A review of transient suppression methods of IIR notch filters used for power-line interference rejection in ECG measurement

A Review of Transient Suppression Methods of IIR Notch Filters Used for Power-line Interference Rejection in ECG Measurement

Shadi Mahdiani[1], Vala Jeyhani[2] and Antti Vehkaoja[2]

[1] Tampere University of Technology/Department of Electronics and Communications Engineering, Tampere, Finland
[2] Tampere University of Technology/Department of Automation Science and Engineering, Tampere, Finland

Abstract— **Bioelectric signals are often corrupted by noise. The most common form of noise is power-line interference and its harmonics. A convenient way for eliminating these unwanted components is to use a single or multiple notch filters. One of the problems about this approach is the effect of transient response of the filter at the beginning of its output in short time measurements. In this work, three initialization methods, which can be used to reduce/overcome this problem are reviewed and their performance and computational complexity are evaluated using ECG as an example signal. These methods are projection initialization, pole radius-varying filtering and vector projection. Additionally, some implementation variations and memory usage considerations are discussed. Our study shows that, pole radius-varying method is computationally cheap but introduces longer transient than the others. On the other hand, vector projection provides a more accurate reconstruction of the signal in the transient part of the output but with a more expensive computation. There are also two drawbacks about vector projection. One is its computational complexity dependency to the sampling frequency of the signal and the other is the fact that it cannot provide the results in real-time.**

Keywords— **ECG, initialization, filter, IIR, transient**

I. Introduction

Electrocardiogram (ECG) signal is the electrical manifestation of the contractile activity of the heart and can be recorded easily with surface electrodes. ECG is a well-known diagnostic tool for the detection of cardiovascular diseases and abnormalities. Therefore the quality of the ECG signal is an important factor which should be taken into consideration in measurement and processing phases of this signal. [1]

Generally, bioelectrical signals can be easily corrupted by different noise sources due to their low amplitude and frequency range. The most common form of noise is power-line interference and its harmonics [2]. Using single or multiple notch filters are the simplest ways for removing this form of noise from measured signals [3]. Although, when using conventional filtering, beginning of the signals, which might include some significant information, is entirely lost due to transient state of the filter.

In designing the notch filters, there is always a trade-off between transient response duration and the rejection bandwidth of the filter. In other words, a great selectivity at notch frequency, which requires a very narrow rejection bandwidth (high quality factor Q), results in a long transient response. One example where this problem should be addressed is where intermittent measurements of ECG are used for arrhythmia detection [4]. Another application is ambulatory monitoring of event related potentials (ERP) of electroencephalogram (EEG) such as mismatch negativity [5]. In ERP monitoring, the relevant measurement time is known and the required recording duration is short, usually less than 1 second. These examples can be extended to a large application area in which the measurements are done intermittently, in order to reduce the power consumption. In these applications, due to the short length of the recorded signals, the transient state of a filter with high quality factor corrupts a large part of the information.

Earlier, various approaches for reduction the transient state of the infinite impulse response (IIR) notch filters have been introduced. However, the error and computational cost of these approaches have not been discussed and compared. In this work, three methods for filter coefficient initialization are compared and the results about their performance and computational complexity are presented. These methods are projection initialization, pole radius-varying filtering and vector projection. There are some other techniques in literature which are not considered in this study because they do not suit notch filters due to following reasons. These methods are step initialization [6] [7], single frequency initialization [8] and exponential initialization [9]. In step initialization, the memories of the IIR filter are initially loaded with their steady-state values for a step input and the transient effects due to the DC component in the clutter is eliminated. This is achieved only if the filter has a steady state zero at DC [8]. The single frequency initialization method needs two channels [10], i.e. in-phase and quadrature in radar applications, which makes it inappropriate for conventional noise removal from biomedical signals. Initialization using two or more fre-

quencies discussed in [10] and [11] have the same problem. In exponential initialization, the signal is modeled as a single complex exponential with a specific amplitude, frequency and phase. Using the formulas explained in [9], the frequency and phase of the model are zero for a real signal and the estimated waveform would be a DC signal.

The rest of the paper is organized as follows. In Section II, initialization techniques and the methods used for their comparison are explained. The results of the comparisons are given in Section III. Finally, some concluding remarks are given in Section IV.

II. Methods

A. Traditional IIR Notch Filter

The input of the filter is considered as the following form

$$x(n) = s(n) + A\sin(n\Omega_0 + \phi), \tag{1}$$

where $s(n)$ is a desired signal corrupted by a sinusoidal interference signal with angular frequency Ω_0, phase shift ϕ and amplitude A. The transfer function of a second-order IIR notch filter with pole radius r and notch frequency Ω_0 can be written as

$$H(z) = \frac{1 - 2\cos(\Omega_0)z^{-1} + z^{-2}}{1 - 2r\cos(\Omega_0)z^{-1} + r^2 z^{-2}} \cdot \tag{2}$$

The filter is stable for $r < 1$. By increasing the radii of the poles inside the unit circle, the selectivity of the filter improves but the transient duration also increases. The zeros, whose angles are equal to notch frequency, are constrained to lie on the unit circle. Poles are placed at the same radial line as zeros. Equation (2) can be expressed in time domain as

$$\begin{aligned} y[n] =& x[n] - 2\cos\Omega_0 x[n-1] + \\ & x[n-2] + 2r\cos\Omega_0 y[n-1] - r^2 y[n-2] \cdot \end{aligned} \tag{3}$$

When no initialization technique is used, the unknown values (e.g. $x(n-1)$ in time instance 0) are replaced with zeros (zero initialization) which introduces an error in the output due to the transient response of the filter.

B. Projection Initialization

This technique is based on the state-space variable [12]. The transfer function of a general second-order notch filter can be written as

$$H(z) \triangleq \frac{Y(z)}{X(z)} = \frac{\alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2}}{1 + \beta_1 z^{-1} + \beta_2 z^{-2}} \cdot \tag{4}$$

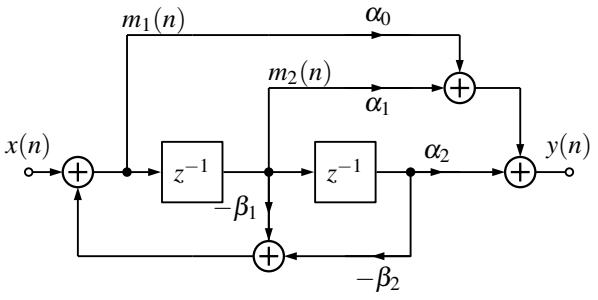

Fig. 1: State-space variable representation of a second-order IIR filter

By comparing Equations (4) and (2), it can be seen that $\alpha_0 = \alpha_2 = 1$, $\alpha_1 = -2\cos\Omega_0$, $\beta_1 = -2r\cos\Omega_0$ and $\beta_2 = r^2$. The internal state of the filter is specified by

$$M[n] = \begin{bmatrix} m_1[n] \\ m_2[n] \end{bmatrix}, \tag{5}$$

and the state-space description of the filtering equations is expressed as

$$M[n] = BM[n-1] + Cx[n] \tag{6}$$

and

$$y[n] = A^T M[n-1] + \alpha_0 x[n], \tag{7}$$

(see Fig. 1) where

$$A \triangleq \begin{bmatrix} \alpha_1 - \alpha_0\beta_1 \\ \alpha_2 - \alpha_0\beta_2 \end{bmatrix}, \quad B \triangleq \begin{bmatrix} -\beta_1 & -\beta_2 \\ 1 & 0 \end{bmatrix}, \quad C \triangleq \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{8}$$

With only a finite windowed version of data available, the state-transient equation can be written as

$$M[n] = B^n M_{-1} + \sum_{k=1}^{n} B^{n-k} C x[k], \tag{9}$$

where $M_{-1} \triangleq \begin{bmatrix} m_1[-1] & m_2[-1] \end{bmatrix}^T$ is a parameter describing the initial state of the filter. The output samples Y can also be expressed in the form of

$$Y = FM_{-1} + GX, \tag{10}$$

where F, which is an $N \times 2$ matrix, is defined as

$$F = \begin{bmatrix} A^T \\ A^T B \\ \vdots \\ A^T B^{N-1} \end{bmatrix} \tag{11}$$

and G, which is an $N \times N$ lower triangular matrix, is defined as

$$G = \begin{bmatrix} \alpha_0 & 0 & \cdots & 0 & 0 & 0 \\ A^T C & \alpha_0 & \cdots & 0 & 0 & 0 \\ A^T BC & A^T C & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ A^T B^{N-4} C & A^T B^{N-5} C & \cdots & \alpha_0 & 0 & 0 \\ A^T B^{N-3} C & A^T B^{N-4} C & \cdots & A^T C & \alpha_0 & 0 \\ A^T B^{N-2} C & A^T B^{N-3} C & \cdots & A^T BC & A^T C & \alpha_0 \end{bmatrix}. \tag{12}$$

Then, the initialization vector M_{-1} is calculated by

$$\hat{M}_{-1} = QX^T, \tag{13}$$

where

$$Q = -(F^T F)^{-1} F^T G. \tag{14}$$

These formulas are then substituted in Equations (6) and (7) for calculation of the output in the transient period of the filter.

For deriving the initialization vector indicated in Equation (14), the matrix Q can be calculated offline and saved in the memory. The size of the matrix Q depends on the number of input samples that undergo the initialization process. The size of Q and X is $2 \times N$ and $1 \times N$, respectively, where N is the number of input samples used for initialization. Therefore, $2N$ multiplications and $2(N-1)$ additions are needed to generate $\hat{M}_{-1}$. This can be done only when all the first N input samples have been received. After receiving the $x[N-1]$, the program starts producing $y[0]$ to $y[N-1]$ output samples while the next coming input sample is $x[N]$. In other words, for real-time applications, all the operations should be done after N-th and before $(N+1)$-th input samples.

To overcome this problem the computations should be distributed between input samples, which can be done by converting the equations to an iterative form (similar to what is proposed for vector projections). Note that this optimization can be performed only for $\hat{M}_{-1}$, because the rest of the processing depends on $\hat{M}_{-1}$ and can be started only when $\hat{M}_{-1}$ is ready. Fig. 2 shows the structure of this implementation. The illustrated multiplication includes two multiplier and ACC shows an accumulator which adds its previous value, which is a 1×2 matrix, to the new value coming from the multipliers. Therefore, two multiplications and two additions are needed for the input samples $x[0]$ to $x[N-1]$, which can be executed right after receiving each of them. The final value is updated after the N-th input sample is received. In this step, $\hat{M}_{-1}$ is ready and $M[0]$, $y[0]$ and $y[1]$ can be calculated, which requires 6 multiplications and 6 additions. Output samples $y[2]$ to $y[N-1]$ can be calculated using Equation (3).

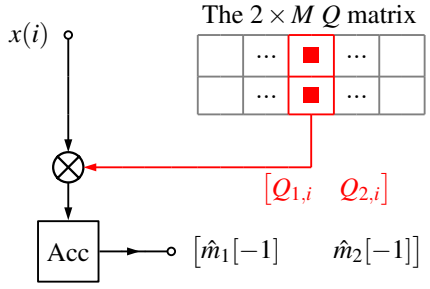

Fig. 2: Computationally distributed implementation for $\hat{M}_{-1}$ derivation in projection initialization

C. Vector Projection

In 1995, Pei and Tseng introduced vector projection to find non-zero initial conditions for IIR notch filter [13]. In this method, first, vector projection is applied to break down the first M samples of noisy signal into two parts; clean (desired) and sinusoidal interference parts. Then, the clean part is used as initial values for the traditional notch filter. The algorithm is explained in following. First, the input data is arranged as vector $X = \begin{bmatrix} x(0) & x(1) & \cdots & x(M-1) \end{bmatrix}^T$, then matrix A is constructed as

$$A = \begin{bmatrix} 1 & \cos(\Omega_0) & \cdots & \cos((M-1)\Omega_0) \\ 1 & \sin(\Omega_0) & \cdots & \sin((M-1)\Omega_0) \end{bmatrix}, \tag{15}$$

and projection matrix P is computed as

$$P = A(A^T A)^{-1} A^T. \tag{16}$$

The first M samples of the output signal are then obtained by

$$Y = \begin{bmatrix} y[0] & y[1] & \cdots & y[M-1] \end{bmatrix}^T = QX, \tag{17}$$

where $Q = (I - P)$ and I is identity matrix. The rest of the signal samples, which are samples M to $N-1$, can be calculated by Equation (3).

According to [14], the length of the input vector should cover the period of power-line fundamental frequency to achieve the optimal non-zero initial conditions for the notch filter. This indicates that with higher sampling rate a longer window of input samples is needed to reconstruct the noise accurately.

Matrix Q can be directly saved into the memory of the processing unit and only the multiplications in Equation (17) are needed to be done online.

Equation (17) shows an $(M \times M) \times (M \times 1)$ matrix multiplication that produces an $(M \times 1)$ output Y. With the normal matrix multiplication, all the input samples must be available to calculate each individual element in Y. In this case, all the

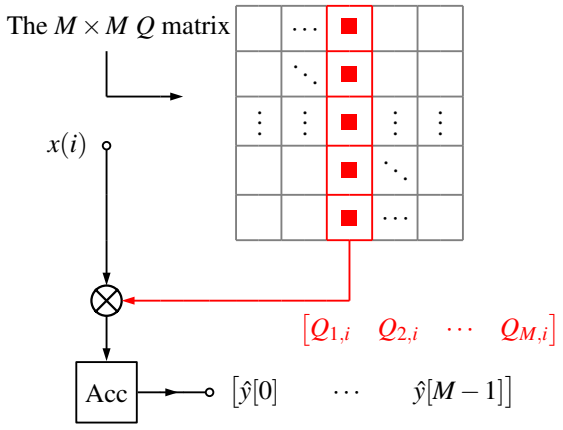

Fig. 3: Computationally distributed implementation of Vector Projection initialization

computations can be done only after receiving the first M input samples. In order to distribute the computations between input samples and be able to execute as many computations as possible by having each input sample, the method illustrated in Fig. 3 can be used.

Fig. 3 shows the computations for i-th input sample. After receiving the i-th sample, it is multiplied by the i-th column of Q matrix and produces an M-element vector. The result is then fed into an accumulator which adds its previous value (which is an $M \times 1$ matrix) to the result of multiplication. After the M-th input sample, the result of accumulator represents Y in Equation (17). Note that this final result is ready only after the M-th input sample. With this method, M multiplications and M additions are required for each input sample.

D. Notch Filter with Time-Varying Pole Radius

In Piskorowski's method [15], it is shown that if the pole radius increases in time, starting from a small value, the transient behavior of the filter at the beginning of its output diminishes. Thus, a function was presented to vary the pole radius of the filter, from a small to the desired value. The second-order IIR notch is presented as:

$$\begin{aligned} y[n] =& x[n] - 2\cos\Omega_0 x[n-1] + x[n-2] + \\ & 2r[n]\cos\Omega_0 y[n-1] - r^2[n] y[n-2] \end{aligned} \tag{18}$$

in which, $r[n]$ is the changing pole radius over time. As the formulation of $r[n]$ is described in [15], only the final equation is presented here.

$$r[n] = \bar{r}.(1 + (d_r - 1)e^{-\frac{n}{\nu f_s}}), \quad n \geq 0 \tag{19}$$

where $\bar{r}$ is the final value of the pole radius. The parameters d_r and ν describe the variation range and exponential variation rate of the function, respectively. This function is similar to the one presented in [16] for formulating *Quality Factor Varying* notch filter.

The implementation structure of this method can be based on the normal IIR notch filter with the same number of multiplications and additions. But, the number of memory units required for saving coefficients would be much larger than in a normal IIR filter. Instead of one coefficient $2r\cos\Omega_0$, M coefficients $2r[n]\cos\Omega_0$ are required, where M is the number of samples which are processed by the pole radius-varying filter. Similarly, M coefficients are needed for $r^2[n]$. More specifically, for the first M input samples, $2M+1$ memory locations are needed for the coefficients.

One alternative for the implementation of pole radius-varying filter is to save only $r[n]$ values. In this case, instead of $2M+1$ memory locations, $M+1$ locations are needed for coefficients. But two more multiplications are required for each input sample.

E. Database and Comparison Method

The ECG signal for testing the methods is from subject 123 in MIT-BIH arrhythmia data base [17]. The signal has been recorded at 360 samples per second. Therefore, according to [14] (as explained earlier), 6 samples are required by the projection initialization and vector projection methods to provide one full period of 60 Hz power-line interference.

The performance of the transient suppression methods was determined by calculating mean square error defined as

$$MSE = \frac{1}{N}\sum_{n-1}^{N}(\hat{y}[n] - y[n])^2, \tag{20}$$

where y and $\hat{y}$ are the filtered signal and the optimal output, respectively.

III. Results and Discussions

For this study, the employed signal was a clean signal which was summed with an artificially generated 60Hz sinusoidal noise with an SNR of 2.2 dB. The signal itself was then used as the ground truth for calculating the error introduced by the initialization techniques. With traditional second order IIR notch filter, the transient response lasts about 400ms.

Table 1 shows the errors for the initialization methods. For projection initialization and vector projection, M was set to 6. In pole radius-varying method, final pole radius $\bar{r}$, variation range d_r and variation rate ν were set to 0.98, 0.8163, and

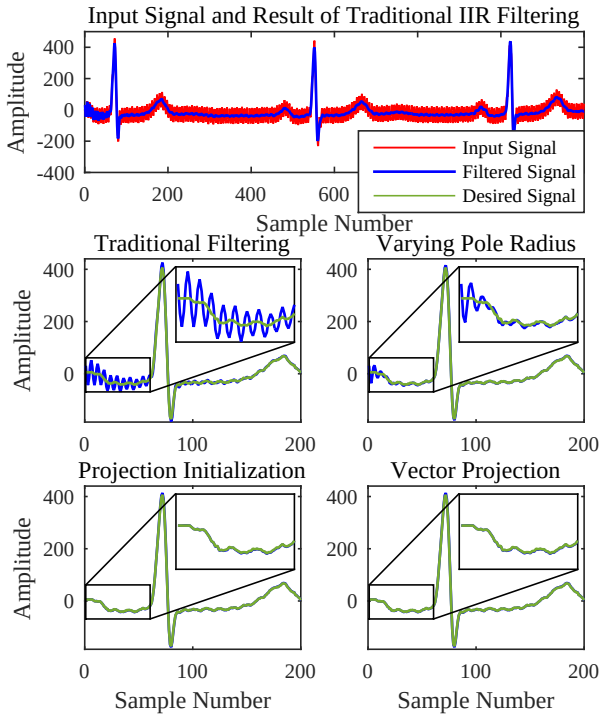

Fig. 4: Input signal and the results of different initialization methods

0.05, respectively. These values were found to produce the best result, in which the transient response and the difference between the filtered and desired signals were as low as possible. The error was calculated in a window of the first 100 samples in order to emphasize on the samples which were affected by the transient response. It can be seen that vector projections and projection initialization methods provide the most accurate results. Fig. 4 shows the output of different initialization techniques. Top panel shows the noisy signal and output of traditional IIR filtering.

Fig. 5 and Fig.6 show the number of multiplications and additions for each method, respectively. The computations are distributed between input samples as described in previous section. In both figures, normal notch filter refers to

Table 1: Amount of error introduced by studied initialization methods

Method	MSE
Zero initialization	17.99
Pole radius-varying	8.07
Vector projection	2.57
Projection initialization	2.57

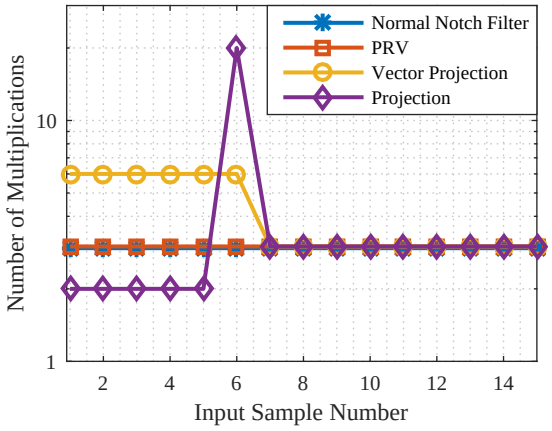

Fig. 5: The number of multiplications for each input sample. PRV refers to the pole radius-varying method.

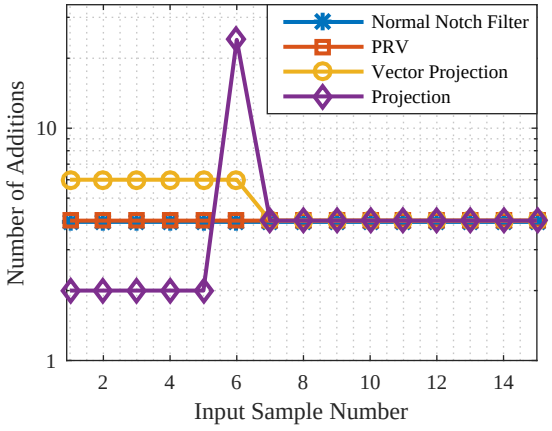

Fig. 6: Number of additions for each input sample. PRV indicates the pole radius-varying method.

the IIR notch filtering with zero-initialization and is considered to be implemented using the Direct Form I structure. Pole radius-varying method is indicated as RPV. It can be seen that the projection initialization method has a high peak in the last input sample. After this step, the input signal is filtered using Equation 3. This peak value contains the computations for $\hat{M}_{-1}$, $M[0]$, $y[0]$ and $y[1]$ using state-space variable implementation, and $y[2]$ to $y[5]$ using normal second order notch filter with Direct Form I implementation. PRV needs the same amount computations as the traditional filtering but it requires $2M+1$ memory locations only for coefficients (thus more memory fetches) while normal second order notch filter needs only 3 memory units. Vector projection needs a few more computations than Direct Form I, but this is also affected by the sampling frequency.

Fig. 7 shows the effect of sampling frequency on number of multiplications for the M-th input sample processed

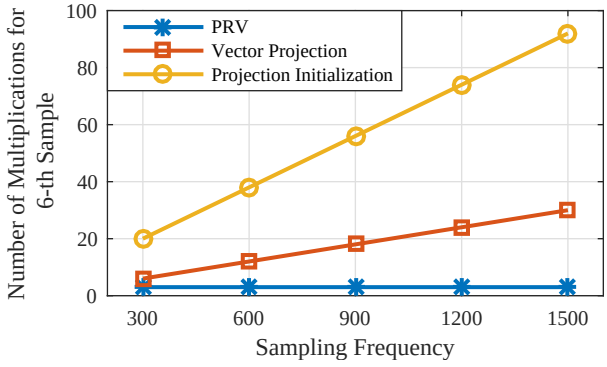

Fig. 7: Effect of sampling rate on the number of multiplications for the last input sample processed by the three initialization methods.

by pole radius-varying initialization, projection initialization and vector projection. It is worth mentioning that in projection initialization, the amount of calculations for samples before the M-th sample is independent of the sampling frequency.

IV. Conclusion

In this work, projection initialization, pole radius-varying filtering, and vector projection techniques for notch filter initialization were compared and the results about their performance and computational complexity were presented. Choosing the best method is a trade-off between the error and the computational complexity and depends on the specifications of implementation and hardware platform.

Pole radius-varying method is computationally cheap but introduces larger error than the other two methods. This is because of two facts. One is the effect of the zero initialization of the first IIR filter which is constructed by $r[0]$ and the second one is removing 60Hz neighboring frequency components due to small Q-factor in early stages of filtering.

Projection initialization and vector projection methods propose the smallest error. The latter has a stable but larger computational complexity. While, the former has a an expensive computational peak which is largely affected by the sampling frequency. There are two main drawbacks in these two methods. One is that the first M output samples can be calculated only after receiving the M-th input sample. Another drawback is that the computational cost for each input sample varies with M which is dependent to sampling frequency.

References

1. Rangayyan R M, Reddy N P. *Biomedical signal analysis: a case-study approach*. New York, NY: Wiley-Interscience 2002.
2. Huhta James C, Webster J G. 60-Hz Interference in Electrocardiography *Biomedical Engineering, IEEE Transactions.* 1973;BME-20:91–101.
3. Pei S C, Tseng C C. IIR multiple notch filter design based on allpass filter *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions.* 1997;44:133–136.
4. Hendrikx T, Rosenqvist M, Wester P, Sandström H, Hörnsten R. Intermittent short ECG recording is more effective than 24-hour Holter ECG in detection of arrhythmias *BMC cardiovascular disorders.* 2014;14:41.
5. Vehkaoja A, Lekkala J. Wireless measurement band for EEG mismatch negativity registration in mobile activities in *Proceedings of XVIII IMEKO world congress: metrology for a sustainable development*(Rio de Janeiro, Brazil) 2006.
6. Burlage D W. An initialization technique for improved MTI performance in phased array radars *Proceedings of the IEEE.* 1972;60:1551–1552.
7. Dewald K, Bersier A, Gardella P J, Jacoby D. IIR filter transient suppression by signal shifting in *Biennial Congress of Argentina (ARGENCON), 2014 IEEE*:153–158 2014.
8. Al-Ahmad H, Ahmed K. A novel technique for initializing digital IIR filters with a finite number of samples at a single frequency *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions.* 1997;44:417–420.
9. Peterson R B, Atlas L E, Beach K W. A comparison of IIR initialization techniques for improved color Doppler wall filter performance in *Ultrasonics Symposium, 1994. Proceedings., 1994 IEEE*;3:1705–1708 1994.
10. Al-Ahmad H, El-Khazali R. A new technique for initializing digital IIR filters at two frequencies in *Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference*;1:64–67 2003.
11. Jeedella J, Ahamad H Al, Al-Mualla M E. Optimum bandwidth IIR notch filters with multiple frequency initializations in *Electronics, Circuits and Systems, 2005. ICECS 2005. 12th IEEE International Conference*:1–4 2005.
12. Chornoboy E S. Initialization for improved IIR filter performance *Signal Processing, IEEE Transactions.* 1992;40:543–550.
13. Pei S C, Tseng C C. Future of health insurance *Elimination of AC interference in electrocardiogram using IIR notch filter with transient suppression.* 1995;42:1128–1132.
14. Piskorowski J. Powerline interference removal from ECG signal using notch filter with non-zero initial conditions in *Medical Measurements and Applications Proceedings (MeMeA), 2012 IEEE International Symposium*:1-3 2012.
15. Piskorowski J. Powerline interference rejection from sEMG signal using notch filter with transient suppression in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*:1447–1451 2012.
16. Piskorowski J. Digital Q -Varying Notch IIR Filter With Transient Suppression *Instrumentation and Measurement, IEEE Transactions.* 2010;59:866–872.
17. Moody G B, Mark R G. The impact of the MIT-BIH Arrhythmia Database *Engineering in Medicine and Biology Magazine, IEEE.* 2001;20:45–50.

Author: Shadi Mahdiani
Institute: Tampere University of Technology
Street: Korkeakoulunkatu
City: Tampere
Country: Finland
Email: shadi.mahdiani@tut.fi