Adaptive Incremental Estimation Filter for AC Noise in the Electrocardiogram

Dobromir Petkov Dobrev and Tatyana Dimitrova Neycheva

Abstract – **Power-line interference is a common disturbing factor in almost all biosignal acquisition applications. Many filtering procedures for mains interference elimination are available, but all of them are still not enough effective to fully overcome the problem. An interesting adaptive filtering technique for the power-line interference, called 'incremental estimation', was published in the literature. It uses a small step to increment or decrement the amplitude of the estimated interference, synthesized as a pure sine wave. This paper gives the frequency response of the filter and investigates its effectiveness with real ECG signals and Matlab simulations.**

Keywords – **Power-Line Interference (PLI), Adaptive filter, Incremental Estimation Filter (IEF), Electrocardiogram**

I. Introduction

Power-line (PL) interference is a common disturbing factor in almost all biosignal acquisition applications. As a consequence of electrode impedance imbalance and the finite value of the amplifier CMRR, some AC noise remains even when special signal recording techniques are applied (shielding, driven right leg, body potential driving, etc.). A further reduction of PLI usually is achieved by digital post-filtering. Many algorithms for PLI suppression are available, starting from simple comb filters [1], to advanced subtraction procedures and lock-in techniques [2, 3], but all of them tend to lose their efficiency when PL frequency differs from its nominal value.

Nowadays, signal processing capability of popular low-cost microcontrollers is continuously improved. Thus, modern adaptive filtering techniques become more and more popular. Adaptive techniques are advantageous in periodic noise filtering, echo cancellation, signal extraction, etc. because they change their characteristics as the noise or signal of interest change. Such filters operate like a servo system with negative control loop, which minimize a given loss or error function while optimizing the filter coefficients and extracting the noise. Usually, the filter minimizes the output signal power by minimizing the Mean Squared Error (MSE), and such approach of iteratively modifying the filter coefficients using the MSE is called the Least Mean Squared (LMS) algorithm [4, 5]. Once the output power is minimized the noise is canceled. The common disadvantage of such algorithms is their complexity, thus they are unsuitable for popular microcontrollers and real time processing.

A very clever and tricky adaptive filtering approach was published in [6]. The technique is called 'Incremental Estimation' (IE), and is invented by Davide Mortara [6]. The approach is well described also in [5] and [7].

Now, a linearized model of IE approach for frequency response evaluation is developed. The effectiveness of the approach is investigated with real ECG signals and Matlab simulations.

D. Dobrev is with the Institute of Biophysics and Biomedical Engineering – Bulgarian Academy of Sciences, Bl. 105 Acad G. Bontchev Str., 1113 Sofia, Bulgaria,
e-mail: dobri@biomed.bas.bg

T. Neycheva is with the Institute of Biophysics and Biomedical Engineering – Bulgarian Academy of Sciences, Bl. 105 Acad G. Bontchev Str., 1113 Sofia, Bulgaria,
e-mail: tatiana@biomed.bas.bg

II. Overview of Adaptive Incremental Estimation Approach for PLI

The operating principle of Incremental Estimation Filter (IEF) is as follows [5, 6, 7]. Let's assume that the PLI is a pure sine wave with amplitude A and frequency ω:

$$e(t) = A\sin(\omega t) \tag{1}$$

For the current n^{th} sample in discrete time processing, the Eq. (1) can be expressed as Eq. (2):

$$e_n = e(nT) = A\sin(\omega nT) \tag{2}$$

Replacing (nT) with ($nT - T$) in Eq. (2), an expression for the past sample can be found:

$$e_{n-1} = e(nT - T) = A\sin(\omega nT - \omega T) \tag{3}$$

Similarly, the same can be done for the future sample ($nT + T$), and Eq. (2) becomes:

$$e_{n+1} = e(nT + T) = A\sin(\omega nT + \omega T) \tag{4}$$

Recalling to the trigonometric identity:

$$\sin(\alpha + \beta) = 2\sin\alpha\cos\beta - \sin(\alpha - \beta) \tag{5}$$

where α and β are:

$$\alpha = \omega nT \; , \; \beta = \omega T \tag{6}$$

and replacing Eq. (5) in Eq. (4) gives:

$$e_{n+1} = 2A\sin(\omega nT)\cos(\omega T) - A\sin(\omega nT - \omega T) \tag{7}$$

The first term in Eq. (7) contains Eq. (2), and the second term is Eq. (3), so the Eq. (7) can be rewritten as Eq. (8):

$$e_{n+1} = 2\cos(\omega T)e_n - e_{n-1} \tag{8}$$

The term $cos(\omega T)$ is a constant determined only by the PL frequency f_{pl} and the sampling frequency f_s=1/T:

$$N = \cos(\omega T) = \cos(2\pi f_{pl} / f_s) \tag{9}$$

Replacing Eq. (9) in Eq. (8) gives a relation for the future sample of the sinusoidal noise, based on the values of the current and the past samples.

$$e_{n+1} = 2Ne_n - e_{n-1} \tag{10}$$

The output of the filter is the difference between the input and the estimated noise:

$$y_{n+1} = x_{n+1} - e_{n+1} \tag{11}$$

Thus, if the input is only noise and the estimate is exactly tracking, the filter output will be zero.

Producing the estimated signal requires multiplication by a fraction N given in Eq. (9). Such a multiplier requires floating point arithmetic, which could considerably slow the algorithm. In order to approximate such multiplier, it might be built on a summation of power-of-two fractions, which can be implemented with simple bit-shift operations and can be faster and only possible on popular microcontrollers. For example, if f_s=2kHz, N=0.9877008 could be realized as: $N \approx 1-2^{-6}+2^{-8}-2^{-11}-2^{-12} = 0.98767$.

The synthesized by Eq. (10) sine wave needs of a negative feedback control loop to adjust the sinusoidal amplitude for each sample, and thus to track the changes in the PLI.

The input signal for two neighbor samples consists of slowly varying signal or DC component, and PLI noise:

$$x_{n+1} = x_{DC_{n+1}} + e_{n+1} \tag{12}$$

$$x_n = x_{DC_n} + e_n \tag{13}$$

Assuming that the input DC level does not change significantly between samples, then an error function can be defined as:

$$f_{err} = x_{DC_{n+1}} - x_{DC_n} \approx 0 \tag{14}$$

Replacing Eq. (12) and Eq. (13) in Eq. (14) gives:

$$f_{err} = (x_{n+1} - e_{n+1}) - (x_n - e_n) \tag{15}$$

Eq. (15) can be rearranged as:

$$f_{err} = (x_{n+1} - x_n) - (e_{n+1} - e_n) \tag{16}$$

Thus, Eq. (16) presents a subtraction of the first difference of the estimated noise from the first difference of the input signal. It cancels the DC levels while simultaneously comparing and adjusting to equalize the increment in the estimated waveform to the increment in the input. That is why, the filter is called ‘incremental estimation’ by Mortara in [6]. In other words, the input and the estimated noise are initially high-pass filtered and after that compared. The result of comparison is the value of the error function f_{err}. If the error function f_{err}>0, the amplitude of the PLI estimate is adjusted upward by a small step size d.

$$e_{n+1} := e_{n+1} + d \tag{17}$$

If the error function f_{err}<0, the PLI estimate is adjusted downward by the same small step size d.

$$e_{n+1} := e_{n+1} - d \tag{18}$$

If the function f_{err}=0, the estimate is not changed:

$$e_{n+1} := e_{n+1} \tag{19}$$

The choice of d is empirically determined and depends on how quickly the filter needs to track the changes in the interfering noise. If d is large, then the filter quickly adapts to the noise change. With a smaller d, the filter requires a longer learning time but provides more exact tracking of the noise. If the value of d is too large or too small, the filter will never converge to a proper noise estimate. A starting value of d could be less than 1LSB, e. g. 0.25LSBs [5].

III. FREQUENCY RESPONSE

It is more convenient for the realization and simulation of the Eq. (10) and Eq. (11) to rewrite them for one sample interval in the past:

$$e_n = 2Ne_{n-1} - e_{n-2} \tag{20}$$

Thus, the output of the filter becomes:

$$y_n = x_n - e_n \tag{21}$$

The filter frequency response can be evaluated on a linearized model shown in *Fig. 1*.

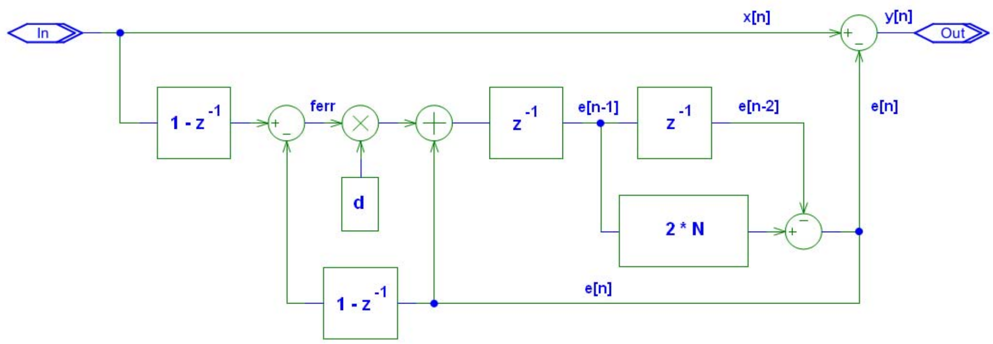

Fig. 1. Linearized model for frequency response simulation of IEF for PLI

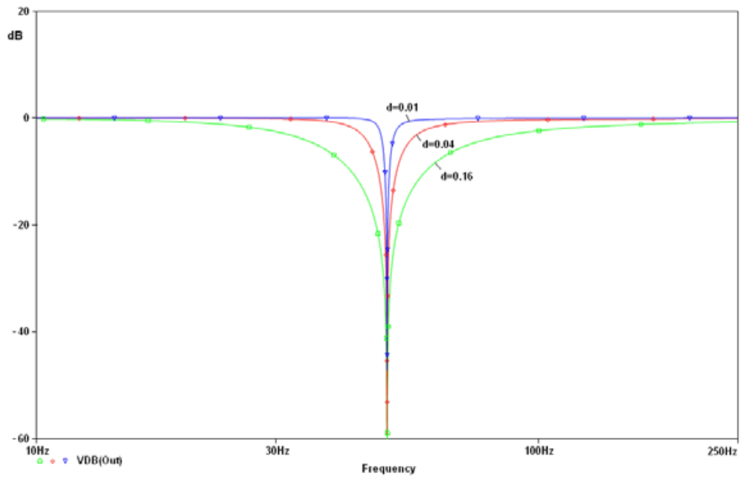

Fig. 2. Frequency response of IEF for PLI

The schematic from *Fig. 1* is simulated and the result is shown in *Fig. 2*.

Looking at the schematic in *Fig. 1* it is clear that it consists of two high-pass first difference filters. One is in series to the input signal *x[n]*. The second one is in series to the estimated signal *e[n]*. The second one also is appeared inside the control loop. So, its characteristic, towards *e[n]*, will be inverted from high-pass to low-pass, and ignoring the other blocks for sine wave generation, we can say that *e[n]* will have a combination of high-pass and low-pass response, i. e. it will exhibit a band-pass characteristic. Subtracting band-pass characteristic from the input, (i. e. from unity), the characteristic is again inverted and for the output *y[n]* the final characteristic becomes band-rejection or notch.

What about the coefficient *d*? The coefficient *d* controls the amount of the feedback and in this manner, the bandwidth, i. e. the quality factor *Q* of the whole filter.

From *Fig. 3*, when d_1=0.01 the bandwidth at 3dB is Δf_1=3.2Hz. When d_2=0.04, the bandwidth is Δf_2=12.7Hz, and when d_3=0.16, the bandwidth is Δf_3=48.6Hz.

The quality factor *Q* can be calculated according the well known formula:

$$Q = \frac{f_{pl}}{\Delta f_{3dB}} \tag{22}$$

Thus, for f_s=2kHz and f_{pl}=50Hz, when d_1= 0.01 Q_1=15.6, for d_2=0.04 Q_2=3.9, and for d_3=0.16 Q_3=1.03. When *d* is referred to 1V, the bandwidth can be expressed as [7]:

$$\Delta f_{3dB} = \frac{d \cdot f_s}{2\pi} \tag{23}$$

Replacing Eq. (23) in Eq. (22), the formula for *Q* becomes:

$$Q = \frac{2\pi f_{pl}}{d \cdot f_s} \tag{24}$$

Thus, using the Eq. (24), the calculated values of *Q* are: Q_1=15.7, Q_2=3.925, Q_3=0.98. As can be seen, the calculated values correspond to the simulated ones, and this proves the truthfulness of Eq. (23) and Eq. (24).

The consequence is that the quality factor *Q* is inversely proportional to the sampling rate f_s, and to the size of the correction step *d*.

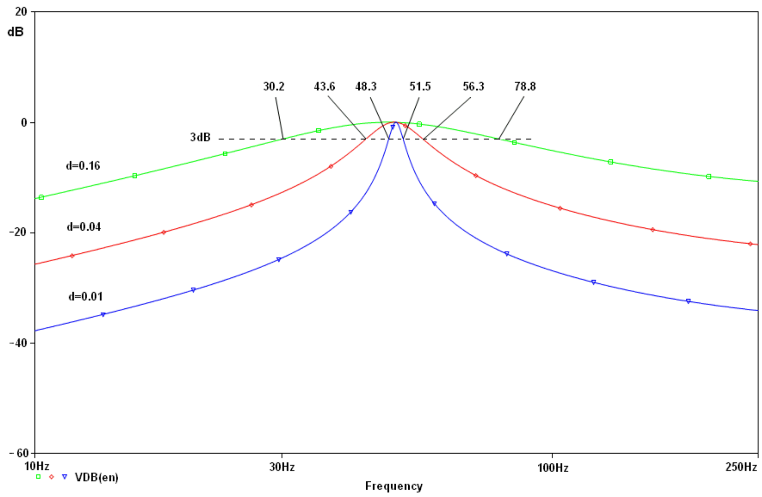

Fig. 3. Frequency response of the estimated PLI at the output *e[n]*

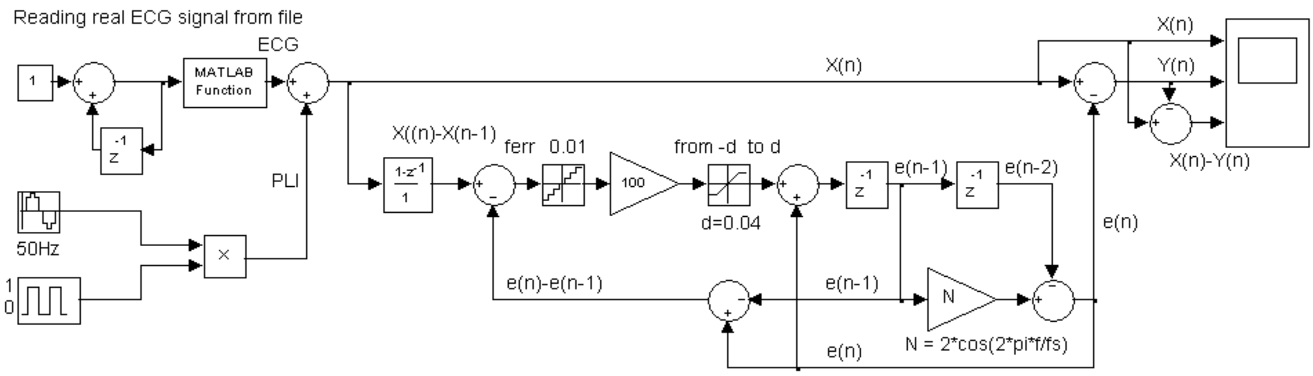

Fig. 4. Simulink schematic for IEF simulation

IV. MATLAB SIMULATIONS

Simulink schematic for IEF simulation is shown in *Fig. 4.* The simulation result is shown in *Fig. 5*, where the first trace is the ECG signal with noise, the second trace is the filtered ECG signal and the third trace is the PLI estimate, i. e. the difference between trace 1 and trace 2. 1LSB corresponds to 1.25uV.

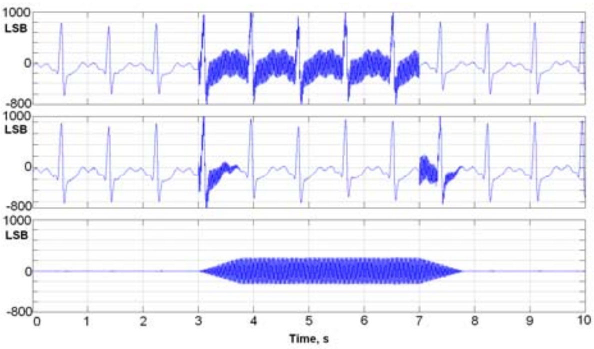

Fig. 5. IEF simulation result with real ECG signal

It can be seen that the IEF quickly tracks the PLI.

V. CONCLUSION

The IEF is presented and its frequency response is analyzed. The filter generates a sine wave and adjusts its amplitude to track the PLI. The quality factor Q is inversely proportional to the sampling rate f_s, and to the size of the correction step d. Generally speaking, to not distort the useful ECG signal, the PLI filter must have bandwidth of ±5Hz or Δf<10Hz, i. e. its Q factor must be higher than 5 or according the Eq. (24), the step size d must be lower than 30m. The filter adaptation time depends on the selected Q factor, and is inversely proportional to the step size d. The filter employs comparison of increments in the input signal and in the generated estimate. This is possible only if the input signal is slowly varying in comparison to PLI noise. Dynamic increase of the step size d can lead to faster adaptation time, but after adaptation, the step d must be returned to its nominal value to avoid generation of spurious residual noise in the ECG.

The filter behavior corresponds to the selected Q factor, and is similar to other high-Q filters [8]. The main advantages of the approach is its relative simplicity, and that the sampling rate is not needed to be multiple to the PL frequency. If the PL frequency is changed, the filter accurately can track the new value only by changing the coefficient N. This is not possible to other filters like [1, 2, 3, 8], where for maximal rejection the sampling rate must be multiple to PLI frequency.

REFERENCES

[1] D. Dobrev, T. Neycheva, N. Mudrov. *Simple High-Q Comb Filter for Mains Interference and Baseline Drift Suppression*, Annual Journal of Electronics, Vol. 3, No 1, pp. 50-52, 2009.
[2] Ch. Levkov, G. Mihov, R. Ivanov, I. Daskalov, I. Christov, I. Dotsinsky. *Removal of Power-Line Interference from the ECG: a Review of the Subtraction Procedure*, Biomed Eng Online 4:50, 2005.
[3] D. Dobrev, T. Neycheva, N. Mudrov. *Digital Lock-in Techniques for Adaptive Power-Line Interference Extraction*, Physiol Meas 29, pp. 803–816, 2008.
[4] B. Widrow, J. Glover, J. McCool, J. Kaunitz, Ch. Williams, R. Hearn, J. Zeidler, E. Dong, R. Goodlin. *Adaptive Noise Canceling: Principles and Applications*. Proc. IEEE, 63(12), pp. 1692–1716, 1975.
[5] W. Tompkins. *Biomedical Digital Signal Processing*, Prentice Hall, pp. 174-183, 1995.
[6] D. Mortara. *Digital Filters for ECG Signals*, Computers in Cardiology, pp. 511-514, 1977.
[7] R. Limacher. *Removal of Power Line Interference from ECG Signal by an Adaptive Digital Filter*, Proc. of 16th European Telemetry Conference ETC96, Garmisch-Partenkirchen, pp. 300-309, 1996.
[8] D. Dobrev, T. Neycheva, N. Mudrov. *High-Q Comb Filter for Mains Interference Suppression*, Annual Journal of Electronics, Vol. 3, No 1, pp. 47-49, 2009.