35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

# Low Cost MATLAB-Based Pulse Oximeter for Deployment in Research and Development Applications

M. Shokouhian, R.C.S. Morling, and I. Kale

*Abstract*— **Problems such as motion artifact and effects of ambient lights have forced developers to design different signal processing techniques and algorithms to increase the reliability and accuracy of the conventional pulse oximeter device. To evaluate the robustness of these techniques, they are applied either to recorded data or are implemented on chip to be applied to real-time data. Recorded data is the most common method of evaluating however it is not as reliable as real-time measurements. On the other hand, hardware implementation can be both expensive and time consuming.**

**This paper presents a low cost MATLAB-based pulse oximeter that can be used for rapid evaluation of newly developed signal processing techniques and algorithms. Flexibility to apply different signal processing techniques, providing both processed and unprocessed data along with low implementation cost are the important features of this design which makes it ideal for research and development purposes, as well as commercial, hospital and healthcare application.**

*Keyword*—**pulse oximeter, pc-based, algorithm development, MATLAB.**

## I. Introduction

Overcoming to the problems which the pulse oximeter developers face during real-time measurement has lead engineers and researcher to develop different signal processing techniques and algorithms [1]-[4].

One of the main challenges in algorithm and signal processing development is the process of evaluation. The fastest way of testing the robustness of a technique is by applying it to a set of recorded data. Although this method of evaluation is fast and cheap, it is not as accurate and repetitive as a real-time evaluation because of the fact that it is restricted to a limited number of samples and the set of measurement conditions the samples have acquired.

For a comprehensive evaluation of newly developed algorithms, they need to be applied to real-time data. However, this method of evaluation has its own difficulties. The main issue for real-time evaluation is that the commercial pulse oximeters usually do not provide any access to the unprocessed data, therefore the designers need to build their own pulse oximeters from scratch which is both time consuming and costly. Other than that the algorithm developer may not have enough knowledge or may not want to be engaged with hardware implementation.

This paper presents a low cost MATLAB-based pulse oximeter which can be deployed by the algorithm developers for fast evaluation of pulse oximeter centric signal processing algorithms. One of the main advantages of the pulse oximeter system presented here is that different signal processing techniques can be implemented in a high level language and environment such as MATLAB and can be applied simultaneously or separately for evaluation purposes. The other advantage of this pulse oximeter system design is that both unprocessed and processed data are provided to the user and this empowers them to evaluate their developed signal processing techniques by comparing unprocessed and processed data.

Finally the presented system design procedure can significantly save both the time and the cost of evaluation. The first prototype of the system has been implemented on a 100mm by 60mm PCB board and it can be easily carried away with its simple PC/laptop interface to different places and deployed for the personal, commercial or research purposes.

## II. Design Methodology

The structure of the current system is shown in Figure 1. The overall structure can be divided into 5 different sections:

1. Commercial pulse oximeter probe
2. LEDs driving circuit
3. Photodiode conditioning and amplifying circuit
4. ATmega32 microcontroller
5. PC with MATLAB

Each of these sections is described briefly in the rest of the paper. The current system design uses ADC/DAC in loop to avoid the photodiode going into saturation. It also keeps the DC level of the detected red and IR signals equal and removes it from both optical channels in order to increase the dynamic range of the circuit [5].

### A. Probe

The pulse oximeter probe consists of one Red and Infra-Red (IR) LED and a photodiode. In order to reduce the number of pins in the probe the LEDs are usually connected back-to-back which makes it impossible to turn them on at the same time.

### B. LED Driving Circuit

The LED driving circuit gets four signals from the microcontroller (two for each LED). Two signals indicate the switching times of the LEDs and determine the time when

*Research supported by University of Westminster Scholarship.

M. Shokouhian is with the Applied DSP and VLSI Research Group, Electronics, Network and Computer Engineering Department, University of Westminster, London, United Kingdom (e-mail: m.shokouhian@ ieee.org).

R. C. S. Morling, is with the same Group and Department at Westminster (e-mail: morlinr@westminster.ac.uk).

I. Kale is also with the same Group and Department at Westminster (e-mail: kalei@westminster.ac.uk).

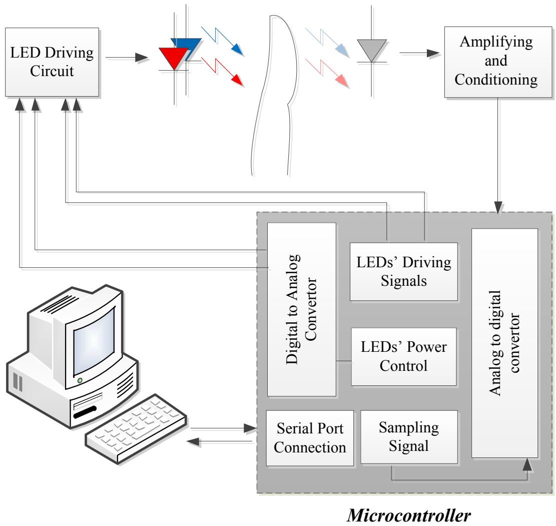


Figure 1. PC-Based Pulse Oximeter Structure

each LED needs to be turned on. The other two signals are the Pulse Width Modulation (PWM) signals which their average determines the intensity of light from each LED. From a top level point of view, the driving circuit consists of two variable current sources which are in series with a switch and a LED. When the switch is closed the current generated by the current source is passed through the LED and generates a proportional light intensity. Since the LEDs are connected back to back and cannot be turned on at the same time an H-bridge is used as the driving circuit [6].

### C. Analogue Signal Conditioning and Amplification

The photodiode generates a small electrical current in response to the detected light. For further processing, this small current needs to be converted to a voltage. This task is done by means of a Trans-Impedance Amplifier (TIA) shown in figure 2. A small bias voltage $V_{bias}$ is necessary for the correct operation of the circuit [6-7].

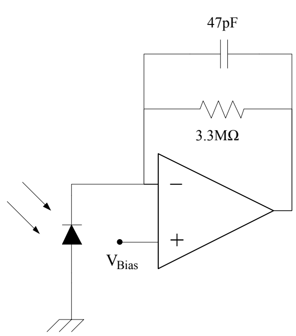


Figure 2. Trans-impedance amplifier circuit schematic

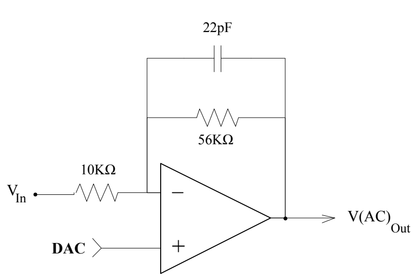


Figure 3. DC remover circuit schematic

Since the AC part of the detected signal is a small portion of the overall signal and because of its importance in oxygen level calculation, the AC part needs to be detected accurately. In order to achieve this, further analogue signal processing has been done on the signal by means of a second amplifier. In the second amplifier, the DC level of received signal is subtracted from the overall signal and the residual AC part is amplified. Figure 3 shows the circuit diagram of the second stage analogue signal conditioning. The DAC in this section generates a DC signal that is equal to the DC level of the converted voltage [6].

### D. ATMega32 Microcontroller

At the heart of the system there is an ATMega32 microcontroller which acts as the overall controller for the entire system. This relatively powerful microcontroller has eight 10-bit ADCs and two 10-bit PWM generators used as a DAC after some filtering. It also has a built-in USART which makes communication easy with computer. All of these features come with a relatively cheap price. The function of microcontroller in the overall system is as follows:

- It generates two non-overlapping square waves used for switching the red and IR LEDs.
- It samples the data synchronously with the driving LED signals at both the TIA and AC amplifier outputs. The former data is used to equalize the DC level of the two optical channels and to generate a DC level equal to that and the latter one is used to calculate the level of oxygen in arterial blood.
- It controls the intensity of light emitting from each LED by looking at the DC level of the incoming red and IR signals and equalizing them by changing the average level of the PWM signals.
- It has a DC tracker mechanism following the DC level of the detected signals and generating a PWM signal proportional to it. The filtered version of this PWM signal is used by the second amplifier (DC Remover shown in figure 3) to remove the DC voltage level from the overall received signal [6].
- It sends the AC component of the red and IR signals to the host computer for processing via MATLAB. The communication protocol used in this prototype was RS232.

### E. MATLAB Program

The transmitted data from the microcontroller is fed into the MATLAB environment and it is written into a buffer for further processing. The designers can then apply their own signal processing techniques to this buffered data in order to test its robustness in targeting the problems for which it was designed for.

A Graphical User Interface (GUI) has been designed such that both red and IR channel can be displayed and accessed before and after each signal processing section. This is a feature that usually the commercial pulse oximeters do not provide. The main advantage of accessing the unprocessed data is that the developer can observe the effectiveness of an algorithm on improving the performance of the overall system.

The GUI is also capable of applying different signal processing schemes simultaneously and this is a great advantage in the evaluation process of an algorithm. At the moment system can only display one signal processing scheme but the output of all signal processing techniques can be saved.

The GUI in Figure 4 were developed in MATLAB for processing and displaying the data.

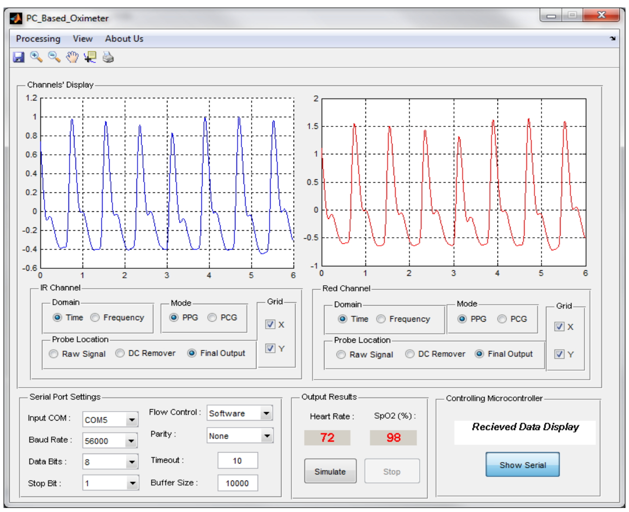


Figure 4. GUI developed in the MATLAB environment

## III. PROCESSING DATA IN MATLAB

As mentioned earlier, the present pulse oximeter system design does all the signal processing using floating point computation in MATLAB. This design approach does not only reduce the time of evaluation of the signal processing algorithm but it also significantly reduces the complexity of the hardware and cost of implementation. To illustrate how this setup can help the developers to rapidly evaluate the effectiveness of their signal processing algorithm individually and in comparison to other available counterparts, two signal processing techniques for the filtering process are presented and applied to real-time data and results of their measurement are compared. Please note that these algorithms were chosen purely for test purposes and they are presented here just to show the flexibility of this hardware setup in dealing with various types of signal processing challenges with different levels of complexity.

The presented signal processing algorithms are:

- Traditional band-pass filtering
- Adaptive filtering

In traditional band-pass filtering a low-pass and a high pass filter is used. The low-pass filter is a 58th order Finite

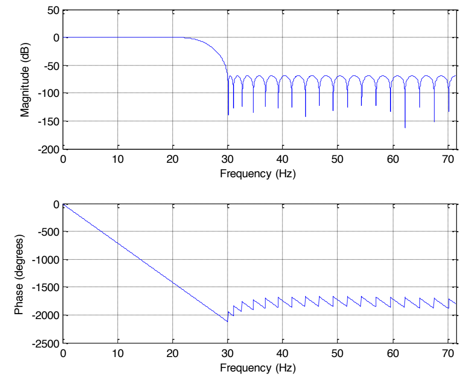


Figure 5. Magnitude and phase response of the low pass filter

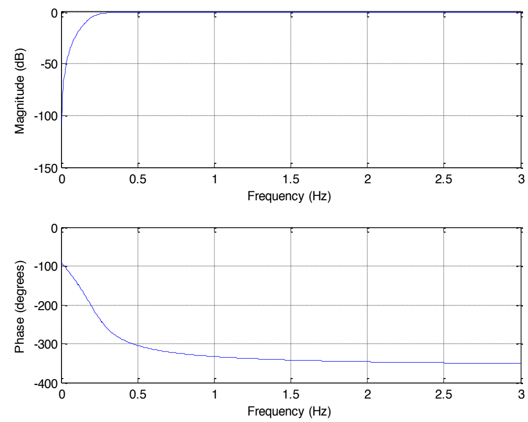


Figure 6. Magnitude and phase response of the high pass

Impulse Response (FIR) filter with a cut-off frequency of 25Hz with a sampling frequency of 250Hz which removes all the out of band noise. The high-pass filter is a 5th order Infinite Impulse Response (IIR) filter with a cuff-off frequency of 0.3 Hz and sampling frequency of 250Hz, which removes the effect of respiratory movements and other low frequency components from the received signal. Figure 5 and 6 show the magnitude and phase response of the low-pass and high-pass filters.

The adaptive digital filtering is another method of processing pulse oximeter data chosen in this study. Compare to the fixed digital filtering, the level of complexity in adaptive digital filtering is considerably higher and therefore it requires more resources for processing the data and more sophisticated chip for hardware implementation. In brief, in this signal processing technique a range of noise references are calculated by choosing an estimated SpO2 value and sweeping it over the whole possible range (1%-100%). These generated reference noise signals are fed to an adaptive noise canceller for further processing. The output Power Spectrum Density (PSD) of the adaptive filter for different SpO2 values is then observed in order to determine the right Spo2 level estimate. The right SpO2 level estimated will have the maximum power among the other estimates of SpO2 in the range [8]-[10].

The mentioned techniques were applied to real-time data in order to observe the flexibility of the setup in implementing

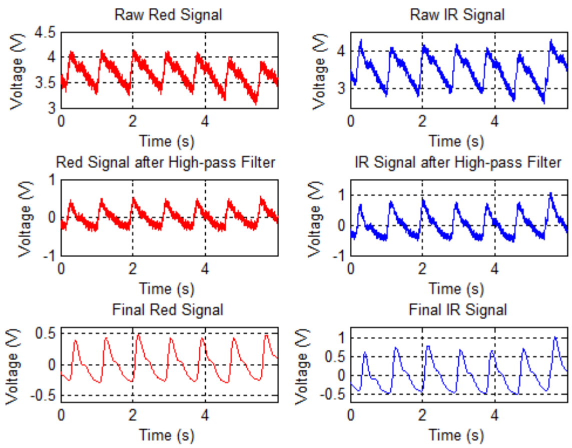


Figure 7. Red and IR signals at different processing stages with fixed band-pass digital filter

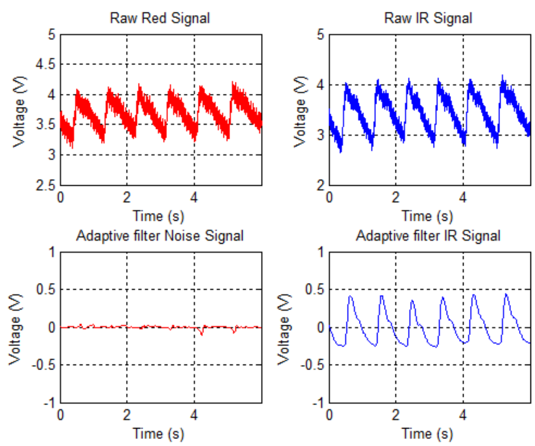


Figure 8. Adaptive digital filter output signals

and evaluating both signal processing techniques. In the next section the results of each algorithm are presented.

## IV. Results and Discussions

The two top plots shown in Figure 7 are the received red and IR signals before processing. The next two plots are the same signals after high-pass digital filtering and the last two plots are the received signals after low-pass filtering. After these two digital filtering stages, all the very low and high frequency noise and interferences were removed from the raw data and the final clean output is achieved. The results of the adaptive digital filtering are shown in Figure 8. The two top plots are the raw red and IR received data sets and the bottom left and right plots are the noise and processed IR signal respectively.

Having access to raw signals are not possible in most of commercial pulse oximeters and this enables the developer to increase the efficiency of their design by properly adjusting the designs' parameters such as the stop-band attenuation or order of the filters.

Another important feature of the MATLAB-based system design is that the signals can be observed in both time and frequency domain without much effort. This feature gives the designers and developers a better understanding about the nature of the pulse oximeter signals, the interferences and other unwanted signals which they are dealing with.

## V. Conclusion

A low cost MATLAB-based pulse oximeter for deployment in novel research and development application is presented in this paper. Two different signal processing techniques with different levels of complexity were implemented to show the flexibility of this system setup for rapid evaluation of an algorithm regardless of its level of complexity. In general this system setup can benefit developers in many different aspects including:

Enabling access to the unprocessed signals and therefore providing this facility to apply different signal processing techniques individually and simultaneously.

Reducing the time and cost for the implementation and evaluation of signal processing algorithms by adopting a MATLAB-based approach and by performing all signal processing inside the host computer.

Finally displaying signals, both processed and raw, in the time and frequency domains is another feature of this pulse oximeter system which gives the designers and developers a clear view about the nature of the signals which they are working with for both the interference and/or desired signal.

All of these mentioned features make our system very attractive for research and development purposes. Moreover, with a little modification to the same structure, it can be used for commercial product and its deployment in challenging hospital and healthcare applications.

## References

[1] G. Mardirossian and R. E. Schneider. Limitations of pulse oximetry Anesth. Prog. 39(6), pp. 194-196. 1992.

[2] P. P. Stetson. *Independent component analysis of pulse oximetry signals*. Presented at Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE.

[3] M. R. Ram, K. V. Madhav, E. H. Krishna, K. N. Reddy and K. A. Reddy. *Use of multi-scale principal component analysis for motion artifact reduction of PPG signals*. Presented at Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE. 2011, .

[4] J. M. Graybeal and M. T. Petterson. *Adaptive filtering and alternative calculations revolutionizes pulse oximetry sensitivity and specificity during motion and low perfusion*. Presented at Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE. 2004.

[5] D. Thompson, A. Wareing, D. Day and S. Warren. *Pulse oximeter improvement with an ADC-DAC feedback loop and a radial reflectance sensor*. Presented at Engineering in Medicine and Biology Society, 2006.

[6] V. Markandey. *Pulse oximeter implementation on the TMS320C5515 DSP medical development kit (rev. A)*. Texas Instrument. June 2010.

[7] J. G. Webster. *Design of Pulse Oximeters*. Medical Science Series.

[8] J. Larsen, T. Jensen, R. Haahr and S. Duun. *Pulse oximetry MATLAB demo software*, October 2009. Available: http://isp.imm.dtu.dk/pulseoximetry.

[9] T. Jensen, S. Duun, J. Larsen, R. G. Haahr, M. H. Toft, B. Belhage and E. V. Thomsen. *Independent component analysis applied to pulse oximetry in the estimation of the arterial oxygen saturation (SpO2)* - a comparative study. Presented at Engineering in Medicine and Biology Society, 2009.

[10] S. J. Barker and S. Morgan. *A laboratory comparison of the newest motion-resistant pulse oximeters during motion and hypoxemia*. Anesth. Analg. 98(5), pp. 2-S2.