

Reconstruction of periodic signals using neural networks

Reconstrucción de señales periódicas usando redes neuronales

JOSÉ DANILO RAI RÁN ANTOLINES

Electrical Engineer, Ph.D. Candidate in Engineering at Universidad Nacional de Colombia. Professor at the Universidad Distrital Francisco José de Caldas. Bogotá, Colombia.

Contact: drairan@udistrital.edu.co

Received: July 3, 2012

Accepted: May 21, 2013

Classification: Research

Funding: Universidad Distrital Francisco José de Caldas

Keywords: backpropagation, frequency measurement, Fourier series, harmonic analysis.

Palabras clave: propagación hacia atrás, medida de frecuencia, series de Fourier, análisis de armónicos.

ABSTRACT

In this paper, we reconstruct a periodic signal by using two neural networks. The first network is trained to approximate the period of a signal, and the second network estimates the corresponding coefficients of the signal's Fourier expansion. The reconstruction strategy consists in minimizing the mean-square error via backpropagation algorithms over a single neuron with a sine transfer function. Additionally, this paper presents mathematical proof about the quality of the approximation as well as a first modification of the algorithm, which requires less data to reach the same estimation; thus making the algorithm suitable for real-time implementations.

RESUMEN

En este artículo se reconstruye una señal periódica por medio de 2 redes neuronales. La primera es entrenada para aproximar el periodo de la señal, y la segunda sirve para estimar los coeficientes de la expansión de Fourier correspondientes. La estrategia consiste en minimizar el error medio cuadrático por medio del algoritmo de propagación hacia atrás sobre una neurona con función de transferencia senoidal. Además, en este artículo se presenta la prueba matemática acerca de la calidad de la aproximación, así como una primera modificación del algoritmo, la cual permite alcanzar el mismo resultado con menos datos, así entonces el algoritmo podría ser utilizado en aplicaciones que requieran tiempo real.

* * *

1. INTRODUCTION

Period estimation is important in signal processing. A classical result in signal processing, the Sampling Theorem, asserts that reconstruction of a periodic target signals up to order n harmonic frequencies from sampling data requires $2n$ data points. Under these conditions, the Discrete Fourier Transform (DFT) computes the period of a sampled signal, but only if a multiple of the sampling rate matches the period of the signal. No other results are known to be applicable otherwise. The signal is generally impossible to reconstruct exactly without *a priori* knowledge of the period.

The most important challenge in approximating the period is the fact that a sampling may not match the period; moreover, in general that is not the case. For example, if a sampling is recorded every 12 hours, is well known that the difference between that sampling and an integer number of years will eventually be in the order of seconds. This is because the rotation of the earth does not match the translation of the earth. The most common adjustment of this difference is the additional day every leap-year. The problem is that the ratio between rotation and translation for the earth may not be a rational number, and this has tragic consequences on prediction. The irrational ratio between sampling and period is considered negligible by current algorithms to approximate periodic systems, such as Fourier Transform or polynomial approximations, but it is evident that a prediction in the long term will be influenced by any error in the approximation.

The frequency estimation has been focused on single sinusoidal signals in [1] - [3]. In [1], authors use an estimation method called phase unwrapping, which attempts to estimate the frequency by performing a linear regression on the

phase of the signal. This method is not useful in real-time applications due to the complexity of the algorithm. The method in [2] is a deterministic method in noiseless conditions, and it is based on the solution of a system of linear equations using two DFTs. The method in [3] is iterative, starting from a matrix of elements that are converted to two unitary vectors, which are optimally combined to give the final frequency estimate. Other works analyze signals with several sinusoidal components. For instance, in [4] the estimate of the frequency for a power system is obtained by minimizing the squared error between an assumed signal model and the actual signal. In [5], authors suggest a time-varying sinusoidal representation to estimate frequency for signals such as speech. Experimental results show that, although is computationally intensive, the algorithm outperform FFT-based frequency approaches in nonstationary environments. Finally, a more general family of algorithms uses DFT as a basis to estimate the frequency. In [6], authors propose a method exclusively for complex exponential waveforms under white noise. In [7], a noniterative method is consisting of two parts: a coarse estimation, given by the FFT, and a fine estimation, using least square minimization of three spectrum lines.

In this paper, we take a discrete approach to solve the signal reconstruction problem, based purely on data from observations of the signal at a fixed sampling rate. We provide one method to get a good approximation of the period, and we also provide some guarantees about the quality of the approximation. The algorithm proposed here are thus capable of estimating the period for a target function with unknown period, such as the Van der Pol nonlinear oscillator or a Duffing oscillator.

The organization of the paper is as follows: In Section 2, we define the problem in terms of general-

izing a periodic dynamical system. In Section 3, a Fourier Neural Network is design to reconstruct the whole dynamics of a given periodic system. In Section 4, we present the main result of this paper. We say that approximating the fundamental frequency is enough to reconstruct a periodic behavior. In Section 5 the mathematical proof of the algorithm is addressed. Finally, some discussions about the results, including a way to reduce the amount of data required for the algorithm, and some conclusions are presented in Section 6 and 7, respectively.

2. METHODOLOGY

In the first step of developing the approximation algorithm a set of neural networks were tested based on preliminary results given by [8]. The idea was to evaluate the learning capabilities of the neural networks having in mind the generalization that they can perform based on data. Since neural networks have been broadly used to generalize, they look like good candidates to solve the estimation problem for periodic signals. The process to train the neural network is the next:

1. The data set is presented to the neural network;
2. The weights of the network are changed by a learning algorithm in such a way that the network memorizes the samples that are used during the training. As a result, the network in addition to memorize also generalize;
3. Finally, the network is simulated with synthetic data. An example of the simulation step is shown in figure 1.

The target function in figure 1 is a perfect sine wave with period 2π and amplitude 1. Once the network has been trained, the output of the sys-

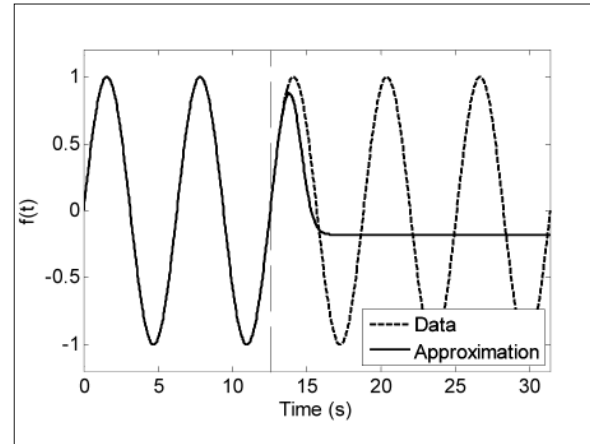


Figure 1. Prediction of the dynamic given by an Elman neural network

Note: The training data set goes from 0 s through 4π s.

Source: own work.

tem has perfectly learned the data, which is in this case a set of samples from 0 to 4π seconds. The prediction looks well for a quarter of a period ahead, but as soon as the time at what the prediction is made overcomes that threshold the output of the system does not resemble the dynamic anymore. The network make a very soft interpolation of the sampling, which in this case correspond to 20 points per period, however when the input of the network is any time outside the training set then the prediction goes wrong. This experiment was repeated for the same target function, a pure sine wave, because it supposes to be an easy target. A large number of architectures were used, such as radial basis networks, cascade-forward backpropagation network, layered-recurrent network, feed-forward backpropagation network, among many others. Also, a number of learning algorithms were tried, as well as different number of layers, and different neurons per layer. The conclusion is that a traditional neural network is not capable to generalize a periodic target function, thus why another type of network is tried in the next section.

3. APPROXIMATION BY FOURIER NEURAL NETWORK

The literature review shows that one of the most common mathematical tools to deal with periodic functions is the Fourier expansion, which is an approximation used to rebuild curves based on sinusoidal signals. An associated tool is the Fourier transform, which is a function that takes functions in the time domain and converts them into the frequency domain, as is indicated in equation (1). The transformation has some advantages especially about the arithmetic, for instance derivatives can be seen as multiplications of the function by angular frequency (in the frequency domain). This property makes the transformation ideal to solve linear differential equations. The transformation is also used to observe some aspects of the signal that are hidden in the time domain or that are hard to see, such as harmonic content.

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{2\pi i \omega t} dt \quad (1)$$

A result that comes from the Fourier transform is exclusively applied to periodic functions. This says that a periodic function can be approximated by a sum of sinusoidal signals with appropriated coefficients and frequencies, as shown in equation (2).

$$f(t) \approx \frac{a_0}{2} + \sum_{m=1}^n (a_m \cos(m\omega t) + b_m \sin(m\omega t)) \quad (2)$$

The target function is $f(t)$, and $a_0/2$ is the mean value of the signal, ω is the angular frequency in rad/s. The frequency of the sinusoidal components in equation (2) are multiples of ω . The first value of $m\omega$, i.e., ω is called fundamental frequency, the second one, 2ω , is named second harmonic, and so on for other multiples. The coefficients a_m and b_m are computed as shown in equation (3) and equation (4)

$$a_m = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(m\omega t) dt \quad (3)$$

$$b_m = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin(m\omega t) dt \quad (4)$$

The result of the approximation of a sawtooth function using 128 harmonics is shown in figure 2. The symmetry of the function makes all coefficients a_m zero. The first part of the figure indicates the number of the harmonic in the x axis, and the value of the coefficient in the y axis. The intermediate figure shows every harmonic plotted in the time domain; the biggest signal is the fundamental signal with amplitude equal to two. The lower part of the figure is the sum of all the harmonics, and that is the approximation of the target function.

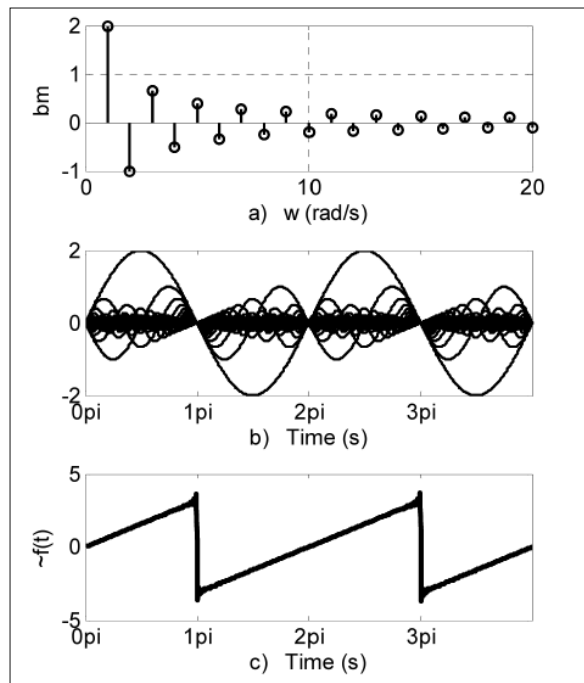


Figure 2. An approximation of a sawtooth function using Fourier expansion

Note: a) first 20 harmonics, b) individual sinusoidal waves at multiples of the fundamental frequency, c) approximation of $f(t)$.

Source: own work.

The work in [8] says that neural networks are good candidates to approximate dynamical systems. Three examples are presented in which the feedforward neural networks are used to approximate autonomous systems. The second work proves that a neural network with a unique neuron in the hidden layer can learn the Fourier transform of any function [9]; however nothing is said about the Fourier expansion. The third work says that a sinusoidal activation function improves the generalization of a neural network and this result is applied in problems called statistically neutral [10]. Those problems can be always reduced into two variables, and both variables have the same probability of occurrence. According to the work in [11], the coefficients of the Fourier expansion can be learned by a Hopfield neural network.

References [9], [10] and [11] are particularly important, because they show that the implementation of a feedforward neural network do learn a Fourier expansion. For the current application, that network has one input, the time, and one output, the prediction of the function value at the time indicated by the input. The network has only one hidden layer and the activation function is sinusoidal. This setup guarantees that the weights of the network, between input and hidden layer, are multiples of the fundamental frequency, whereas the weights between hidden layer and output are the coefficients of the Fourier expansion. The activation function of the output neuron is a linear function. Bias signal for every neuron is fixed on zero to reproduce the sine terms of the expansion, and $\pi/2$ for the terms associated with cosine. Bias of the output neuron is the average value of the function. The mathematical form of the neural network is shown in equation (5).

$$f^*(t) = b_0 + \sum_{n=1}^{\infty} a_n \cdot \sin((\omega_n \cdot t) + b_n) \quad (5)$$

The comparison between the Fourier neural network in equation (5) and the Fourier expansion in equation (2) shows that b_0 is equivalent to $a_0/2$; also ω_n to $m\omega$; and b_n to 0 or $\pi/2$; finally, a_n in equation (5) is equivalent to the coefficients a_n y b_n in equation (2).

Once the architecture of the Fourier neural network has been defined, as is indicated in equation (5), the remaining work is to train the net based on the data set. The chosen training algorithm is the modified backpropagation Levenberg-Marquardt. One of the best results is shown in figure 3.

The Fourier neural network in figure 3 is trained using 100 data points per period, and the target function is a sawtooth function. The training data covers two periods. The validation of the training is made using the next eight periods. The lower part in figure 3 shows the target function and

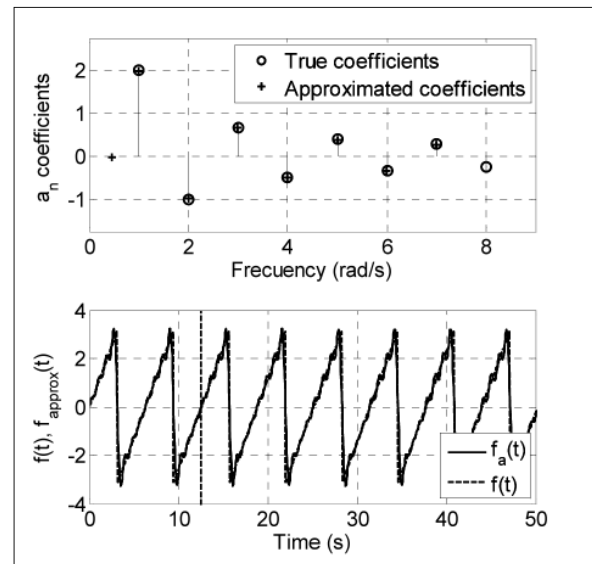


Figure 3. Output of a Fourier neural network after the training process

Note: The dashed line represents the end of the training data set.

Source: own work.

the approximation given by the network. The coefficients of the Fourier expansion as well as the approximations are plotted in the higher part in figure 3. The result in figure 3 looks promising but actually it is not. The prediction is very good for times smaller than or equal to the validation time window, but the prediction degenerates as the time increases. This problem is associated with the error allowed by the coefficients and multiples of the fundamental frequency.

The error on the parameters after the learning process has two sources: errors in magnitude, and errors in frequency. The analysis of these errors leads to say that errors on the frequency computation influence the approximation more than the errors on magnitude. A study of the deviation of the frequency is shown in figure 4. That example in figure 4a shows a target function compounds by two sinusoidal signals: one with unitary frequency and the other with frequency equal to 2 rad/s.

If the root mean square error $RMSE$ for the two functions $f_1(t)$ and $f_2(t)$ is compared, is clear that the error is zero for $\omega_2 = 2$, and one else where. No matter how close the value of ω_2 is, the error is there, except for $\omega_2 = 2$. Then, if ω_2 is not the exact value then sooner or later the approximation will reach the maximum RMSE, and will come up with errors as bad as any other approximation. A derived result from figure 4.a is that the period of the combination $f_1(t)$ and $f_2(t)$ has a period that last about 10^4 seconds, and its value is two only when ω_2 is exactly equal to 2. Even though good approximations will remain good for more time than coarser approximations. Figure 4.c shows a window for the prediction, which is useful to define how long the approximation will be under certain RMSE, for example, in figure 4.c. that RMSE is 0,2 and the correspondent time is 493 s. Every approximation later than 493 s has RMSE bigger than 0,2.

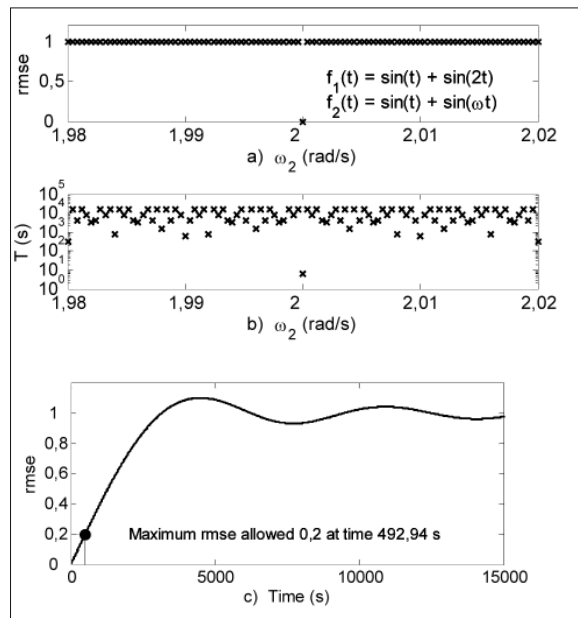


Figure 4. Error on the frequency computation

Note: a) rmse over a period for “w” variable, b) composed period for $f_1(t)$ and $f_2(t)$, c) time window under a rmse bound.

Source: own work.

4. APPROXIMATION OF THE FUNDAMENTAL FREQUENCY

The problem about approximating the fundamental frequency or its multiples using the Fourier Neural Network is that regardless of how close the estimation is the prediction gets worse and worse as the time goes. The next analysis has as a goal to answer why the learning algorithm, in this case backpropagation, causes that error. The answer is that the problem is precisely to minimize the RMSE, as shown in figure 5. To understand the problem let suppose that the data set has only two points $t = 0$ and $t = 2\pi$, and both are equal to one, because they come up from a square function with period 2π and amplitude equal to one. The analysis starts with the approximation given by only one harmonic, the fundamental frequency. The minimization of the RMSE is equivalent to find the sinusoidal that minimizes the distance between the data points and the harmonic.

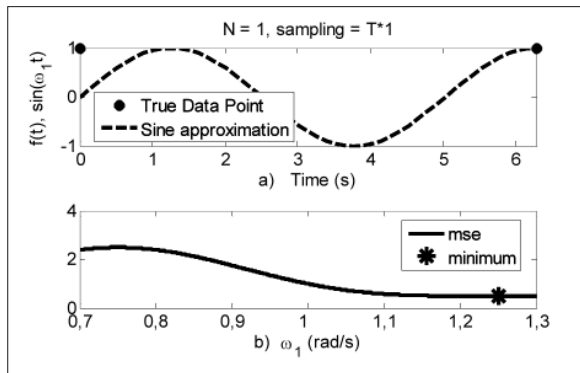


Figure 5. Root mean squared error for a couple of points approximated by a pure sine wave

Note: The data length is $N = 1$ period.

Source: own work.

The first observation coming out from the result in figure 5 is that it is not possible to fix the error at zero, due to the difference in value between the approximation and the data at $t = 0$; and that difference happens regardless of the number of data points in the sampling. The second observation is about the frequency that minimizes the error: that frequency is different than the true frequency, regardless of the number of data points, and it happens even if the sampling contains the true period. That is because backpropagation minimizes the RMSE, then no matter how finer the sampling

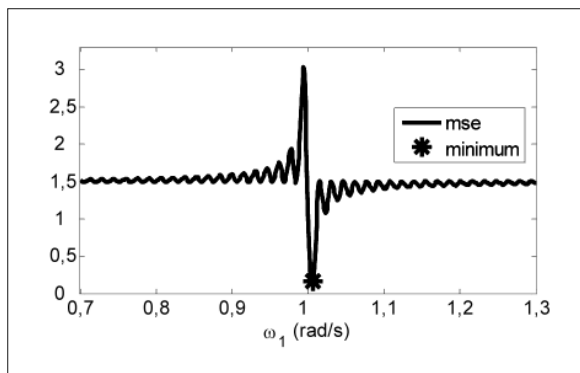


Figure 6. RMSE for a square function. 64 periods, sampling every T

Source: own work.

is the true frequency will not ever be reached, as shown by an asterisk in figure 5. For the next experiment the data set has a sampling rate of one data point per period across 64 periods, as it is shown in figure 6.

The good news about using more data is that the minimum gets closer to the true frequency value, so the minimization of the RMSE will get a better approximation if the data length is increased. Another characteristic is that the sampling does not need to match the true period, as shown in figure 7. The value 0,9 in figure 7 means that the sampling has been taken every 0,9 periods. The frequency that minimize the RMSE using just a couple of data points is within 10% above and under the true value, in addition, as the quantity of data increases the maximum error shrinks, and makes the estimation closer to the true frequency. All the curves tend to 1 rad/s as the number of data points grows, it is the true frequency.

There are two problems that come from approximating the frequency if that is made by minimizing RMSE. The first one is that the error depends on the distribution of the samplings, in other words, different target functions require different amounts of data to get the same approximation; and the second problem is the learning algorithm

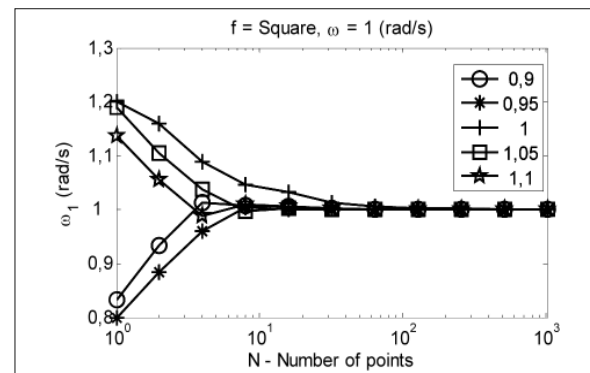


Figure 7. Approximation of the frequency at different sampling rates

Source: own work.

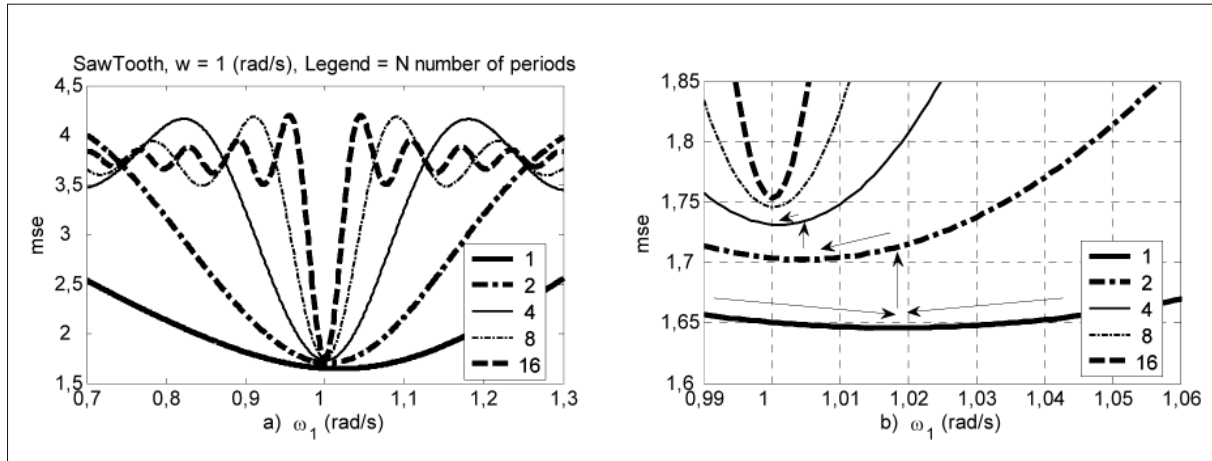


Figure 8. Successive approximations of the fundamental frequency

Note: a) RMSE for different length of data, b) approximation algorithm strategy.

Source: own work.

itself, because it makes a search of the local minima from an initial guess and gets stuck at a local minimum that is not necessarily the desired one, so the initial estimation to start with the minimization is quite critical. The second problem is detailed in figure 8 for a sawtooth target function. All cases were simulated using 15 points per period; the lower curve represents RMSE when the data set has 15 points, it is one period, and subsequent curves are simulation for an increasing data set at the same sampling rate, so the length of the next curve is two periods, 4, 8, and finally 16 for the highest curve.

The number of local minima in figure 8 is incremented as the data increases, as is shown also in figure 6. Then, the strategy starts with few data points, where the number of local minima is low. Then, the data set is enlarged iteratively, and once a local minimum is reached for the first iteration, the data is incremented, for instance, to the double. In the next step the initial frequency for the enlarged data set is the same frequency that minimized RMSE in the first stage of the algorithm. This algorithm is repeated as far as it is necessary or data available.

5. MATHEMATICAL FACTS ABOUT THE APPROXIMATION

What is next is to prove mathematically that the minimization process already described certainly provides an approximation of the fundamental frequency, regardless of what the target function is. The root mean squared error between a function $f(t)$ and a sinusoidal wave with amplitude a_1^* and frequency ω^* in discrete time is given in equation (6):

$$mse_n = \frac{\sum_{i=1}^n (f(t_i) - a_1^* \sin(\omega^* t_i))^2}{n} \quad (6)$$

Lemma 1

The minimization of the RMSE does not result in the fundamental frequency, regardless of the number of data points used to approximate the period.

The mean squared error in continuous time is defined in equation (7).

$$mse = \frac{1}{T} \int_0^T (f(t) - a_1^* \sin(\omega^* t))^2 \cdot dt \quad (7)$$

When $n \rightarrow \infty$, then $mse_n \rightarrow MSE$, as Riemman sum says. In particular T in equation (7) can be $Tl = (2\pi)/\omega^*$. If the function $f(t)$ is periodic, and in order to simplify the algebra, $f(t)$ can be shifted back, so the first component of the Fourier expansion is completely sinusoidal. From now on, and looking for simplicity, the proof will be made for target function that can be written as a sum of sinusoidal waves exclusively. The Fourier expansion in equation (7) can be written as is shown in equation (8).

$$mse = \frac{1}{Tl} \int_0^{Tl} \left(\sum_{j=1}^{\infty} a_j \sin(j\omega t) - a_1^* \sin(\omega^* t) \right)^2 \cdot dt \quad (8)$$

The computation of the integral is described in equation (9).

$$mse = -\frac{1}{4} a_1^{*2} \frac{\sin(2\omega^* Tl)}{\omega^* Tl} + \sum_{j=1}^{\infty} \left(-a_j a_1^* \frac{\sin((j\omega - \omega^*) Tl)}{(j\omega - \omega^*) Tl} + a_j a_1^* \frac{\sin((j\omega + \omega^*) Tl)}{(j\omega + \omega^*) Tl} + \Delta_j \right) \quad (9)$$

The variable Δ_j represents all the independent terms of ω^* . The derivative $dmse(\omega^*)/d\omega^*$ is computed in equation (10) to find the frequency ω^* that minimize the root mean squared error.

$$\begin{aligned} \frac{dmse(\omega^*)}{d\omega^*} &= \frac{1}{4} a_1^{*2} \frac{\sin(2\omega^* Tl)}{\omega^{*2} Tl} - \frac{1}{2} a_1^{*2} \frac{\cos(2\omega^* Tl)}{\omega^*} + \dots \\ &+ \sum_{j=1}^{\infty} \left(a_j a_1^* \frac{\cos((j\omega - \omega^*) Tl)}{(j\omega - \omega^*)} - a_j a_1^* \frac{\sin((j\omega - \omega^*) Tl)}{(j\omega - \omega^*)^2 Tl} + \dots \right. \\ &\left. + a_j a_1^* \frac{\cos((j\omega + \omega^*) Tl)}{(j\omega + \omega^*)} - a_j a_1^* \frac{\sin((j\omega + \omega^*) Tl)}{(j\omega + \omega^*)^2 Tl} \right) \quad (10) \end{aligned}$$

Since $Tl = (2\pi)/\omega^*$, then in equation (11):

$$\begin{aligned} \frac{dmse(\omega^*)}{d\omega^*} &= -\frac{1}{2} \frac{a_1^*}{\omega^*} + \sum_{j=1}^{\infty} a_j \left(\frac{\cos((j\omega - \omega^*) Tl)}{(j\omega - \omega^*)} - \frac{\sin((j\omega - \omega^*) Tl)}{(j\omega - \omega^*)^2 Tl} + \dots \right. \\ &\left. + \frac{\cos((j\omega + \omega^*) Tl)}{(j\omega + \omega^*)} - \frac{\sin((j\omega + \omega^*) Tl)}{(j\omega + \omega^*)^2 Tl} \right) \quad (11) \end{aligned}$$

The limit of $\omega^* \rightarrow \omega$. Using l'Hôpital's rule in equation (12) and equation (13) is:

$$\begin{aligned} \lim_{\omega^* \rightarrow \omega} \frac{dmse(\omega^*)}{d\omega^*} &= -\frac{1}{2} \frac{a_1^*}{\omega^*} + \sum_{j=1}^{\infty} a_j \left(\frac{\sin((j\omega - \omega^*) Tl)}{1} + \frac{\sin((j\omega - \omega^*) Tl)}{2Tl} + \dots \right. \\ &\left. \dots + \frac{-\sin((j\omega + \omega^*) Tl)}{1} + \frac{\sin((j\omega + \omega^*) Tl)}{2Tl} \right) \quad (12) \end{aligned}$$

$$\lim_{\omega^* \rightarrow \omega} \frac{dmse(\omega^*)}{d\omega^*} = \lim_{\omega^* \rightarrow \omega} \left(-\frac{1}{2} \frac{a_1^*}{\omega^*} \right) = -\frac{1}{2} \frac{a_1^*}{\omega} \quad (13)$$

Since the limit of $\omega^* \rightarrow \omega$ for $dmse(\omega^*)/d\omega^*$ is different than zero, then the minimum MSE shows up at $\omega^* = \omega$, when $T = Tl$. So, we can conclude that minimizing the MSE, the ω^* will not be ω at $T = Tl$.

Lemma 2:

In the limit when $T \rightarrow \infty$ the frequency $\omega^* \rightarrow \omega$

The value of Tl from equation (11) will be replace for Tz to simplify the analysis, so $Tz = Z^* Tl + Tl/8$, and Z is the set of natural numbers.

$$\begin{aligned} \frac{dmse(\omega^*)}{d\omega^*} &= \frac{1}{4} a_1^{*2} \frac{1}{\omega^{*2} Tz} + \sum_{j=1}^{\infty} \left(a_j a_1^* \frac{\cos((j\omega - \omega^*) Tz)}{(j\omega - \omega^*)} - \dots \right. \\ &\dots - a_j a_1^* \frac{\sin((j\omega - \omega^*) Tz)}{(j\omega - \omega^*)^2 Tz} + a_j a_1^* \frac{\cos((j\omega + \omega^*) Tz)}{(j\omega + \omega^*)} - \dots \\ &\left. \dots - a_j a_1^* \frac{\sin((j\omega + \omega^*) Tz)}{(j\omega + \omega^*)^2 Tz} \right) \quad (14) \end{aligned}$$

To simplify equation (14) in equation (15) the l'Hôpital's rule is applied to finde the limit when $\omega^* \rightarrow \omega$.

$$\begin{aligned} \lim_{Tz \rightarrow \infty} \left(\lim_{\omega^* \rightarrow \omega} \frac{dmse(\omega^*)}{d\omega^*} \right) &= \left(\lim_{\omega^* \rightarrow \omega} \left(\frac{1}{4} \frac{a_1^*}{\omega^{*2} Tz} + \dots \right. \right. \\ &\dots + \sum_{j=1}^{\infty} a_j \left(\frac{\sin((j\omega - \omega^*) Tz)}{-1} + \frac{\sin((j\omega - \omega^*) Tz)}{2Tz} + \dots \right. \\ &\left. \left. \dots + \frac{-\sin((j\omega + \omega^*) Tz)}{1} + \frac{\sin((j\omega + \omega^*) Tz)}{2Tz} \right) \right) \quad (15) \end{aligned}$$

$$\lim_{Tz \rightarrow \infty} \left(\lim_{\omega^* \rightarrow \omega} \frac{dmse(\omega^*)}{d\omega^*} \right) = \lim_{Tz \rightarrow \infty} \left(\frac{1}{4} \frac{a_1^*}{\omega^2 Tz} \right) = 0 \quad (16)$$

The conclusion is that ω^* in equation (16) gets closer a closer to the true ω when $Z \rightarrow \infty$.

Lemma 3:

The number of local minima of the RMSE around the fundamental frequency increases as the length of data set increases.

Let remember equation (11), but instead of TI let use NTI in equation (17) and equation (18), where N is a natural number.

$$\begin{aligned} \frac{dmse(\omega^*)}{d\omega^*} = & -\frac{1}{2} \frac{a_1^*}{\omega^*} + \sum_{j=1}^{\infty} a_j \left(\frac{\cos((j\omega - \omega^*)NTI)}{(j\omega - \omega^*)} \dots \right. \\ & \dots - \frac{\sin((j\omega - \omega^*)NTI)}{(j\omega - \omega^*)^2 NTI} + \frac{\cos((j\omega + \omega^*)NTI)}{(j\omega + \omega^*)} \dots \\ & \left. \dots - \frac{\sin((j\omega + \omega^*)NTI)}{(j\omega + \omega^*)^2 NTI} \right) \quad (17) \end{aligned}$$

$$\lim_{NTI \rightarrow \infty} \frac{dmse(\omega^*)}{d\omega^*} = -\frac{1}{2} \frac{a_1^*}{\omega^*} + \frac{a_1 \cos((\omega - \omega^*)NTI)}{(\omega - \omega^*)} \quad (18)$$

Since $TI = 2\pi/\omega^*$, and the minimum for RMSE shows up at $dmse(\omega^*)/d\omega^* = 0$, then in equation (19):

$$\lim_{NTI \rightarrow \infty} \frac{dmse(\omega^*)}{d\omega^*} = 0 = \frac{1}{2} \frac{a_1^*}{a_1} \left(1 - \frac{\omega}{\omega^*} \right) + \cos\left(2\pi N \frac{\omega}{\omega^*} \right) \quad (19)$$

The result to remark so far happens when $N \rightarrow \infty$, for this case the frequency of the cosine term grows linearly with N , so the number of local minima increases. The minimization of the MSE using backpropagation will get stuck at the closest local minimum from the initial point of the search, and that initial frequency not necessarily is the one that makes the algorithm to halt at the minimum with a value closest to the true frequen-

cy. The limit of $NTI \rightarrow \infty$ is $\omega^* \neq \omega$, as it is shown in equation (19).

6. RESULTS

The result of the three lemmas can be extended for an approximation of the second harmonic, in other words, when the minimization of the MSE estimates the double of the fundamental frequency. The intuitive idea is that a signal of period T is also periodic for an integer multiple of T , then if the data set has two periods, the approximation will provide two outputs, one close to T and another one close to $2T$, and so on for any number of periods. The motivation for this intuitive idea is that instead of making approximations with one pure sinusoidal signal it is possible to use another signal, also periodic, with a shape like the function that is going to be approximate. The similarity between signals can be understood as the smallest distance, using any metric, between the harmonics of the true target and the approximant function.

If the number of terms for the approximant function is one, as it is shown in equation (20), the function is a sinusoidal wave, and its amplitude is one, in the same way that has been shown so far. If m is 2 then the approximation has 2 sinusoidal waves, and one has the double frequency of the other one, and so on.

$$f^*(\omega) = \sum_{m=1}^n \frac{1}{m} \sin(m\omega t) \quad (20)$$

The difference between using only a sinusoidal wave and several harmonics looks subtle, but the result is remarkable. The number of data points required to make any approximation is reduced as it is shown in figure 9. For the experiment in figure 9 the sampling rate is $1,01T$ of a square signal, and the data set has only four data points. The x axis corresponds to the number of harmo-

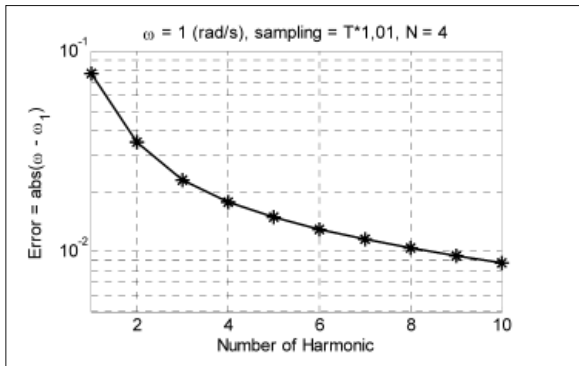


Figure 9. Effect of harmonics on the approximant function

Note: N = number of periods in the data set.

Source: own work.

tics, n , and the result indicates that increasing the number of harmonics, leaving the number of data points fixed, improves the approximation. When the multiples of a sinusoidal wave are included in the approximant, the size of the expression in equation (20) increases and this has a computational cost; however the requirements for the length of the data sets are reduced. It seems that combining the two strategies is a good way to deal with the approximation problem. The procedure requires to increase the number of data points per period, and to increase the number of periods in the data set, and also to increase the number of harmonics in an iterative process in order to avoid undesirable local minima.

Once the frequency has been approximated, the approximation is used to train the Fourier neural network as it is shown in equation (5). The fundamental frequency and its multiples will be given by the approximation algorithm; in the same way as it is shown in figure 3. The network is the perceptron shown in figure 10. That architecture has an additional advantage: the phase angle that was required to learn in equation (5) is not required anymore, so the new goal is exclusively to learn the coefficients of the Fourier expansion using the weights of the network. By definition the coeffi-

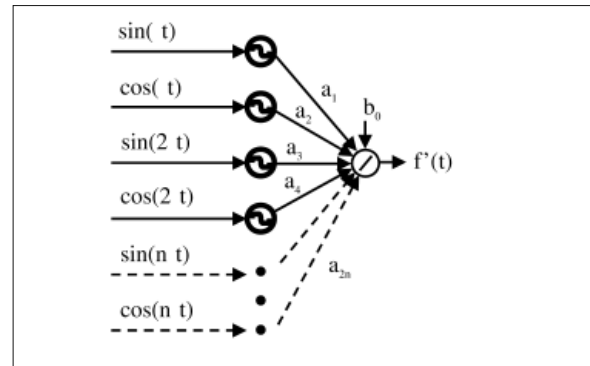


Figure 10. Perceptron architecture to learn the Fourier expansion

Source: own work.

cient of the Fourier expansion minimizes the mean square error, so running backpropagation, which minimizes MSE, is an ideal way to find the coefficient.

Experiments show that the modified backpropagation algorithm called Levenberg-Marquardt gets stable after just *one* epoch; however there is an error on the value of the coefficients, they are just approximations. Even though, increasing the number of data points per period, it is refining the sampling improves the approximation. Results of the estimation of a triangular function are shown in figure 11. The approximation of the frequency used for running this experiments is 1,0001, and the true value is 1 (rad/s).

The prediction in figure 11 is very close to the true target for low values of time; and even after 500 seconds, which are 500 periods of the function, the approximation is relatively good; even though the error is visible at times like 1000 s. The approximations for other three types of function are shown in figure 12. The results are better for smooth functions like the oscillator, because of the effect well known as Gibbs phenomenon; this condition predicts problems around the discontinuities.

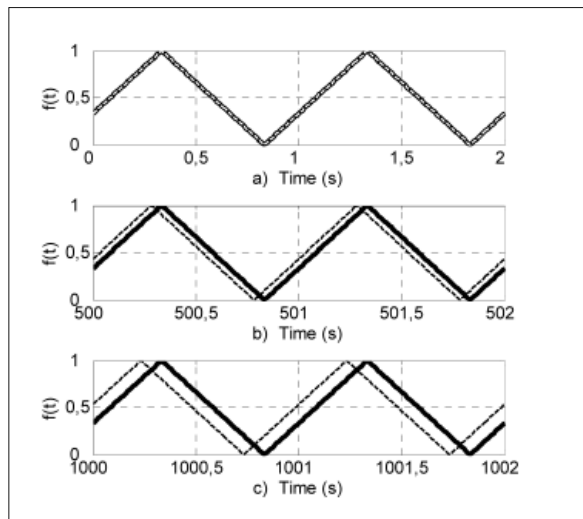


Figure 11. Prediction of the the value for a triangular function

Note: Phase shift = 60° , $T = 1$ s, $\omega_1 = 1,0001 \cdot 2\pi$ (rad/s). Continuous line = Target function, dashed line = approximation.

Source: own work.

7. CONCLUSIONS

The reconstruction of periodic signals can be performed by approximating the period of the signal, which can be seen also as approximating the fundamental frequency. This frequency is estimated using a single neuron with a sine transfer function and backpropagation as training algorithm. The approximation leads to an inevitable error that can be minimized only increasing the amount of data in an iterative fashion. This paper presents an algorithm that smartly uses data samples, evenly spaced, to approximate the period of a signal trough successive iterations. The first iteration requires a rough approximation of the period and data points to cover at least the guessed period, and then the minimization of the mean squared error provides a first estimation for the period. The estimation at the end of each iteration is used as the initial guess for the next iteration, at the same time, the data length is progressively

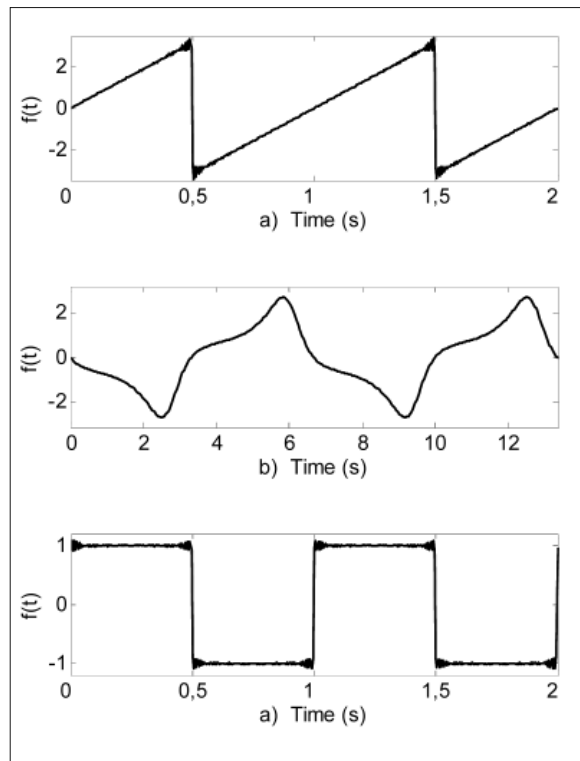


Figure 12. Prediction for three functions

Note: a) Sawtooth, b) Van Der Pol, c) Square function.

Source: own work.

increased every iteration, which guarantee that the next iteration will provide a better approximation than the previous one. This paper includes the mathematical proof about the quality of the estimation, and in addition proposes a first modification of the algorithm to reduce the amount of data required to make the estimation, which is by means of approximating not only the fundamental frequency, but also the harmonics. Once the period is estimated, then a perceptron network can be used to approximate the Fourier coefficients, which serves to reconstruct the entire dynamic of a periodic system. Future works could study ways to reduce the quantity of data require for the algorithm, and also ways to validate the results of the algorithm for quasi periodic signals.

REFERENCIAS

- [1] R. G. McMillan, B. G. Quinn, I. V. L. Clarkson, and B. Moran, "Frequency estimation by phase unwrapping", *IEEE Trans. on Signal Processing*, vol. 58, no. 6, pp. 2953-2963, Jun. 2010.
- [2] S. Provencher, "Estimation of complex single-tone parameters in the DFT domain", *IEEE Trans. on Signal Processing*, vol. 58, no. 7, pp. 3879-3883, Jul. 2010.
- [3] H. C. So, and F. K. W. Chan, W. Sun, "Subspace approach for fast and accurate single-tone frequency estimation", *IEEE Trans. on Signal Processing*, vol. 59, no. 2, pp. 827-831, Feb. 2011.
- [4] R. Chudamani, K. Vasudevan, and C. S. Ramalingam, "Real-time estimation of power system frequency using nonlinear least squares", *IEEE Trans. on Power Delivery*, vol. 24, no. 3, pp. 1021-1028, Jul. 2009.
- [5] Y. Pantazis, O. Rosec, and Y. Stylianou, "Iterative estimation of sinusoidal signal parameters", *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 461-464, May. 2010.
- [6] C. Candan, "A method for fine resolution frequency estimation from three DFT samples", *IEEE Trans. on Signal Processing Letters*, vol. 18, no. 6, pp. 351-354, Jun. 2011.
- [7] C. Yang, and G. Wei, "A noniterative frequency estimator with rational combination of three spectrum lines", *IEEE Trans. on Signal Processing*, vol. 59, no. 10, pp. 5065-5070, Oct. 2011.
- [8] J. M. Rodríguez, and M. H. Garzón. "Neural Networks can learn to Approximate Autonomous Flows", *Fluids and Waves: Recent Trends in Applied Analysis*, University of Memphis, TN, vol. 440, 2007, pp. 197-205.
- [9] A. Menon, K. Mehrotra, C. K. Mohan, and S. Ranka, "Characterization of a Class of Sigmoid Functions with Applications to Neural Networks", *Elsevier Science Neural Networks*, vol. 9, no. 5, pp. 819-835, Jul. 1996.
- [10] A. A. Ghorbani, and K. Owrangh, "Stacked generalization in neural networks: generalization on statistically neutral problems", in *Proc. of the International joint conference on Neural Networks IJCNN 2001*, Washington USA, 2001, vol. 3, pp. 1715-1720.
- [11] M. Karam, M. S. Fadali, and K. White, "A Fourier/Hopfield neural network for identification of nonlinear periodic systems", in *Proc. of the 35th Southeastern Symposium on System Theory*, Morgantown, WV, USA, 2003, pp. 53-57.