UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Visión Electrónica
Más que un estado sólido
http: revistas.udistrital.edu.co/ojs/index.php/visele/index

VISIÓN ELECTRÓNICA
A RESEARCH VISION

A Fourier based algorithm to estimate the period of a sampled signal

Un algoritmo para estimar el periodo de una señal muestreada basado en la transformada de Fourier

José Danilo Rairán Antolines[1]

INFORMACIÓN DEL ARTÍCULO

Historia del artículo:
Enviado: 07/09/2016
Recibido: 22/10/2017
Aceptado: 01/11/2017

Keywords:
Period estimation
Physiological signals
Power grid frequency
Discrete Fourier Transform

Palabras clave:
Estimación del periodo
Señales fisiológicas
Frecuencia de un sistema de potencia
Transformada Discreta de Fourier

ABSTRACT

Given a sampled signal, in general, is not possible to compute its period, but just an approximation. We propose an algorithm to approximate the period, based on the Discrete Fourier Transform. If that transformation uses data length for multiples of the true period, some of its harmonics have null value. Thus, the best candidate to be a multiple of the period minimizes the value of those harmonics. The validation for noiseless data shows an upper bound in the error equal to a quarter of the time between two consecutive samples, whereas the result for noisy data demonstrates robustness. As application, the algorithm estimates the period of physiological signals, and tracks the frequency of the power grid in real time, which evidence its versatility.

RESUMEN

Dada una señal muestreada, en general, no es posible calcular su periodo, sino solo una aproximación. En este artículo se propone un algoritmo para aproximar el periodo, basado en la Transformada Discreta de Fourier. Si esa transformación utiliza datos por un múltiplo del número de periodos, algunos de sus armónicos resultan nulos. Así, el mejor candidato a ser un múltiplo del periodo es el que minimiza el valor de esos armónicos. La validación para datos sin ruido muestra un límite máximo para el error de un cuarto del tiempo entre dos muestras consecutivas, mientras que el resultado para señales con ruido demuestra robustez. Como aplicación, el algoritmo es utilizado para estimar el periodo de una señal fisiológica, y el seguimiento de la frecuencia de un sistema de potencia, en tiempo real, lo cual evidencia la versatilidad del algoritmo.

[1]PhD. In Engineering - Systems and Computing, Universidad Nacional de Colombia, Colombia. Current position: Professor, Universidad Distrital Francisco José de Caldas, Colombia. E-mail: drairan@udistrital.edu.co.

Citar este artículo como: J. D. Rairán. "A Fourier based algorithm to estimate the period of a sampled signal". Visión Electrónica, algo más que un estado sólido, Vol. 11, No. 2, 152-160, july-december 2017. https://doi.org/10.14483/22484728.13133

1. Introduction

Signal processing requires computing the period of a function, for instance to reconstruct a signal, which finds applications in fault detection, physiological signal analysis, and astronomy, among other areas. However, in general, is impossible to know the true period, because the sampling process carried out to record data loses information of the signal, which worsens given conditions such as noise and quasi- periodicity, as is the case in real life. Thus, instead of looking after the true period, the best strategy corresponds to pick up a candidate period (an estimate) using an algorithm.

Some estimation algorithms address the case of sinusoidal waves. For instance, authors in [1] use a method called phase unwrapping. This method uses a linear regression, applied to the phase of a signal, in order to estimate its frequency. But the time required to come up with an approximation makes the algorithm inadequate for real-time applications. Authors in [2] guarantee an estimation for noiseless signals, given a method that computes the Discrete Fourier Transform (DTF) at two instants in time, followed by the solution of a linear system of equations. Another method, in [3], provides an approximation trough an iterative procedure applied to two unitary vectors.

Other proposals, besides the pure sinusoidal wave, allow signals with harmonic content. For instance, authors in [4] minimizes the mean quadratic error between a signal and its mathematical idealization in order to estimate the frequency of a power grid. Unfortunately, that proposal does not contrast the algorithm performance with other estimators. Authors in [5] suggest the sinusoidal representation of a dynamic signal in order to approximate the frequency of an audio signal. Results show that the algorithm, though computationally intensive, surpasses the approaches based on Fourier Transformation for dynamical scenarios.

Finally, a broader family of algorithms uses DFT to estimate the frequency. One of them, the method in [6], deal with complex exponential signals. Authors assure that the estimation process provides accurate estimations for slow or medium sampling rates, but say nothing about fast sampling rates. Authors in [7] propose a method to run in two steps. The first, a coarse estimation, and then a fine estimation.

On the other hand, the proposed algorithm in this paper handles signals with harmonic content. The algorithm starts with a set of candidate periods. The best approximation among those candidates minimizes an index that comes from the application of DFT for data covering multiples of an initial estimation. The evaluation of the performance for the algorithm includes a comparison with three algorithms, one of them the periodogram, which may be the most common algorithm to approximate a period to the point that has become a standard estimator.

The rest of the paper is organized as follows. Section 2 presents the algorithm, including the pseudo code. Section 3 shows the analysis of the error for noisy and noiseless data. Section 4 details two applications: the estimation of physiological signals, and the tracking of frequency for a power grid. Finally, Section 5 presents conclusions and future work.

2. Algorithm definition

This section presents the intuitive idea of the algorithm, followed by a formal definition, the pseudo code, and finally an example of application. The proposed algorithm -denoted ΔH - approximates a period for signals formed by a finite number of harmonics. The core of the method resides in the observation of some of those harmonics, when its computation includes data for multiple periods instead of one.

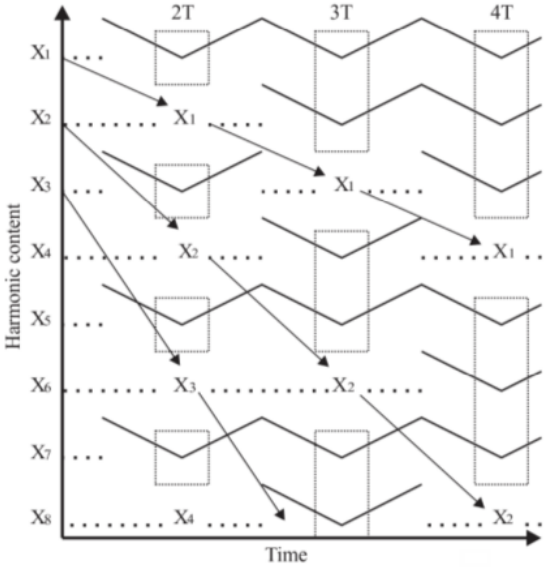
2.1. Algorithm approach

The periodical functions in this paper, $f(t+T) = f(t)$, allow sampling at constant rate, Ts, as well as the computing of its Discrete Fourier Transform. Given this set of functions, the explanation of the intuitive idea implies the analysis of four observations. The first observation starts by looking at the value of the first harmonic, $x1$, for data covering one period, which reappears at the second harmonic, $x2$, when the data covers two periods, as shown in Figure 1. This figure also shows that the value of $x1$ reappears in the third harmonic when data covers three periods. Thus, the value of $x1$ reappears in the xn harmonic if the data covers n periods. Similarly, the second harmonic, $x2$, reappears in the position $2n$ when data covers n periods. In general, the harmonic xi reappears at the position $n \times i$ when data covers n periods.

The second observation emphasizes the appearance of gaps in the DFT, which result from the jumping of the harmonics. For instance, a first gap in $x1$ show up when the data length equals two periods, because the value of $x1$ moves to $x2$ leaving an empty place at $x1(2T)$. This gap corresponds to a dashed rectangle in Figure 1. The interesting result of this fact is that

DFT fills each gap with a zero. In other words, and for an integer ratio T/Ts, looking at the pattern of zeros that come from the application of DFT would be enough to find the true period. For instance, if the pattern equals 0 0 $x1$ 0 0 $x2$ 0 0..., then the true period equals the greatest time in the data divide by three. Unfortunately, this process demands an integer ratio T/Ts, which in general is impossible to guarantee.

Figure 1: Effect of the data length in the harmonics value.



Source: own.

Third observation regards the value of the gaps for data covering a time length different from, but close to, a multiple of the true period. The DFT for this scenario fills the gap with a number different from zero; however, approaching zero as the time comes to a multiple of the period. Thus, the value of a harmonic around a multiple of the period forms a valley (a "V" shape) with minimum at the multiple of the true period, as shown in Figure 1. This last feature establishes the essence of the proposed algorithm: the way to select the best candidate to be a multiple of the period consists in choosing the candidate that minimizes the value of the harmonic when looking at a valley.

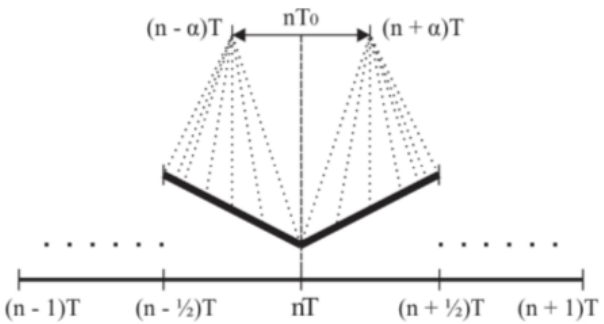
The fourth and last observation stands that instead of looking for a best candidate around a single valley, may be better to combine some of them before the minimization stage. The addition of valleys preserve both the location and the value of a minimum, but increases the steepness of the resulting valley facilitating the minimization. Adding up valleys has a double effect: the positive that the influence of the errors in a particular valley decreases, the negative that the algorithm requires more computations.

2.2. *Initial estimation and zone of search*

The ΔH algorithm requires three parameters from the user: an initial estimation of the period, T_0; the data length expressed in number of periods, n; and the number of valleys to compute the best approximation, nH. This section starts by defining the rank of values for the first parameter, T_0, which find its base in Figure 2. That figure shows a typical valley starting at $(n-1/2)T$ and finishing at $(n+1/2)T$, which guarantee the uniqueness of the minimum, around nT, if the search is confined to that region. Thus, the searching process analysis the candidate periods, where nT should be included, at both left and right of a particular value of nT_0.

Figure 2 shows the rank of values for nT_0, in particular for its extremes. For instance, the search of candidates for the end $nT_0 = (n-\alpha)T$ will go from $(n-1/2)T$ to nT, where α corresponds to an undefined new parameter. In other words, that search goes from the left end of the valley to the multiple of the true period, as indicated by the dashed lines in Figure 2. The search for the best estimate at the right end of nT_0, which equals $(n+\alpha)T$, goes from nT to an open end, which is not fixed yet in order to find a unique value of α.

Figure 2: Definition of the zone of search in a valley as a function of T_0.



Source: own.

Computing α requires the definition of a new factor, k_L, which multiplied by nT_0 generates the left end of the zone of search. Equation (1) shows the values for both ends of nT_0.

$$k_L(n-\alpha)T = (n-\frac{1}{2})T \qquad (1)$$
$$k_L(n+\alpha)T = nT$$

The value of α in equation (2) comes from eliminating k L in equation (1).

$$\alpha = 4n - 1 \tag{2}$$

Given the rank for $nT_0, (n-\alpha)T \leq nT_0 \leq (n+\alpha)T$, as seen in Figure **2**, and given the value of α in equation (2), the rank of values for T 0 is the indicated in equation (3).

$$\frac{(4n-2)}{4n-1}T \leq T_0 \leq \frac{4n}{4n-1}T \tag{3}$$

In particular, when searching around two periods, then $\frac{6}{7}T \leq T_0 \leq \frac{8}{7}T$. In other words, the error in initial estimation, T_0, should not go beyond $\pm\frac{1}{7}T$, which equals a maximum error of 14,3 %. Using more data (i.e. increasing the value of n), forces the algorithm to start with better initial estimates.

The determination of the zone of search, *zos*, follows the already explained definition of T_0. This zone corresponds to the rank of values from left to right of nT_0 where to look for the best candidate, nT', were T' is the estimation of the period. The left end of the *zos* is $k_L nT_0$, were k_L comes from equation (1). The right end of the *zos* is $k_R nT_0$, where k_R holds that $k_R(n-\alpha)T = nT$. Thus, the zone of search equals the expression in equation (4).

$$(n - \frac{1}{4})T_0 \leq zos \leq \frac{4n-1}{4n-2}nT_0 \tag{4}$$

The result in equation (4) depends on the initial approximation, T_0, and the number of periods under observation, n. For instance, if $n = 2$ the zone of search goes from $\frac{7}{4}T_0$ to $\frac{7}{3}T_0$, and increases in n shrinks the zone of search as happens with T_0.

Instead of defining the zone of search as function of the time, as described in equation (4), the implementation of the algorithm identifies the index of the ends for the *zos* in relation with the data vector. Thus, the search for the best candidate starts at the element with index N_i and finishes at the element with index N_f. For instance, if $n = 2$, the left end of the zos matches $N_i = |\frac{7}{4}\frac{T_0}{T_s}|$, whereas the right end equals $N_f = |\frac{7}{3}\frac{T_0}{T_s}|$. The floor and ceiling functions serves to guarantee an integer number for the indexes.

2.3. *Minimization process*

The search for a minimum inside the zos starts by defining a section of the data, which will be called D, from the first data point to the N data point, where $N_i \leq N \leq N_f$, (at least for now). Then, the decomposition of each $D(N)$ in harmonics, carried out using DFT, serves to compute the functional ΔH, which gives the name to the proposed algorithm. This ΔH functional, as defined in equation (5), add up nH valleys, where x_i represents the valley number i.

$$\Delta H(N) = \sum_{i=1}^{nH} x_i \tag{5}$$

The best candidate to be nT' corresponds to the time at which ΔH finds its minimum (i.e. $t(N')$), this according to the intuitive idea already explained. Thus, $nT' = t(N') = Ts(N'-1)$. The subtraction of one in the last expression makes the initial time equal to zero, $t(1) = 0$.

2.4. *Subsampling*

Searching across the whole zos may make the ΔH algorithm slow for real-time applications. For instance if $T_0 = 1$, $n = 2$, and $Ts = 1 \times 10^{-4}$s, then $Ni = 17{,}500$ and $Nf = 23{,}334$, which means running 5,834 computations of the functional ΔH. Instead of all those computations, the algorithm starts by subsampling the zos (a coarse-to-fine approach). Experiments with the subsampling rate showed the greatest reduction in computational time when the zos passes to 14 data points. It is important to remark that each of those 14 computations uses a whole section D, which means that the data remains whole.

The subsampling for the example in the previous paragraph means that the search starts at $Ni = 17{,}500$, followed by $N = 17{,}949$, then $N = 18{,}398$, and so on, until $Nf = 23{,}334$. One of those candidates minimizes ΔH, and then the algorithm continuous using a iterative process. A new zos goes from the left of the previous candidate to the right of it. For instance, if the best candidate was $N = 17{,}949$, then the new zos corresponds to $Ni = 17{,}500$ and $Nf = 18{,}398$. This new zos omits cases from 18,399 to 23,334, in other words, a saving of 4,935 out of 5,834 computations. The processes of subsampling and searching continues until the minimal different among the elements in zos reaches 1. This example passes from 5,834 required computations to a number of about 50. Details of the iterative procedure, including especial cases, are shown in Table **3**.

2.5. *Pseudocode*

In addition to the data, D, and the sampling time, Ts, the proposed algorithm has three inputs: an initial estimation of the period, T_0; the number of periods, n; the number of valleys, nH. The output of the algorithm is an estimation of the true period, T'. Given those inputs,

the algorithm defines an initial zone of search, by Ni and Nf as shown in Table 1. Then, an according to the size of the $(Nf - Ni)$, what may follow is the subsampling.

Table 1: Core of the Algorithm.

Inputs: T_0, n, nH, D, Ts
Output: Estimated Period (T')

cn ← 14
Ni ← ⌊(7T0/4)/Ts)⌋
Nf ← ⌈(7T_0/3)/Ts)⌉
If 'Nf - Ni' ≥ '2cn'
  zos ← round(linspace(Ni, Nf, cn))
  N' ← F2(D, zos, Ni, Nf, cn, n, nH)
else
  zos ← Ni : Nf
  j ← F1(D, zos, n, nH)
  N' ← zos (j)
end
T' ← (Ts (N' - 1))/n

Source: own.

If the zos has less than 28 points, the algorithm runs the code in Table 2. The output of this part of the code corresponds to the index of best candidate, $N' = zos(j)$.

Table 2: F1, Minimization.

Inputs: D, zos, n, nH
Output: j

Gr ← ⌊$(nH-1)/(n-1)$⌋
Re ← $nH - Gr(n-1)$
Gen ← $Gr*n + Re$
While 'm ← 1 : numel(zos)'
  F ← fft (D(1: zos(m))) / zos(m)
  FM (m,:) ← F(2 : Gen + 1)
end
FM ← ||FM||
FM ← Extract columns (nT : n : Gen) from FM
ΔH ← Add up FM by rows
j ← Index of the element in zos that minimizes ΔH

Source: own.

If the zos has more than 28 points, the algorithm runs the subsampling process in Table 3. The result of the subsampling is again $N' = zos(j)$.

Table 3: F2, Subsampling.

Inputs: D, zos, Ni, Nf, cn, n, nH
Output: N'

A ← minimal difference among data in zos
While 'A' > '1'
  j ← $F1(D, zos, n, nH)$
  N' ← zos(j)
  If 'j + 2' ≥ 'length of zos'
  Right limit ← length of zos
  else
  Right limit ← j + 2
  end
  If 'j - 2' ≤ '1'
  Left limit ← '1'
  else
  Left limit ← j - 2
  end
  Ni ← zos(left limit)
  Nf ← zos(rigth limit)
  zos ← round(linspace(Ni, Nf, cn))
  A ← Minimal difference among data in zos
End

Source: own.

2.6. *Example*

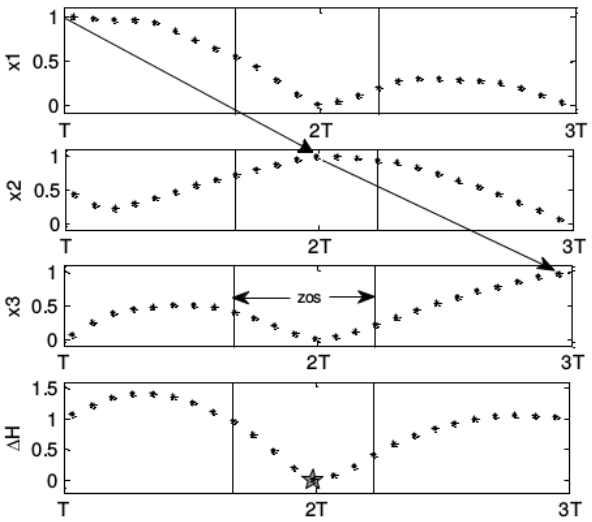
Consider a function $f(t) = sin(t) + 0{,}5sin(2t)$, whereas $Ts = 0{,}5s$, $T_0 = 6$ s, $n = 2$, and $nH = 2$. The graphical application of the ΔH algorithm is shown in Figure 3. The value of the harmonics at $t = T$ is $x1 = 1$, $x2 = 0{,}5$, $x3 = 0$, as expected given $f(t)$. In addition, the value of $x1(T)$ reapears at $x2(2T)$ and $x3(3T)$, as indicated by the arrows. On the other hand, the continues and vertical lines bound the zone of search, which result in 12,5 s as the best candidate, therefore $T' = 6{,}25$ s. Thus, the relative error equals $e_r = 0{,}06$.

3. Error analysis

This section analysis the error in the period estimation when using the proposed algorithm for noiseless and noisy signals. The main goal of this section consists in finding bounds for the error in comparison with the bounds for other three algorithms.

The first of those algorithms is the periodogram. This algorithm assumes as the estimated period the inverse of the frequency for the harmonic with the highest amplitude. Thus, uses DFT to decompose a signal into harmonics. Although popular to estimate periods, it requires data for several periods. The second algorithm, Δf, compares two consecutive sections of a signal with length t_N, and chooses as the estimated period the t_N

Figure 3: Application example of the ΔH algorithm.



Source: own.

that produces the smallest variation among sections. The third algorithm, ΔS, similar to the previous algorithm, also compared two sections, but proposes as period the t_N that maximizes the number of data points shuffling data points of the first section, according to the explanation in [8].

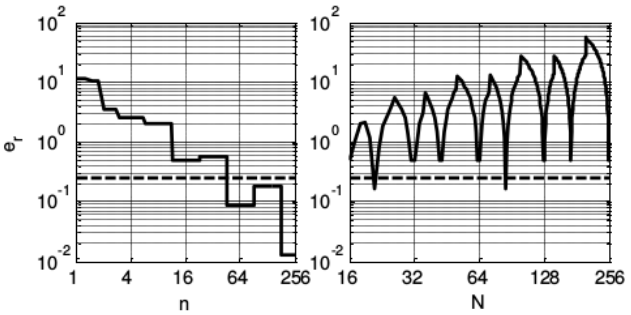
The Discrete Fourier Transform, base of the proposed algorithm, assumes that a data point matches the period, but in practice a distance (always smaller than Ts) separates the period from the closest data point. Thus, $T = t_i + \Delta NTs$, where t_i equals the data point at the left of the period, whereas ΔN equals the fraction of Ts necessary for t_i to become T. Therefore, $\Delta N = \frac{T-t_i}{Ts}$. This ΔN fraction causes errors in the estimations for algorithms based on the DFT, because this transformation assumes $\Delta N = 0$, when actually $0 \leq \Delta N < 1$. Given that rank of change for ΔN this fraction results ideal to analyze the error, which denotes the distance between the real period, T, and its approximation, T', relative to the sampling period, namely $e_r = \frac{|T-T'|}{Ts}$.

Experiments show that the behavior of the error found in this section applies for all the tried functions; however, in order to provide a guarantee in the bound of the error, the results will be limited to functions described by a number smaller or equal than 11 harmonics. This, given computation limits. In addition to the restriction in the number of harmonics of a signal, the amplitude of each harmonic will be also limited: from -1 to 1 for the first harmonic, whereas the second has smaller amplitudes, decreasing linearly, until the 12th (and successive harmonics,) with zero amplitude. The first harmonic has 11 values evenly distributed, 10 for the second, 9 for the third, and so on. Thus, the combination of all harmonics totals around 40 million functions.

3.1. Error for noiseless signals

The ΔN fraction influences the error, e_r, but the biggest cause of error for a periodogram resides in the data length, n, and in the number of data points per period, N, as shown in Figure 4. That figure shows the error e_r for one of the functions with harmonic content described above, where $\Delta N = 0{,}5$. The left part of the figure, where $N = 18$, shows an error e_r falling under 0,25 when $n \geq 46$. The bound 0.25, added for comparison, is the maximum error using the proposed algorithm as will be shown. The right part in the figure 4 shows the error for data length equal to two periods, $n = 2$, when the number of data points per period, N, changes from 16 and 64. Instead of decreasing, the error increases in relation to N.

Figure 4: Error characterization for a periodogram.



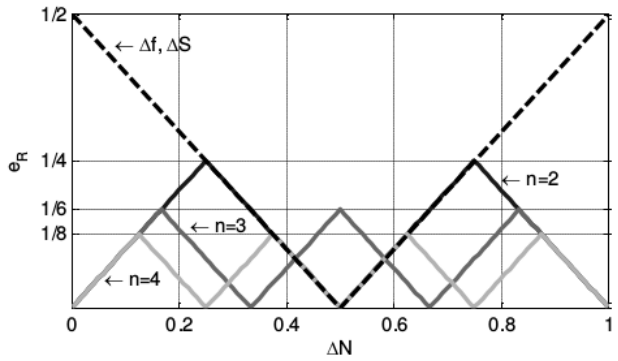
Source: own.

In contrast to the error caused by a periodogram, the error for Δf and Δs algorithms solely depends on the value of ΔN, and has a bound equal to 1/2, as shown in Figure 5. That bound means that the worst approximation provided by those algorithms falls below $\frac{Ts}{2}$, regardless of the function, period, or the number of data points per period.

The error for the proposed algorithm also depends exclusively on ΔN, but if $n = 2$ the maximum error for ΔH equals 14 (half the value for Δf and Δs). Another advantage of the proposed algorithm consists in the reduction of the maximum error 1 merely increasing the length of data, n, which results in a bound equal to $2n$. This expression comes from the observation of the experimental results, such as the behaviors in

Figure 5. On the other hand, increasing n requires higher precisions in the initial approximation, T_0, according to the expression in equation (3).

Figure 5: Relative errors for Δf, Δs, and ΔH algorithms.



Source: own.

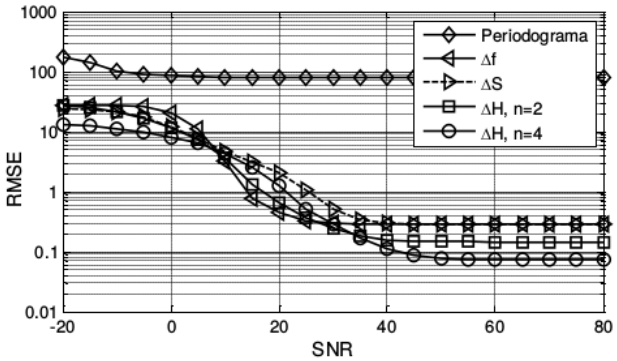
We anticipated that the number of valleys nH influences the error behavior, but nH does not have any effect for noiseless signals. This is because all the valleys have the minimum at the same location. Thus, using one or several valleys produces the same estimations; however, nH does affect the error value for noisy data, as shown in the application section.

3.2. *Error for noisy signals*

The evaluation of the error's behavior for noisy signal uses the Monte Carlo method. Its application starts by defining the Signal Noise Ratio, SNR which equals $20log_{10}\frac{Signal\quad Amplitud}{Noise\quad Amplitud}$ in decibels. Thus, cleaner signals have higher values of SNR. Each SNR requires the running of 1×10^5 experiments. During each experiment, and randomly, the method changes the value of N, ΔN, T,T_0 , $f(t)$, whereas $nH = 1$. Thus, $70 \leq N \leq 300$; $0 \leq \Delta N < 1$; T adopts one of ten possibilities: $\frac{3}{2},\frac{3}{4},\frac{1}{2},\frac{4}{5},1,\pi,\sqrt{2},\sqrt{3},e^1,\frac{1+\sqrt{5}}{2}$; T_0 takes a random value according to equation (3); and finally, $f(t)$ corresponds to one of the functions from the previous section. Then, the Root-Mean- Square-Error ($RMSE$) compacts the errors for the 1×10^5 experiments in a single value.

High values of SNR in Figure 6 match the results in the previous section for noiseless signals:$e_r = 12$ for Δf y Δs, $e_r = 14$ for $n = 2$ for ΔH, and $e_r = 18$ for $n = 4$. Contrary, the error for the periodogram rounds $e_r = 100$, because N adopts a random value, whereas the algorithm expects multiples of 2. The proposed algorithm, ΔH, keeps its performance even until $SNR = 40$, and then the error grows, but the same happens for all the algorithms. The best result for extremely noisy signals ($SNR = -20$), corresponds to ΔH, when $n = 4$, which shows robustness of the algorithm.

Figure 6: Effect of the noise content in the performance of four algorimths.



Source: own.

4. Applications

Another way to test the proposed algorithm corresponds to apply it to solve real problems. For instance, this section uses the algorithm to approximate the period of physiological signals and the frequency of electrical signals. The challenge with physiological signals resides in its quasi-periodic nature, whereas the challenge with electrical signals rests in the speed of these signals.

4.1. *Physiological signals*

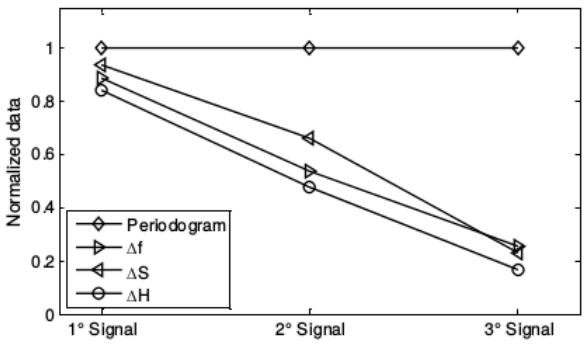
This section uses three signals from Physionet [9]. The first from the database "Apnea-ECG Database (apnea-ecg)", record a02. The second from the database "AF Termination Challenge Database (aftdb)", record test-set-a/a02, signal ECG (second wave). The third signal also from "AF Termination Challenge Database (aftdb)", but record test-set-a/a06, signal ECG (first wave). The first signal belongs to the family of apnea (temporary cessation of breathing during sleep). The other two signals belong to the family of auricular fibrillation (an abnormal heart rhythm). The main feature of these signals relates its quasi-periodic nature, with variations in amplitude, shape and period; in addition, they have high noise content that represents a challenge for any algorithm to estimate the period [10].

Evaluating the quality of an estimation for these databases emerges as a problem, because the high

variability of the signals left the data without a reference to contrast estimations. The definition of a method to contrast results starts by assuming the first point of the database as the first data point of the signal. Once the algorithms produce their approximations, the measure of error corresponds to summing up the magnitude of the first nine harmonics for the difference between two consecutive sections: first, from 0 and T', and the second, from T' to $2T'$. The experimentation shows that those nine harmonics contain most of the information; also, that higher similarity among sections produces less harmonic content, and thus smaller errors.

If the evaluation of the error finishes there, any algorithm would be the best, given random variations in the signals. T_0 eliminate that effect, the contrast procedure continues by changing the signal. The second time, the first data corresponds to the second point of the database, whereas the initial estimation, T_0, equals previous T' . This procedure continues until exhausting data. The contrast process finishes by adding up the errors for all trials, and normalizing them in such a way that the worst result turns into 1, whereas the other results changes proportionally, as shown in Figure 7.

Figure 7: Normalized error for three physiologinal signals.



Source: own.

The periodogram repeats as the worst approximation in Figure 7, because $n = 2$, which is few data for a periodogram. On the other hand, ΔH has the smallest error among the remaining algorithms, but not far from Δf and Δs. This trial, in contrast with the analysis in previous section, shows the effect of changing the number of valleys, nH. The result in Figure 7 corresponds to $nH = 4$, smaller values of nH results in Δf as the best approximation.

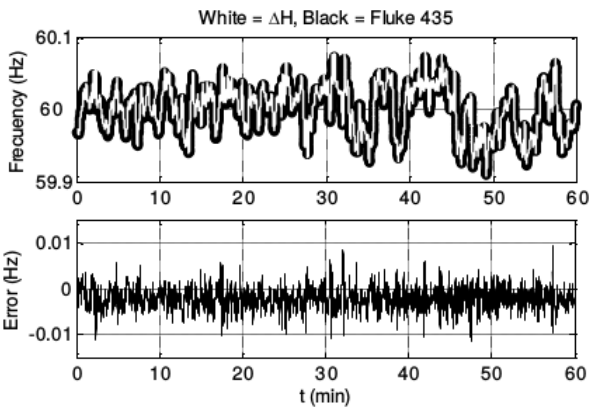
4.2. *Frequency of an electrical signal*

The frequency of the power grid changes as a function of the size of incoming or outgoing loads in comparison with the size of the generators. Maintaining the quality of the electric service implies restricting that frequency under certain limits. Thus, frequency estimation becomes essential to keep quality under the limits imposed by regulations.

The experiment with alternating voltage consists on approximate its frequency ($f' = \frac{1}{T'}$) using ΔH, while measuring the same signal, as comparison, with a power quality and energy analyzer Fluke 435 series II. This device provides estimations every 0,5 s. The approximation process starts by recording data every $Ts = 5\mu s$, which corresponds to the limit of the data acquisition card, the PCI 6024E of National Instruments. The recording of data last 85 ms (about 5 periods if $f = 60$ Hz), and then the algorithm approximates the frequency, which last around 100 ms in a pentium IV. Same process happens every 0.5 s and last for an hour. The acquisition also requires changing the voltage from 120 V to 5 V with a conventional transformer.

The initial approximation equals $T_0 = 60$ s . In addition, trials with the number of valleys comes up with $nH = 4$, because the approximations through time describe a smoother curve, as shown in the higher part of Figure 8. Smaller values generates peaks, whereas higher do not improve the estimation any more. The experiment in Figure 8 was carried out in October 13 , 2016 , from 16: 17: 18 and 17: 17: 18 in Bogota, Colombia.

Figure 8: Estimation of the frequency for a power grid.



Source: own.

The higher part of the Figure 8 shows the estimation of the specialize device, using a thick black line, and over

that line, in white, the result for the proposed algorithm. The estimation coming from ΔH matches the frequency of the power grid given by the analyzer. The lower part of the figure presents the difference between estimations, which falls under 10 mHz.

5. Conclusions

In this paper, we present an algorithm based on the Discrete Fourier Transform to estimate the period of a signal given samples at a constant sampling rate. The initialization of the algorithm requires data and the definition of three parameters: an initial estimation, T_0; a number of periods n where to look for the estimation; and the number of harmonics to analyze, nH. Results for noiseless signals, or low noise content, shows a bound of the relative errrelative error depends only on the parameter n, regardless of the function, period, or sampling rate. The proposed algorithm, in the applications section, serves to estimate the period of physiological human signals, which became a challenge because the high variability of the signals left the data without a reference to contrast estimations. The second application consists in estimating the frequency of a power grid; the main difficulty in this case corresponds to the real time requirements. This paper proposes a subsampling method aiming to make estimations faster, but future work may look for using the data smartly in order to reduce the computational time even more. For instance, given the valley shape in the zone of search, it would interesting to study the application of a predictive algorithm to come faster with the minimum in that zone.or in the estimation equal to $\frac{1}{2n}$. Thus, the relative error depends only on the parameter n, regardless of the function, period, or sampling rate. The proposed algorithm, in the applications section, serves to estimate the period of physiological human signals, which became a challenge because the high variability of the signals left the data without a reference to contrast estimations. The second application consists in estimating the frequency of a power grid; the main difficulty in this case corresponds to the real time requirements. This paper proposes a subsampling method aiming to make estimations faster, but future work may look for using the data smartly in order to reduce the computational time even more. For instance, given the valley shape in the zone of search, it would interesting to study the application of a predictive algorithm to come faster with the minimum in that zone.

References

[1] R. G. McKilliam, B. G. Quinn, I.V.L Clarkson, and B. Moran, "Frequency estimation by phase unwrapping," in *IEEE Transactions on Signal Processing*, Vol. 58, no. 6, 2010, pp. 2953–2963, https://doi.org/10.1109/TSP.2010.2045786

[2] S. Provencher, "Estimation of complex single-tone parameters in the DFT domain," in *IEEE Transactions on Signal Processing*, Vol. 58, no. 7, 2010, pp. 3879– 3883, https://doi.org/10.1109/TSP.2010.2046693

[3] H. C. So, F. K. W. Chan, and S. Weize, "Subspace approach for fast and accurate single-tone frequency estimation", in *IEEE Transactions on Signal Processing*, Vol. 59, no. 2, 2010, pp. 827–831, https://doi.org/10.1109/TSP.2010.2090875

[4] R. Chudamani, K. Vasudevan, C. S. Ramalingam, "Real-time estimation of power system frequency using nonlinear least squares," in *IEEE Transactions on Power Delivery*, Vol. 24, no. 3, 2009, pp. 1021–1028, https://doi.org/10.1109/TPWRD.2009.2021047

[5] Y. Pantazis, O. Rosec, Y. Stylianou, "Iterative estimation of sinusoidal signal parameters," in *IEEE Signal Processing Letters*, Vol. 17, no. 5, 2010, pp. 461–464, https://doi.org/10.1109/LSP.2010.2043153

[6] C. Candan, "A method for fine resolution frequency estimation from three DFT samples," in *IEEE Signal Processing Letters*, Vol. 18, no. 6, 2011, pp. 351–354, https://doi.org/10.1109/LSP.2011.2136378

[7] C. Yang, and G. Wei, "A noniterative frequency estimator with rational combination of three spectrum lines," in *IEEE Transactions on Signal Processing*, Vol. 59, no. 10, 2011, pp. 5065–5070, https://doi.org/10.1109/TSP.2011.2160257

[8] J. D. Rairan, "Two Algorithms for Estimating the Period of a Discrete Signal," in *Ingenieria e investigación Journal*, Vol. 34, no. 3, 2014, pp. 57–63, doi: dx.doi.org/10.15446/ing.investig.v34n3.41943

[9] Physionet, "PhysioBank ATM". 2016 [Online] Available: http: physionet.org/cgi- bin/atm/ATM

[10] R. Llinares, and J. Igual, "Exploiting periodicity to extract the atrial activity in atrial arrhythmias", in *EURASIP Journal on Advances in Signal Processing*, 2011, pp. 1–16, https://doi.org/10.1186/1687-6180-2011-134