Subtraction Procedure for Removing the Baseline Drift from ECG Signals: Adaptation For Real Time Operation With Programmable Devices

Tsvetan Shoshkov[1] and Georgy Mihov[2]

Abstract – **The present work is focused on developing a generalized subtraction procedure for suppressing baseline drift from electrocardiogram (ECG) signals, operating in real time and adapting it for programmable logic (FPGA) operation. The development of the algorithm is done in Matlab environment and is tested for many practical scenarios. The prototype version of the subtraction procedure is implemented for programmable logic using VHDL. Here is presented the generated programmable logic structure in FPGA. Converting the theoretical algorithm to real time execution determines the use in calculations of single signal values in their natural sequence. The execution results show that the developed real time algorithm successfully eliminates the baseline drift in one ingoing ECG sample, concluded before the appearance of the next one. This research represents a suitable platform for future investigations and development of the subtraction procedure and also for particular practical applications.**

Keywords – **ECG, drift, drift removing, subtraction procedure.**

I. Introduction

The electrocardiogram (ECG) is often affected by drift, due to complex mechanical and electro-chemical electrode-to-skin processes. The subtraction procedure shows good results in removing baseline drift from ECG signals. The process flow is shown on fig. 1. Its structure contains three main stages:

– *Detecting linear segments*. Each ECG sample is checked if it belongs to a linear segment by using appropriate linearity criterion. The linearity is defined by comparing the criterion with a predefined threshold M.

$$Cr \le M \,. \tag{1}$$

– *Baseline drift calculation*. If it is detected a linear segment the baseline drift is calculated using a digital filter and it is stored in a temporal FIFO buffer. In the same time the baseline drift is subtracted from the linear segment;

– *Baseline drift extrapolation*. If the current sample belongs to a nonlinear segment the value of the drift is calculated using the data stored in the temporal buffer and it is subtracted from the signal in a nonlinear segment.

The present work describes the organization of the subtraction procedure for ECG drift removing in real time and its adaptation with programmable logic FPGA.

[1]Tsvetan Shoshkov is with the Faculty of Electronic Engineering and Technologies at Technical University of Sofia, 8, St. Kl. Ohridsky Blvd, 1000 Sofia, Bulgaria, E-mail: tsh@tu-sofia.bg.

[2]Georgy Mihov is with the Faculty of Electronic Engineering and Technologies at Technical University of Sofia, 8, St. Kl. Ohridsky Blvd, 1000 Sofia, Bulgaria, E-mail: gsm@tu-sofia.bg.

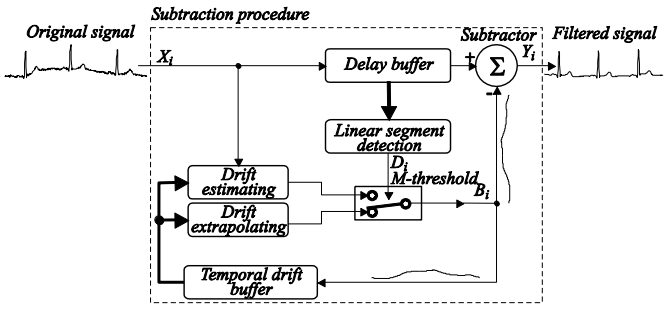

Fig. 1. Basic Structure of the Subtraction Procedure for Baseline Drift Removing.

II. Adapting the Subtraction Procedure for Real Time Drift Removing

The prototype of the drift removing algorithm using the subtraction procedure is developed on the basis of the latest investigations of the subtraction procedure [1, 2]. The baseline drift filtering algorithm in [1] is a procedure that is not working in real time and is processing a buffer with samples of the input signal.

An important point that distinguishes the real time subtraction procedure algorithm is the processing of the signal samples in their natural sequence. Thus the digital filter needs access to several of the previous and next samples. The input signal is being stored in a delay buffer where a sequence of samples that surround the current one can be accessed.

The delay buffer length is defined by the length of the longest epoch that is processed in one stage of the procedure. In this case this is the linearity detection procedure. The linearity criterion is applied over k periods of the power line frequency, that is $N = k.n + 1$. The $n = \Phi/F$ ratio represents the multiplicity of the sampling frequency Φ and the power line frequency F. It is rarely an integer and in the calculations it is used the rounded value $n = round(\Phi/F)$. Due to some considerations it is selected $k = 6$. The length of the delay buffer is $N = 6.n+1$ samples, its organization is shown on fig. 3. The input sample X_i is stored in the temporal buffer $X_i \rightarrow X_{B0}$ and all other values are shifted in one step forward.

The used linearity criterion is

$$Cr = |D| \,. \tag{1}$$

The value of D represents the signal acceleration that is calculated using its second derivative. This corresponds to the second difference in digital signal processing. It is calculated

```
for i=1+N: 1: length(X);                % Start of Subtraction procedure
    for j=N: -1: 2; XB(j)=XB(j-1); end % Delay buffer shifting
    XB(1) = X(i);            % Storing ongoing sample into Delay buffer
    if Max<XB(1+(N-1)/2);Max=XB(1+(N-1)/2);
    else Max=Max-(Max-Min)/N/20; end;
    if Min>XB(1+(N-1)/2); Min=XB(1+(N-1)/2);
    else Min=Min+(Max-Min)/N/20; end;
    Mpp= Max-Min;
    if Mmu>Mpp; Mmu=Mpp;
    else Mmu=Mmu+Mpp/N/20; end
    Md = Mpp*mu;             % Dynamic threshold
    D = ((XB(1)-2*XB(1+(N-1)/2)+XB(N)))/4; % Normalised second difference
    FD = ((XB(1)-XB(N)))/2;                % Normalised first difference
    Cr = abs(D);             % Linearity criterion
    FCr = abs(FD)/5;         % Second criterion
    if Cr > Md;              % Non-linear segment
        B=DB(1)+(DB(1)-DB(N))/(N-1)/2; % Drift calculating
    else                     % Linear segment
        if FCr > Md;         % Second criterion threshold
            B=DB(1)+(XB(1)-DB(1))/(N-1)/1; % Drift calculating
        else
            B=DB(1)+(XB(1)-DB(1))/(N-1)/2; % Drift calculating
        end
    end
    for j=N: -1: 2; DB(j)=DB(j-1); end % Drift buffer shifting
    DB(1)=B;                           % Storing Drift into Drift buffer
    Y(i-(N-1)/2)=X(i-(N-1)/2)-DB(1);  % Ongoing output sample
end                           % End of Subtraction procedure
```

Fig.2. Fragment of the Program in the Matlab Environment for Baseline Drift Removing Using the Subtraction Procedure.

by subtracting the first two differences *FD* of the signal. Each of the first differences is obtained using the amplitude of two samples that are separated by *k.n*/2 periods of the power line frequency. The equation of the linearity detecting filter is

$$D_i = (X_{Bi+(N-1)/2} - 2.X_{Bi} + X_{Bi-(N-1)/2})/4. \quad (3)$$

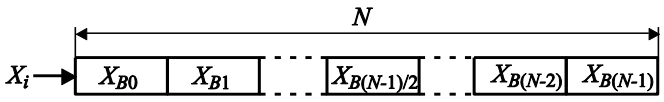

Fig. 3. Organization of the Delay Buffer.

The variety of ECG signals and the difference of their amplitudes demands dynamical defining of the linearity threshold *M*. In order to do so there are used two maximum value detectors M_{max} and M_{min}, each of them with an appropriate decreasing coefficient. If M_{max} is lower than the current value of X_{Bi}, then $M_{max} = X_{Bi}$. In the other case M_{max} is subtracted with $(M_{max} - X_{Bi})/(16.N)$. If M_{min} is larger than the current value of X_{Bi}, then $M_{min} = X_{Bi}$. In the other case M_{min} is decreased with $(X_{Bi} - M_{min})/(16.N)$. The difference $M_{pp} = M_{max} - M_{min}$ represents the coefficient that is used to determine the dynamical threshold

$$M = M_{pp}.\mu. \quad (4)$$

In this equation μ is selected experimentally for ECG signals in the range of 1 *mV*.

The *baseline drift extrapolation* stage is used in nonlinear segment. The easiest approach for extrapolation is the linear one. It requires two samples of the temporal buffer. The value of the drift in the nonlinear segments is

$$B_{Bi} = B_{Bi-1} + (B_{Bi-1} - B_{Bi-N})/(N-1). \quad (5)$$

A decreased second term of the equation in (5) is used to prevent drift overcompensating.

The temporal buffer contains *N* drift samples. Its organization is shown on fig. 4. The calculated drift is stored in the drift buffer $B_i \rightarrow B_{B1}$ and all other values are shifted forward.

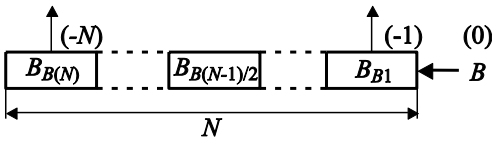

Fig. 4. Temporal Buffer for Drift Storing.

The *baseline drift calculation* stage is used when the linearity criterion detects a linear segment. Here are used two variants of calculation, one described in (6) and another one with decreased second term in (7).

$$B_{Bi} = B_{Bi-1} + \frac{B_{Bi-1} - B_{Bi-N}}{N-1}. \quad (6)$$

$$B_{Bi} = B_{Bi-1} + \frac{B_{Bi-1} - B_{Bi-N}}{2(N-1)}. \quad (7)$$

The periods in which the two methods are used are defined by a second criterion. It is calculated by using the first difference of the signal and defines its slope.

$$FD_i = (X_{Bi+(N-1)/2} - X_{Bi-(N-1)/2})/10. \quad (8)$$

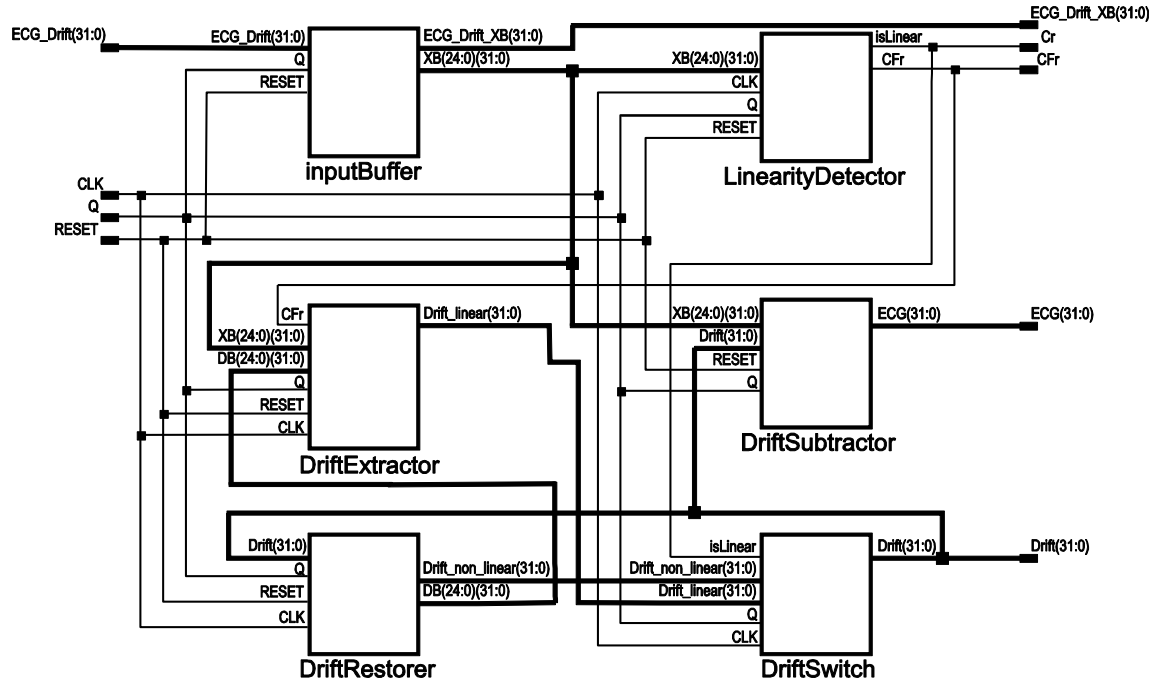

Fig.5. Structure of the Subtraction Procedure Realized For Programmable Logic.

The absolute value of the calculated criterion is compared with the dynamic threshold *M*.

$$FCr = \left| FD_i \right| . \tag{9}$$

$$FCr < M . \tag{10}$$

Depending on the result of the comparison one of the described equations for drift calculation is used. If (10) is true the equation that is used is (6), otherwise (7) is used.

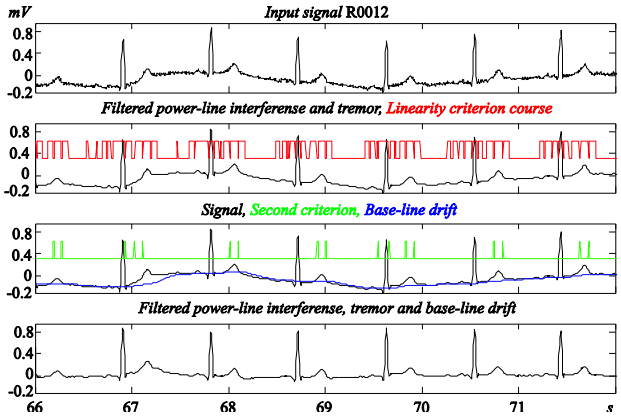

Fig. 6. Experiment in the Environment of Matlab.

Fig. 6 shows an experiment that is processed in the environment of Matlab with a six second long ECG signal that is sampled by 200 *Hz* signal and with 50 *Hz* frequency of the power line signal. The experiment shows that the baseline drift is successfully removed using the modeled subtraction procedure in Matlab. On the figure are also shown the graphics of the calculated linearity criterion and the extracted baseline drift.

III. Adaptation for Real Time Operation With Programmable Logic FPGA

After verifying the efficiency of the subtraction method for drift removing in Matlab, it is developed a hardware realization for real time operation. Programmable logic FPGA and VHDL programming language are used.

The description of the architecture of the device is shown on fig. 5 and it has the same structure of the procedure that is shown on fig. 1. The highest level of the description is structural where are declared and connected the main functional blocks. The descriptions of the blocks are functional and they are located in lower layers of the project. The basic functions of the algorithm are realized by:

– *shift register* is used to store a sequence of input signal samples (InputBuffer);

– *arithmetic logic block* to process input signal drift calculation (DriftExtractor);

– *shift register* stores a sequence of extracted drift samples and an *arithmetic logic block* restores the drift in the nonlinear segments (DriftRestorer);

– *arithmetic logic block* to calculate the linearity criterion (LinearityDetector);

– *signal switching block* that is controlled by the linearity criterion (DriftSwitch);

– *subtraction block* to calculate the output filtered signal using the difference of the input signal and the drift (DriftSubrtactor).

There are several approaches used in calculations to simplify the hardware realization. All the calculations are processed using floating point values. The values of the signals used to process the data are 32-bits wide, the higher 16

bits contain the integer part and the lower 16 bits contains the fractional part. In order to avoid calculation mistakes from losing accuracy some of the VHDL signals are 64-bits wide. Fig. 7 shows VHDL code of the DriftSwitch block.

```
entity Drift_Switch is
Port ( isLinear : in STD_LOGIC;
    Drift_linear : in STD_LOGIC_VECTOR (31 downto
0);
    Drift_non_linear : in STD_LOGIC_VECTOR (31
downto 0);
    Drift : out STD_LOGIC_VECTOR (31 downto 0);
    Q : in STD_LOGIC;
    CLK : in STD_LOGIC );
end Drift_Switch;
architecture Behavioral of Drift_Switch is
begin
    process (CLK) begin
        if (CLK'event and CLK = '1') then
            if Q = '0' then
                if isLinear = '0' then
                    Drift <= Drift_linear;
                else
                    Drift <= Drift_non_linear;
                end if;
            end if;
        end if;
    end process;
end Behevioral;
```

Fig.7. Fragment of the VHDL Code in Xilinx ISE

Two clock signals CLK and Q are used. The results of the calculations are processed for several periods of the CLK signal. The signal Q has frequency that is equal to the sampling frequency (in this case Φ = 200 Hz), it controls the input of the ECG signal (ECG_Drift), outputting the values of the filtered signal (ECG), subtracting the (Drift), linearity criterions (*Cr* and *FCr*). The second clock (CLK) must have frequency that is 32 times higher than the frequency of Q, a value that is defined by the digits of the numbers, so the required calculations can be processed. The speed of the present FPGA chips allows using clock signals with frequencies up to ten and hundreds MHz, so the frequencies of the CLK and Q signals can be realized. In real time processing of the data the output signal is delayed in comparison to the input ECG signal. This is the time delay needed to fill the input buffer. For more convenient observation of the results, in the output graphics it is used a delayed version of the input signal that is taken from the middle of the input buffer.

IV. Testing The Hardware Realization

The hardware description of the subtraction method for drift removing from ECG signals in real time is realized using the integrated development environment Project Navigator that is included in Xilinx ISE Design Suite 13.1.

To make simulations and test the described digital device, it is created a file with input signals (stimulus) that defines the behevior of the input signals of the device [5]. The input signals data is read from external files and output signals data is stored in external text files. The simulations are processed using the Xilinx ISim simulator and the results are displayed using Matlab. Observation and setting the work of the hardware realization is done with real ECG data.

The input and output signals of the processed simulation are shown on fig. 8. It is shown a processed signal from recording MO1_012.DCD (the same one is used in the experiment in Matlab on fig. 4).

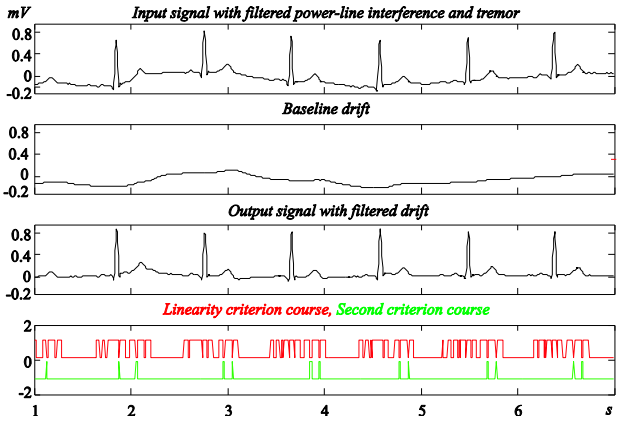

Fig. 8. Test with MO1_012.DCD.

The achieved accuracy of the hardware realization is remarkable and fully matches the theoretical model that is explored using Matlab. Using 32 bits signals leads to high accuracy and keeps the error of the filtration in the lower 1-2 bits.

V. Conclusion

The present work describes a practical realization of a very promising method for real time baseline drift removing from ECG signals. The theoretical basics of the method are realized and validated in Matlab. The developed theoretical model is realized in hardware using FPGA. In the process of creating the hardware description of the procedure there are made certain improvements in the calculation procedures. The simulations that are processed and the obtained results show high accuracy and reliability of the device that match the theoretical method.

References

[1] G. Mihov, I. Dotsinsky. “Subtraction Procedure For Removing The Baseline Drift From ECG Signals”, Annual Journal of Electronics, pp. 118-122, 2010.

[2] G. Mihov, I. Dotsinsky, C. Levkov, R. Ivanov, “Generalised Equations and Algorithm of the Subtraction Procedure for Removing Power-line Interference from ECG”, Proceedings of the Technical University of Sofia, Vol. 58, b. 2, Sofia, pp. 31-38, 2008.

[3] G. Mihov, I. Dotsinsky, T. Georgieva, “Subtraction Procedure for Powerline Interference Removing from ECG: Improvement for Non-Multiple Sampling”, Journal of Med. Engineering & Technology, 29, No 5, pp. 238-243, 2005.

[4] Ch. Levkov, G. Mihov, R. Ivanov, Ivan K. Daskalov, I. Christov, I. Dotsinsky, “Removal of power-line interference from the ECG: a review of the subtraction procedure”, BioMedical Engineering OnLine, 4:50, 2005.

[5] S. Mihov, R. Ivanov, A. Popov, “Real-Time Subtraction Procedure for Eliminating Power-Line Interference from ECG”, ET2008, b. 1, Sozopol, pp. 55-60, 2008.