here, the size of the box (i.e., the one with 21 points) is the result of an optimization, analyzing different box sizes currently used in previous works.

- Butterworth filtering

This method uses analogical filters [8], and it is part of the MATLAB Signal Processing Toolbox software (Matlab® Signal Processing Toolbox©). The Butterworth filter is a type of signal processing filter designed to have as flat a frequency response as possible in the passband. The filter returns the transfer function coefficients of an nth-order low-pass digital Butterworth filter with specified cutoff frequency Wn. This methodology is a commonly used trend removal method, also analyzed by Bertocci and Huet [13], among others.

- MICS

This method, named MICS (Median of Intervals and Cubic Splines) [16], is a trend removal technique that we proposed and extensively tested in a previous work [8]. The methodology is based on dividing the data set into intervals, computing the median position and the average time. The resulting points are considered as the nodes for determining a cubic spline. The noise is obtained by calculating the difference between the simulated data and the cubic spline interpolation.

In this first approach, MICS showed the best training performance. Nevertheless, due to low relative errors in case of trend that become high in case of noise because of their different magnitudes, it should be observed that in most of these methods, the relative error level is acceptable for trend but usually unacceptable and useless in case of noise.

Therefore, a different approach was needed and our next attempt was directed to the use of artificial neural networks (ANNs) for detrending.

4. Using neural networks

From the very beginning, ANNs were mostly employed on function approximation [17, 18]. From this viewpoint, trend removal can be seen as a particular case of function approximation. ANNs are also widely used as a statistical analysis tool.

The main idea of our second approach was to apply an ANN methodology to improve the trend removal procedure. Moreover, our domain (environmental corrosion) is essentially

dynamic, due to high variance of functional trends, so it seems to be especially adequate the use of ANNs because of their inherent dynamic nature. Due to a big quantity of both patterns and output data, we preferred feed-forward ANNs that have proved to be especially useful in function approximation. According to Universal Approximation Theorem [19], feed-forward ANNs of three or more layers with backpropagation algorithm can approximate every continuous function (such as functions of Section 2.1), so it is particularly recommended for use in our case.

Anyway, we implemented these functions into the tool in order to make possible the "self-training" of the ANN with the generated data. Indeed, the effort is well worth the investment because this approach has the advantage of allowing the self-training along with more exhaustive testing.

With simulated data, backpropagation ANNs were trained splitting the signal in intervals and then training three different neural networks, depending on the mean derivative of each segment. Once done, a data testing set was generated for cross-validation.

The implemented backpropagation ANN is a supervised-learning neural network, which means that real results are provided with the training data set. Then, the ANN applies an algorithm based on the descendent gradient in order to minimize the mean square error (MSE)—the classical error measure in these cases. In the input layer, we began with 50 data (i.e., 50 neurons, one for each input data) although in this case we obtained poor convergence so we later reviewed this parameter, deciding to increase resolution several times until we fixed it in 400 input neurons which proved to be the best.

About the topology used, according to Funahashi's Universal Approximation Theorem, one hidden layer (three-layered ANN) is enough to approximate the noise function [20, 21]. Anyway, for performance reasons, we added a second hidden layer in the tuning phase—a practice recommended in especially complex problems [22]. In order to accelerate convergence and avoid oscillations, learning rate was fixed at 0.1 and a threshold was implemented in the backpropagation ANN. A momentum factor was also added and fixed at 0.3 [23].

Considering that we want to utilize this method to approximate different functions, the whole range was not used as an input vector [24]. This aspect is critical, since the ANN may memorize the input function, and the methodology would not be useful when working with other trend lines. For this reason, an interval pre-processing was applied, consisting of a moving window, frequently used in other cases like temporary series. We used a 40-data interval, where the original vector was split into 10 vectors of 40 data each. These 40-data vectors were then used as inputs for the ANN. **Figure 1** shows a graphical representation of the procedure decrypted.

By this approach, the ANN improved. Many optimization techniques had been suggested for feed-forward, one of them was the use of an All-Class-One-Network (ACON) structure that is to say, using many sub-nets depending on data entries. In our case, we used three sub-nets of equal topology but with different weight matrix. In the input layer, we loaded 40-data vectors in one of the three sub-nets depending on the media signal derivative on the interval (we
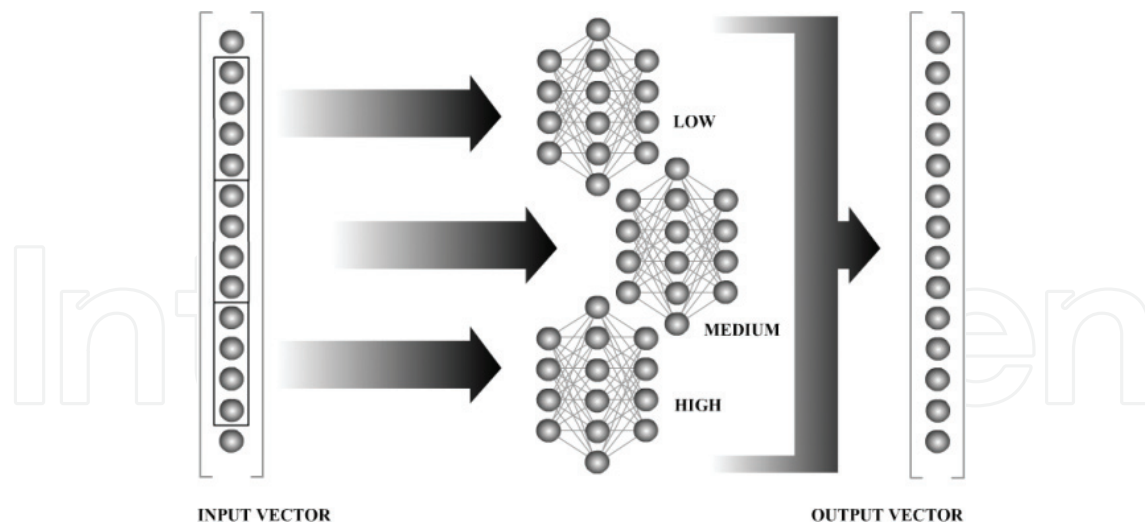
Figure 1. Split and join of signal segments.

divided the function range in three sectors, namely low, medium, and high derivative using finite differences). At the end, the obtained vectors in the output layer are joined into one vector of 400 data elements, comprising the final solution (see **Figure 1**). Different values of learning rate and momentum were evaluated during the training phase. As expected, oscillations were observed for learning rates higher than 0.3. The final value was fixed at 0.15, resulting in little oscillations when the ANN advances uniformly to the solution. Moreover, with the momentum factor added, close to a local minimum there are more steps—and also little ones—and the convergence accelerates when a local minimum is still far.

Then, the ANN was integrated with the data generators to complete the tool. The data inputs and outputs were implemented with text files or spreadsheets [24].

4.1. Training process

At first, in the training process, we used a single expression for the noise, training the ANN continuously with it. In this first phase, we could prove the ANN memorizing ability as a first step. Obviously, with single noise data, the ANN could approximate to any desired error level. In the second phase, we trained the net with a Lorentz function with fixed parameters but with random noise (inverse Gaussian distributed). We loaded the ANN with different noise values at each step, according to real conditions. In this phase, data files were saved after 10, 100, 1000, 5000, 10,000 and 30,000 iterations. These results showed a certain oscillation level but with a clear media and standard deviation-descendent direction of the mean squared error (MSE). Based on the second phase, it can be observed that with 30,000 training epochs, a 0.2% trend relative error and a 7% noise relative error are quite good when considering its random function and unpredictability. In a third and final phase, we randomized both noise and Lorentz function parameters into a determined interval according to real conditions [24]. Thus, with the most exhaustive testing, we found that relative errors were increased—as expected— being still acceptable in comparison with other methods used before.

4.2. Validation

Several statistical measures are useful for ANN validation: media and standard deviation are primary metrics. More information can be obtained with kurtosis and PSD. Also, cross-validation [19–25] was not used with the classical procedure of splitting data in training and testing sets but generating a testing set once the training phase concluded, as it was always possible to obtain fresh data. An online cross-validation was also applied, as we have already done in previous ANNs works [26], although using new testing data each step. With the optimizations applied, the convergence was good, although it could be improved with second-order [27] or even genetic algorithms [6–28].

4.3. Results obtained with ANN detrending

Once the implementation of the ANN and the generators was done, we applied the self-training of the tool with the generated data. Since Lorentz function is randomly parameterized and later perturbed with random noise (inverse Gaussian distributed), a certain oscillation level takes place, although it tends to stabilize after the 10,000 epochs. **Figure 2** shows the MSE value during the training process.

Figure 2 shows media and standard deviation of MSE values (in groups of 10) ,are shown in **Figure 3**.

According to the prediction power of the ANN, this methodology succeeds in isolating noise with an acceptable relative error level (see **Figure 4**).

As it can be observed, in **Figure 4**, the ANN had the ability to identify frequency variances and so distinguish noise from trend function. Nevertheless, the results must be checked with PSD, a suitable methodology for assessing trend removal methods.

The PSD at the end of training process, corresponding to 30,000 iterations, illustrated in **Figure 5**.

In order to prioritize inference over memorization, a suitable alternative is to perform earlier stop training. Indeed, at 1000 epochs, MSE remains low, with the advantage of improving ANN generalization ability. **Figure 6** shows the PSD at this level.
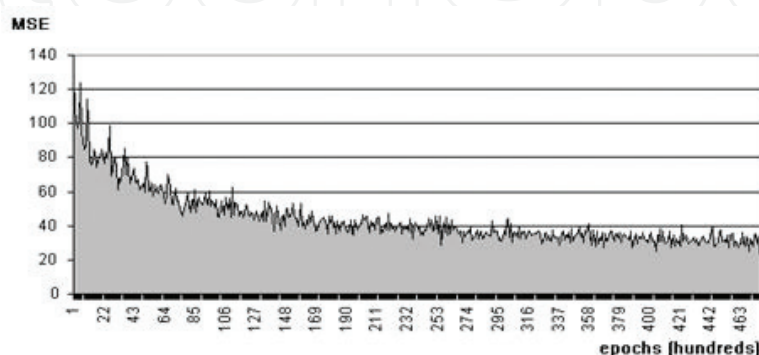

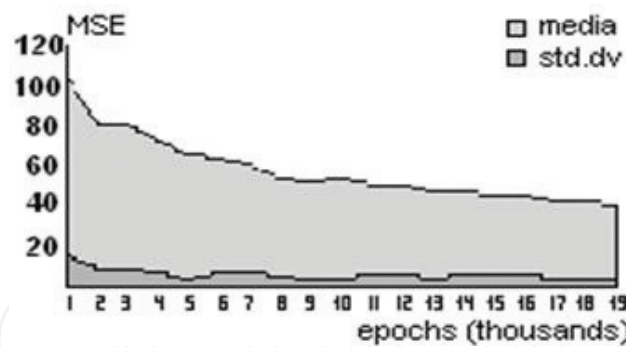**Figure 2.** Evolution of the MSE expressed in hundred of epochs.

Figure 3. Evolution of media and standard deviation of the MSE.

Analyzing **Figures 5** and **6**, we observe that they are similar in medium potential order. It can be noted that the difference between predicted and real noise appears only in low frequencies. At the beginning of the training process, detrending is partial. Also, it can be concluded that the most trained ANN obtains more concordance between real and predicted noise spectral. At 30,000 epochs, differences are about 10 Hz, which is remarkable.

The results of this second approach based on ANNs methodology, showed good predictions for the Lorentz's function trend line, superimposed with Gaussian noise. It was observed that,
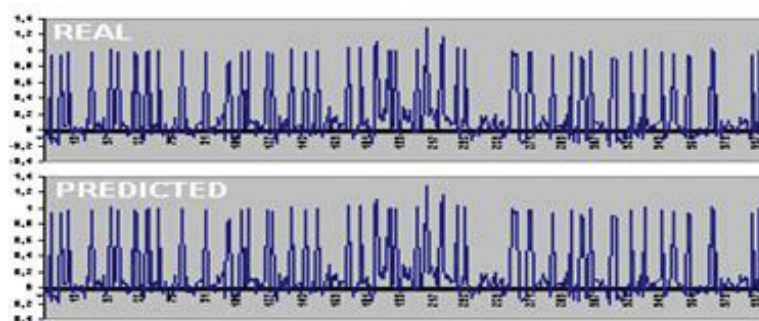

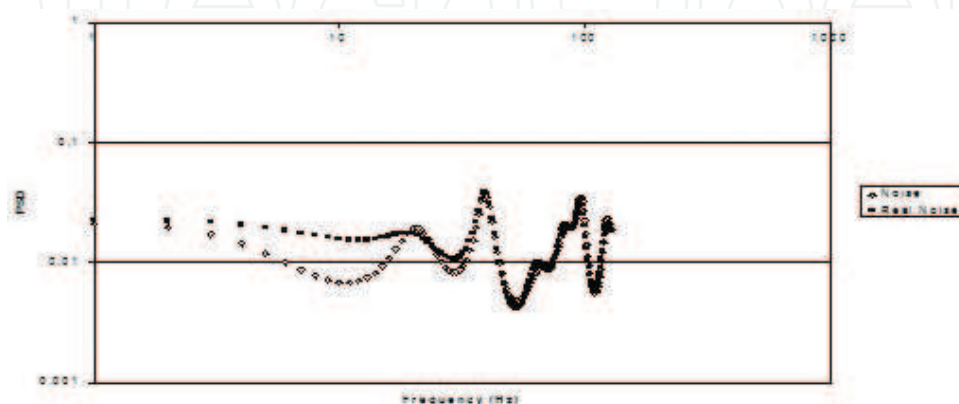**Figure 4.** Comparison between real and ANN predicted noise.

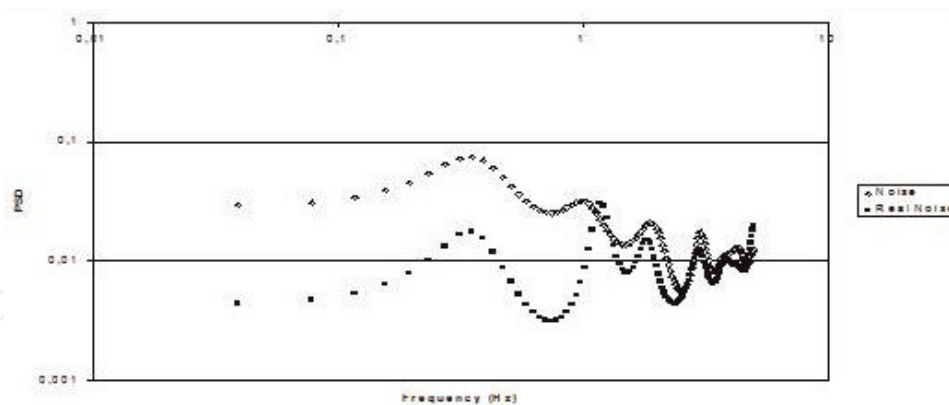**Figure 5.** PSD at 30,000 iterations.

Figure 6. PSD at 1000 iterations.

especially at high frequencies, the superimposed signal has no influence when performing the ANN detrending. The corresponding results and their validation by PSD were published in a previous paper [24].

Nevertheless, poor results were obtained when the degrees of freedom were increased. This situation was particularly observed in low frequencies, when a non-Gaussian-simulated noise was superimposed. Moreover, if the difference between trend and noise increases, the results show that the method loses accuracy. As a consequence, the use of ANNs with real experimental data (having an unpredictable trend shape) would need a set of training curves or a previous selection of the network to be applied. Thus, a different approach for detrending was needed.

Even more, when analyzing experimental data, it is obvious that a Gaussian noise can be considered only as a preliminary approximation, since real curves show a different kind of noise consisting of nonstationary pits superimposed on the trend line.

For these reasons, our third approach to the problem had to include a different noise simulation in order to get a better approximation to experimental real-life curves.

5. New simulated signals

As in the previous cases, the signals to be simulated needed to be generated using different nontrivial tendencies. For this purpose, curves 1 and 2 of Section 2.1 were reutilized, but in this case, a new computer-generated noise was superimposed.

In this case, noise employed was a discrete transient of exponential decay that simulates a nonstationary pit. More precisely, the noise consisted of a pulse train with an amplitude factor (A) corresponding to a uniform distribution of [0, 2.5). Initial time was obtained from a binomial distribution with a parameter $p=0.02$ and its sign had a binomial distribution with parameter $p=0.5$. As a consequence of these facts, pits can appear randomly in 2% of the points, and it can be either positive or negative with the same probability.

The transients are simulated by the pulse function: $f(t) = A\,B\,t\exp\left(-b\,(t-c)^d\right)$. The parameters b, c, and d were assigned to obtain a signal similar to the experimental transient. The A factor represented the transient amplitude and B determined the sign of the function. Taking into account the previous description, the simulated pit—if it is positive—grows fast and then decays exponentially as it usually does in experimental curves. A similar behavior is presented in negative nonstationary pits.

As a final remark, another characteristic of the simulated transients is the similarity of format, frequency, and amplitude relative to the real signal with respect to real transients.

Figure 7 shows a detailed shape of individual simulated transitory.

Then, two computer-simulated data sets were generated by adding the new noise—described in this section—to both trend lines presented in Section 2. A first signal was obtained by adding the new noise function to the first curve of Section 2.1, and the same procedure was followed to obtain a second simulated signal, adding noise to the second curve of Section 2.2. In all cases, the interval between data points was 0.7 s.

Figure 8 represents the simulated pulse train and signals, where parameters were fitted in order to obtain curves with similar power trend lines.

Both curves represent more realistic simulations, since their shapes and pits are similar to those observed in the laboratory (this fact can be observed in [30], where simulated and experimental curves were compared).

The simulated Signal 1 (New noise superimposed to Curve 1) was treated with polynomial, MMR10, and MAR10 methods. Other methodologies previously studied as MICS, Butterworth
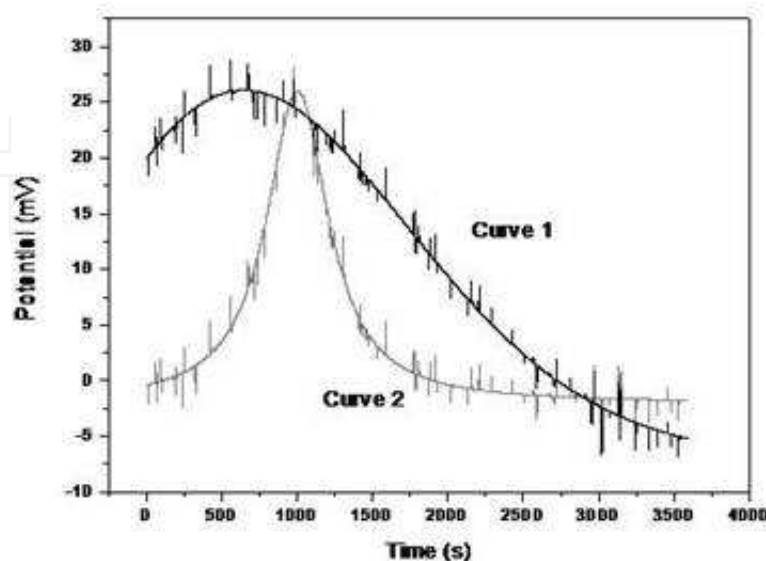

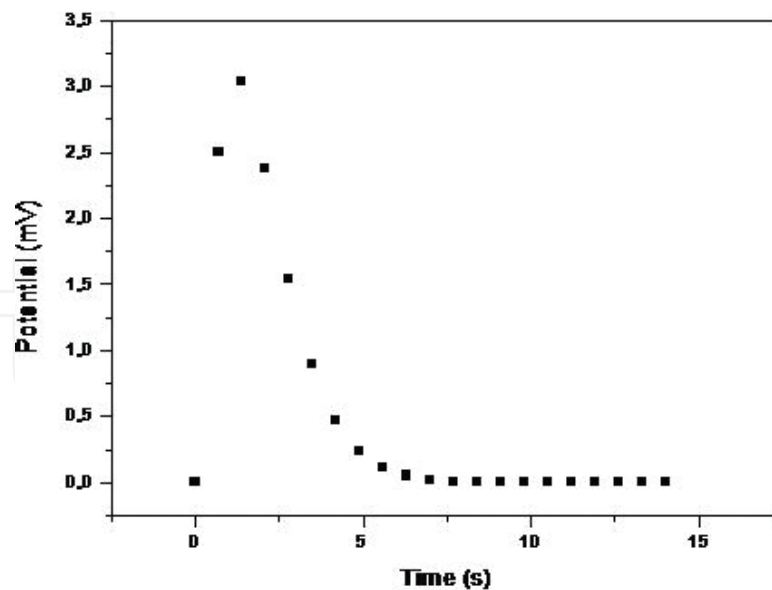**Figure 7.** The simulated transitory detailed.

Figure 8. Computationally generated signals, obtained by adding simulated noise data and two different trend lines. Curve 1 (black) and Curve 2 (grey).
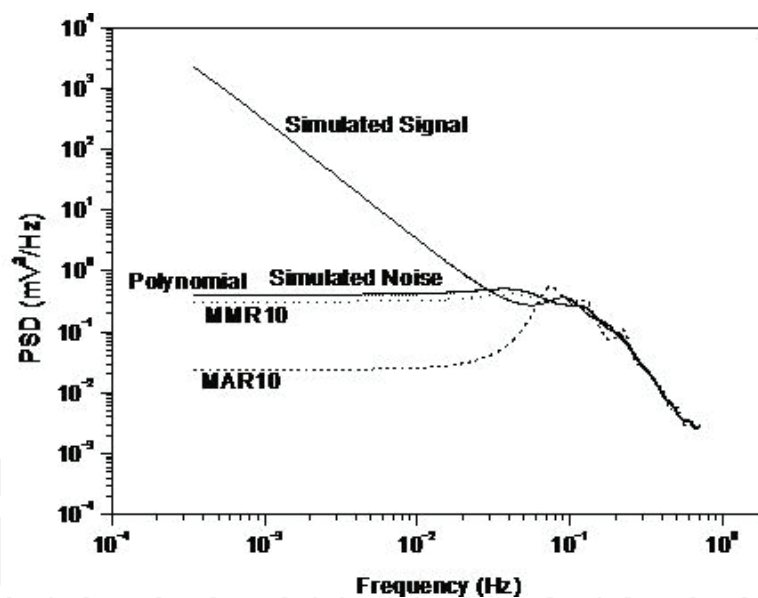

Figure 9. PSD of simulated Signal 1 (Noise + Curve 1), treated with polynomial, MMR10, and MAR10 methods.

filters, and ANNs were not considered at this stage as their results were very poor when they were assessed by using this new signal.

Figure 9 shows simulated Signal 1 and noise spectra, obtained by the maximum entropy method (MEM −15 order, ENAnalize® program).

Likewise, simulated Signal 1 (New noise superimposed on Curve 2) was treated with the same three detrending methods: polynomial approximation, MMR10, and MAR10. As in the
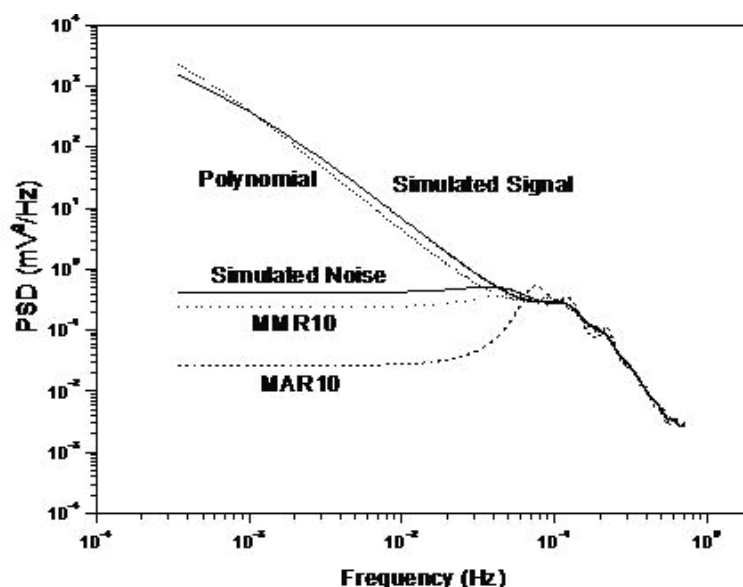
Figure 10. PSD of simulated Signal 2 (Noise + Curve 2) treated with polynomial, MMR10, and MAR10 methods.

previous case, MICS, Butterworth filters, and ANNs showed very poor results when working with this new signal, and so they were not considered.

Figure 10 shows the results corresponding to Signal 2.

The power spectrum obtained depends on the order of the MEM method applied. A smooth spectrum is observed when the MEM order is small, whereas the spectrum appears much noisier with a high order. The comparison between the spectra obtained by applying different detrending methods is easier when using a relatively low-order MEM [3]. More accurate results are provided by Fast Fourier Transform (FFT), in spite of the spectra obtained being noisy. Then, the comparison of PSD results in the same graphic is not plainly depicted. It is not possible to compare the PSD results for FFT with the MEM results, and, for this reason, they are not included in **Figures 9** and **10**.

6. A comparison of the performance of different detrending methods

The aim of this new approach was to analyze the performance of several detrending methods, when assessed by using the new signals with simulated nonstationary pits instead of Gaussian noise [29].

Due to their performance with these new simulated data, the selected methods were polynomial detrending, moving average removal (MAR), and moving median removal (MMR). The first two were described before and the third one can be considered similar to the second one, except for the substitution of the moving average by the moving median. In MMR-10 method, the noise is computed as $x_n - k_n$, whereas x_n denotes an experimental value and k_n represents the moving median: $k_n = median \ [x_{n-10}, \dots, x_n, \dots x_{n+10}]$.

		Signal 1	Signal 2
MAR-10	Mean	2.49×10^{-4}	-2.93×10^{-4}
	Standard deviation	0.3204	0.3204
MMR-10	Mean	8.50×10^{-3}	5.90×10^{-3}
	Standard deviation	0.3600	0.3550
Polynomial	Mean	-1.39×10^{-5}	4.13×10^{-3}
	Standard deviation	0.3735	1.8261

Table 1. Mean and standard deviation for trend removal methods: MAR-10, MMR-10, and ninth-order polynomial fitting, under simulated pulse train noise (mean: -0.0112, standard deviation: 0.3736).

The simulated noise data set had a mean of -0.0112 and standard deviation of 0.3736. **Table 1** shows the statistical results for the trend removal methods employed for simulated Signal 1 and Signal 2.

The noise mean value reported was close to zero with all the methods performed (**Table 1**), so it is not possible to conclude about the performance of detrending methods considering only this parameter.

- Ninth-order polynomial fitting

In the case of Signal 1 (with a smoother trend), **Table 1** shows that the standard deviation is similar to the one of the original noise, while a greater standard deviation was obtained for Signal 2.

This behavior is confirmed by the spectra analysis, since the polynomial methodology does not remove the power in the low-frequency region for Signal 2. Moreover, the spectrum obtained for Signal 1 is a good representation of the trace of the original simulated noise. Thus, this detrending method strongly depends on the simulated trend curve, and it is not reliable as a pre-processing method.

- MAR-10

This method does not depend on the trend curve utilized (**Table 1**). Indeed, the data standard deviation showed a good agreement with the simulated noise. Examining the spectrum, we can affirm that at high frequencies the noise obtained fits well with the original noise. On the other hand, for frequencies below 0.01 Hz, there was a maximum at ca. 0.07 Hz and a plateau was reached, with smaller power values than the original noise. A frequency breakpoint value was obtained after processing the original noise spectrum data with MAR compared to the one predicted by the interval obtained by using the transfer function. Bertocci et al. [13] presented the MAR-p transfer function, as can be seen below:

$$H_{MAR}(f) = 1 - \frac{1}{2p+1} \frac{\sin\left([2p+1]\pi \frac{f}{f_s}\right)}{\sin\left(\pi \frac{f}{f_s}\right)}, \text{ being } f_s \text{ the sample frequency.}$$

The spectrum is not attenuated between $f_{max}/(2p + 1)$ and f_{max}. The highest representative frequency is the Nyquist frequency ($f_{max} = f_s/2$). In the case considered, the interval used was between 3.4×10^{-2} and 0.7 Hz.

In sum, a lower standard deviation was obtained by the spectrum recovered by MAR-10 than the one related to the original data.

- MMR-10

MMR and MAR methods were robust (considering the type of signal processing), when compared with polynomial fitting. MMR standard deviation was much closer than the one obtained by MAR respect to the original values. A lower difference between the original value and the one reached by MMR was obtained for Signal 1. The MMR-10 spectrum fits better with the original noise than the one obtained by MAR-10 in all the frequency range. These results were discussed in-depth in a previous paper [29].

The mean is the most commonly used statistic measure of location in corrosion processes. However, extreme values will have great influence on it. The average may not be the most appropriate location measure if there are outliers in the sample, that is, if one or more values are much larger or smaller than the others [30].

On the one hand, data must be ordered increasingly for calculating a median, and for a large data set, this is a time-consuming task and so calculating medians are computationally more demanding than calculating averages. On the other hand, medians remain unaffected by a small group of outliers [29, 30]. This is a very important characteristic for approximating the trend in an ENA data set. For these reasons, MMR provides a better baseline than MAR, since the subtraction of medians preserves the signal trace, showing less attenuation than the corresponding subtraction of the average.

7. Conclusion

As observed, the method used for trend removal affected the results obtained. An ideal methodology should not introduce external effects, and at same time, it should recover most of the signal information.

The polynomial fitting method showed a strong dependence when different trends were considered. This methodology presented good results for smooth tendencies and poor performances when curves changed suddenly in relation to their slope and convexity.

When working with different curves, MAR presented robust results. Nevertheless, alterations in the low-frequency zone of spectra were observed, and as a consequence of this fact, the standard deviation results were not accurate.

Besides, MMR, our proposed method, demonstrated a better performance when working with different simulated signals, although in the low-frequency zone of spectra, it showed the same behavior as MAR.

Taking into account the results obtained with simulated data, it is difficult to arrive to definitive conclusions. As it was observed, the polynomial fitting method is not reliable for trend removal at least when working with ENA data. On the other hand, methods like MAR and MMR achieve better results, but they do not offer a definitive solution. In fact, the results suggest that the application of these methods produces an apparently attenuated response with low noise signals.

Hence, MMR method seems to show the best results. Nevertheless, increasing the order of magnitude, this method did not improve its performance, and as previously observed, calculating the median is computationally more troublesome than calculating the average of a given data set.

As it was shown with the new simulated signals and, despite all the studied methods have been useful in many particular cases, a really accurate procedure for trend removal is still an open problem, giving new opportunities for further research.

Acknowledgements

Author details

Victor Martinez-Luaces* and Mauricio Ohanian

*Address all correspondence to: victorml@fing.edu.uy

Faculty of Engineering, UdelaR, Montevideo, Uruguay

References

[1] Almeida E, Mariaca L, Rodriguez A, Uruchurtu J, Veloz M. Characterization of prerusted steels in some Ibero-American atmospheres by electrochemical potential noise measurement. In: Kearns J, Scully J, Roberge P, Reichert D, Dawson J, editors. Electrochemical Noise Measurement for Corrosion Applications. West Conshohocken, USA: ASTM International; 1996. pp. 411-426. DOI: 10.1520/STP37974S

[2] Dawson J. Electrochemical noise measurement: The definitive in-situ technique for corrosion applications? In: Kearns J, Scully J, Roberge P, Reichert D, Dawson J, editors. Electrochemical Noise Measurement for Corrosion Applications. West Conshohocken, USA: ASTM International; 1996. pp. 3–38. DOI: 10.1520/STP37949S

[3]   Cottis R. Interpretation of electrochemical noise data. Corrosion. 2001;**57**(3):265-285. DOI: 10.5006/1.3290350

[4]   Sachs L. Applied Statistics: A Handbook of Techniques. 2nd ed. New York: Springer-Verlag; 2012. 707 p. DOI: 10.1007/978-1-4612-5246-7

[5]   Hetch-Nielsen R. Neurocomputing. Massachusetts: Addison-Wesley; 1990. 433 p

[6]   Font C, Manrique D, Ríos J. Redes de Neuronas Artificiales y Computación Evolutiva. Madrid: Universidad Politécnica de Madrid; 2005. 239 p

[7]   Koza J, Rice J. Genetic generation of both the weights and architecture for a neural network. In: Proceedings of Seattle International Joint Conference on Neural Networks (IJCNN-91); 8-12 July 1991; Seattle IEEE; 1991. pp. 397-404

[8]   Ohanian M, Martinez-Luaces V, Guineo G. Highly dispersed electrochemical noise data: Searching for reasons and possible solutions. The Journal of Corrosion Science and Engineering. 2004;**7**:preprint 6

[9]   Ralston A, Rabinowitz P. First Course in Numerical Analysis. 2nd ed. New York: Dover Publications; 2001. 606 p

[10] Chan K, Hayya J, Ord J. A note on trend removal methods: The case of polynomial regression versus variate differencing. Econometrica: Journal of the Econometric Society. 1977;**45-3**:737-744. DOI: 10.2307/1911686

[11] Mansfeld F, Sun Z, Hsu C, Nagiub A. Concerning trend removal in electrochemical noise measurements. Corrosion Science. 2001;**43**(2):341-352. DOI: 10.1016/S0010-938X(00)00064-0

[12] Tan Y, Bailey S, Kinsella B. Factors affecting the determination of electrochemical noise resistance. Corrosion. 1999;**55**(5):469-475. DOI: 10.5006/1.3284009

[13] Bertocci U, Huet F, Nogueira R, Rousseau P. Drift removal procedures in the analysis of electrochemical noise. Corrosion. 2002;**58**(4):337-347. DOI: 10.5006/1.3287684

[14] Ohanian M, Caraballo R, Dalchiele E, Guineo-Cobs G, Martínez-Luaces V, Quagliata E. Productos de corrosión formados en ambiente marino: Vinculación de variables estructurales y potencial de corrosión. Revista de Metalurgia. 2005;**41-3**:175-185. DOI: 10.3989/revmetalm.2005.v41.i3

[15] Ohanian M. Corrosión Atmosférica: Caracterización estructural y electroquímica de productos de corrosión formados sobre acero de baja aleación en intemperie marina [Thesis]. Montevideo, Uruguay: Universidad de la República; 2004

[16] Faires J, Burden R. Numerical Methods. 4th ed. Boston: Brooks/Cole; 2012. 608 p

[17] Hilera J, Martínez V. Redes Neuronales Artificiales: Fundamentos, Métodos y Modelos. Editorial Alfaomega: México; 2000. 390 p

[18] Stergiou C, Siganos D. Neural Networks [Internet]. 2010. Available from: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html [Accessed: 2017-12-26]

[19] Haykin S. Neural Networks: A Comprehensive Foundation. New Jersey: Prentice-Hall; 1999. 842 p

[20] Martinez Luaces M. Metodologías en Redes Neuronales: Diseño, Entrenamiento y Validación de Perceptrones Multi-Capa. In: Proceedings of COMPUMAT; La Habana; 2003

[21] Martinez Luaces M, Martinez-Luaces V. Teachers assessment: A comparative study with neural networks. In: Proceedings of DELTA Conference (DELTA'03); 23–27 November 2003; Queenstown. Dunedin: ISC-DELTA; 2003. pp. 180-185

[22] Hassoun M. Fundamentals of Artificial Neural Networks. Massachusetts: MIT Press; 1995. 511 p

[23] Picton P. What is a neural network? In: Introduction to Neural Networks. London: Palgrave; 1994. pp. 1-12. DOI: 10.1007/978-1-349-13530-1_1

[24] Martinez Luaces M, Martinez-Luaces V, Ohanian M. Trend-removal with neural networks: Data simulation, preprocessing and different training algorithms applied to electrochemical noise studies. In: Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases; 15-17 February 2006; Madrid. Madrid: WSEAS; 2006. pp. 810-817

[25] Freeman J, Skapura D. Neural Networks: Algorithms, Applications and Programming Techniques. Massachusetts: Addison-Wesley; 1991. 401 p

[26] Salvetto P, Martínez Luaces M, Luna C, Segovia J. A very early estimation of software development using neural network. In: Proceedings of X Congreso Argentino de Ciencias de la Computación (CACIC'10); 4-8 October 2004; San Justo. Buenos Aires: CACIC

[27] Isasi P, Galván I. Redes de neuronas artificiales: Un enfoque práctico. Pearson Education: Madrid, Spain; 2004. 229 p

[28] Ventura D, Andersen T, Martinez T. Using evolutionary computation to generate training set data for neural networks. In: Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms; 19-21 April 1995; Alès, France. Berlin: Springer; 1995. pp. 468-471

[29] Ohanian M, Martinez-Luaces V, Diaz V. Trend removal from electrochemical noise data. The Journal of Corrosion Science and Engineering. 2010;**13**:preprint 52

[30] Utts J, Heckard R. Mind on Statistics. 5th ed. Cengage Learning: Stamford; 2014. 682 p