

INTEGRATED MASTER IN BIOMEDICAL ENGINEERING  
FACULTY OF SCIENCES AND TECHNOLOGY  
UNIVERSITY OF COIMBRA

**PHOTOPLETHYSMOGRAPHIC LOGGER  
WITH CONTACT FORCE AND  
HYDROSTATIC  
PRESSURE MONITORING**



PEDRO FILIPE VISEU DOS SANTOS  
SEPTEMBER, 2012



INTEGRATED MASTER IN BIOMEDICAL ENGINEERING  
FACULTY OF SCIENCES AND TECHNOLOGY  
UNIVERSITY OF COIMBRA

**PHOTOPLETHYSMOGRAPHIC LOGGER  
WITH CONTACT FORCE AND  
HYDROSTATIC  
PRESSURE MONITORING**

Scientific Supervisors:

PhD Professor Carlos Manuel Bolota Alexandre Correia  
PhD João Manuel Rendeiro Cardoso

Supervisor:

MSc Vânia Maria Gomes de Almeida

Dissertation submitted to the Faculty of  
Sciences and Technology of the  
University of Coimbra in partial  
fulfilment of the requirements for the  
MSc degree in Biomedical Engineering.

PEDRO FILIPE VISEU DOS SANTOS

SEPTEMBER, 2012



Esta cópia da tese é fornecida na condição de que quem a consulta reconhece que os direitos de autor são pertença da Universidade de Coimbra e que nenhuma citação ou informação obtida a partir dela pode ser publicada sem a referência apropriada.

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognize that its copyright rests with University of Coimbra and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.



This work is funded by FEDER, through the *Programa Operacional Factores de Competitividade* – COMPETE – and by National funds through FCT – *Fundação para a Ciência e Tecnologia* – in the frame of “*Centro de Instrumentação*” – *Unidade 217*.







*Aos meus avôs, Carlos e Vitorino,  
que não puderam estar presentes.*



# Agradecimentos

Agradeço, em primeiro lugar, àqueles que mais de perto me apoiaram e aconselharam durante este ano. Aos meus orientadores, Professor Dr. Carlos Correia e Dr. João Cardoso do Grupo de Electrónica e Instrumentação, uma palavra de gratidão pelas sugestões e ensinamentos científicos que me facultaram. Um agradecimento especial à minha supervisora, Vânia Almeida, pelo seu apoio incessante e pelo espírito crítico e capacidade de improviso que muito contribuíram para este trabalho. No fundo, qualidades de uma Engenheira com ‘E’ grande!

Aos colegas (nem sempre) de trabalho, André Cortez, Inês Santos e João Borba, um reconhecimento pela capacidade de marcar de forma intensa cada dia deste último ano. Um grupo de trabalho que certamente deixará saudades.

Aos companheiros Gonçalo Costa, Hugo Ramalho e Tiago Leal, que ao meu lado deram incontáveis braçadas (eu mais do que eles!) dentro e fora da piscina, um agradecimento particular. Uma palavra de apreço também para o amigo de infância, Ricardo Pimentel, pelos muitos momentos partilhados ao longo de todo este tempo.

Ainda um brinde em honra de quem nestes anos me acompanhou: Carla Guerra, Carolina Francisco, Gabriel Campos, Mariana Sequeira e Susete Neiva.

Um muito obrigado àqueles que mais do que colegas de curso foram companheiros de vida, Anastácio Sousa, Danilo Jesus, João Dias, Pedro Vaz e Tobias Correia. A sua amizade incondicional e o seu apoio constante foram fundamentais neste percurso e foram os momentos com eles passados que me permitem afirmar sem hesitação que nestes 5 anos “foi tudo à grande”!

E como não poderia deixar de ser, um enorme agradecimento à minha família, em especial aos meus pais e irmão pelos inúmeros sacrifícios que fizeram ao longo de todos estes anos. É para eles esta última palavra de reconhecimento porque é deles a culpa de eu estar aqui hoje.

Obrigado!



# Abstract

Oxygen saturation ( $SpO_2$ ) is a key parameter for assessing a patient condition, mainly in critical care and anaesthesiology. Its fusion with other physiological measurements, such as pulse wave analysis or arterial blood pressure, may lead to new ways of detecting cardiovascular diseases.

The current work aims to develop a stand-alone pulse oximeter and a photoplethysmographic (PPG) system with measurements of the contact force and height of the finger relatively to the heart. This module, together with other measurements (e.g. electrocardiogram and pulse wave velocity), will be part of a multichannel platform that will be taken to clinical environment.

The project comprises the design of hardware for signal conditioning, as well as the development of firmware to implement in a microcontroller unit (the Arduino™). The latter is used to control the system, acquire data and compute  $SpO_2$ , heart rate ( $HR$ ), contact force and height.

Currently, the system is able to retrieve a stable PPG waveform and compute the  $SpO_2$  in real time through the PPG signal peaks and valleys. Signals acquired in a healthy subject lead to a value of  $SpO_2 = 91.5 \pm 1.1\%$ . Moreover, the subject arm and forearm orientation is retrieved by the use of accelerometers and used to compute the offset of the finger relatively to the heart. Contact force between the finger and the probe is also measured.

**Keywords:** pulse oximetry, photoplethysmography, contact force, accelerometry, Arduino™.



# Resumo

A saturação de oxigénio ( $SpO_2$ ) é um parâmetro indispensável para avaliar a condição de um paciente, principalmente em cuidados intensivos e em anestesiologia. A sua fusão com outras medidas fisiológicas, como a análise da onda de pulso ou a pressão arterial, poderá levar a novas formas de detectar doenças cardiovasculares.

O presente trabalho pretende desenvolver um oxímetro de pulso automático e um sistema de fotopletismografia (PPG) com medição da força de contacto e da altura do dedo em relação ao coração. Este módulo, juntamente com outras medidas (e.g. electrocardiograma e velocidade da onda de pulso), fará parte de uma plataforma multicanal que será levada para ambiente clínico.

O projecto inclui o design do *hardware* para condicionamento de sinal, bem como o desenvolvimento do *firmware* a implementar num microcontrolador (o Arduino™). Este é usado para controlar o sistema, adquirir dados e calcular a  $SpO_2$ , o ritmo cardíaco ( $HR$ ), a força de contacto e a altura.

Actualmente, o sistema é capaz de adquirir uma onda de PPG estável e de calcular a  $SpO_2$  em tempo real através dos picos e vales do sinal de PPG. Sinais adquiridos em sujeitos saudáveis levaram a um valor de  $SpO_2 = 91.5 \pm 1.1\%$ . Além disso, a orientação do braço e antebraço do sujeito é estimada através do uso de acelerómetros e usada para calcular o desnível do dedo em relação ao coração. A força de contacto entre o dedo e a ponta de prova é igualmente medida.

**Palavras-chave:** oximetria de pulso, fotopletismografia, força de contacto, acelerometria, Arduino™.





# Contents

<b>AGRADECIMENTOS</b> .....	<b>ix</b>
<b>ABSTRACT</b> .....	<b>xi</b>
<b>RESUMO</b> .....	<b>xiii</b>
<b>LIST OF FIGURES</b> .....	<b>xix</b>
<b>LIST OF TABLES</b> .....	<b>xxv</b>
<b>LIST OF ACRONYMS</b> .....	<b>xxvii</b>
<b>CHAPTER 1 INTRODUCTION</b> .....	<b>1</b>
1.1 MOTIVATION.....	2
1.2 MAIN CONTRIBUTIONS .....	2
1.3 PREVIOUS WORK .....	3
1.4 TEAM .....	3
1.5 CONTENTS BY CHAPTER .....	4
<b>CHAPTER 2 THEORETICAL BACKGROUND</b> .....	<b>5</b>
2.1 PHYSIOLOGICAL INTRODUCTION .....	6
2.1.1 Oxygen needs.....	6
2.1.2 Respiration.....	6
2.1.3 Transport of gases in blood – the haemoglobin .....	7
2.2 MEASURING OXYGEN SATURATION .....	8
2.2.1 Optical properties of the tissues .....	9
2.2.2 Photoplethysmography .....	12
2.2.3 Pulse oximetry .....	13
2.3 HEIGHT MEASUREMENT .....	17
2.3.1 Module architecture .....	17
2.3.2 Working principles .....	19
2.3.3 Height computing .....	20
2.4 CONTACT FORCE MEASUREMENTS .....	21
<b>CHAPTER 3 HARDWARE</b> .....	<b>25</b>
3.1 PULSE OXIMETER .....	26
3.1.1 LED driver module .....	26
3.1.2 Photodetection module .....	28
3.1.3 Sample-and-Hold.....	33
3.1.4 Signal conditioning .....	34

3.1.5	Peak and valley detector.....	39
3.2	ACCELEROMETERS .....	47
3.3	FORCE SENSOR .....	47
3.4	MICROCONTROLLER – THE ARDUINO™ .....	50
3.4.1	Processor .....	51
3.4.2	I/O Ports .....	51
3.4.3	Timer/Counters .....	52
3.4.4	Interrupts .....	52
3.4.5	Communication .....	52
3.4.6	Shields.....	53
<b>CHAPTER 4</b>	<b>FIRMWARE.....</b>	<b>55</b>
4.1	LEDS TIMING .....	56
4.2	PHOTOPLETHYSMOGRAM .....	58
4.3	SpO <sub>2</sub> COMPUTATION .....	61
4.4	HEART RATE COMPUTATION .....	63
4.5	GAIN AND LEVEL SHIFT CONTROL.....	64
4.5.1	Approaches.....	64
4.5.2	Controller .....	66
4.6	HEIGHT MEASUREMENT.....	69
4.7	CONTACT FORCE MEASUREMENT .....	71
4.8	COMMUNICATION.....	72
4.8.1	I <sup>2</sup> C .....	72
4.8.2	Serial .....	73
4.9	DATA STORAGE / SENDING.....	74
<b>CHAPTER 5</b>	<b>RESULTS.....</b>	<b>75</b>
5.1	SYSTEM VERSIONS.....	76
5.1.1	Version 1 .....	76
5.1.2	Version 2.....	77
5.1.3	Version 3.....	77
5.2	LED DRIVER.....	80
5.3	PHOTODETECTION AND SAMPLE-AND-HOLD.....	81
5.4	SIGNAL CONDITIONING .....	83
5.4.1	DC conditioning .....	83
5.4.2	AC conditioning .....	86
5.5	PEAK AND VALLEY DETECTOR.....	87
5.6	PULSE OXIMETRY .....	91
5.7	ARDUINO™ ACQUISITION.....	93
5.8	HEIGHT MEASUREMENT.....	95
5.9	CONTACT FORCE MEASUREMENT .....	98
5.9.1	Calibration setup .....	98

---

5.9.2	Calibration curve .....	100
5.9.3	Results .....	106
5.10	GRAPHICAL USER INTERFACE .....	109
<b>CHAPTER 6</b>	<b>FINAL REMARKS .....</b>	<b>111</b>
6.1	CONCLUSIONS .....	112
6.2	FUTURE WORK .....	114
6.2.1	Power consumption.....	114
6.2.2	LED driver .....	114
6.2.3	Digipot controller.....	115
6.2.4	Peak and valley detector .....	115
6.2.5	SpO <sub>2</sub> computation .....	115
6.2.6	Accelerometers .....	115
6.2.7	Force sensor .....	116
6.2.8	Arduino™ / computer communication .....	116
<b>APPENDICES</b> .....	<b>117</b>	
APPENDIX A – PPG AC CONDITIONING CUTOFF .....	119	
APPENDIX B – PEAK DETECTOR TRIGGER FILTERING CUTOFF .....	121	
APPENDIX C – DC CONDITIONING CONTROLLER – CASES OF INTEREST .....	127	
<b>REFERENCES</b> .....	<b>129</b>	



# List of Figures

Figure 2.1 – Oxyhaemoglobin dissociation curve. Conformational changes allow the haemoglobin to bind $O_2$ in the lungs where partial pressure is high (red bar) and release it in the tissues, where partial pressure is low (blue bar). Adapted from [20].	8
Figure 2.2 – Absorption spectra of biological tissues from 0.1 to 10 $\mu\text{m}$ [28].	10
Figure 2.3 – Extinction coefficients of the four most common species of haemoglobin at the wavelengths of interest on pulse oximetry. Adapted from [26].	11
Figure 2.4 – Absorbed and transmitted light in pulse oximetry. A constant amount of attenuation is due to the muscle, bone, skin and other tissue plus the venous and the non-pulsatile arterial blood. The varying light attenuation is due to the arterial pulsatile blood: during the systole the light transmitted is minimal ( $I_L$ ) while at the end of the diastole it is maximal ( $I_H$ ). Adapted from [24].	13
Figure 2.5 – Beer Lambert’s law in pulse oximetry. The incident light is attenuated by the DC component (muscle, bone, skin and other tissue plus the venous and non-pulsatile arterial blood) over an optical path length of $d_{DC}$ . At the end of the diastole, light intensity is $I_H$ and the optical path length through the arteries is $d_{min}$ ; during the systole, light intensity decreases to $I_L$ and the optical path length through the arteries increases to $d_{max}$ . On air, light absorption is negligible. Adapted from [24].	14
Figure 2.6 – Theoretical curve for pulse oximetry: oxygen saturation ( $SpO_2$ ) vs. Ratio of Ratios ( $R_{OS}$ ). Curve changes with different wavelengths: 660nm and 905 nm were used.	16
Figure 2.7 – Accelerometers positioning in the arm and forearm and their axes orientation. The arm and forearm lengths are indicated, as well as the increase in height due to the orientation of these segments ( $h_2$ and $h_1$ , respectively). The distance between the heart and the shoulder is also shown. All this lead to the wanted variable: the total height between the finger and the heart.	18
Figure 2.8 – MEMS Accelerometer: movable and fixed plates make capacitors; acceleration produces a change in distance between plates and thus changes the capacitance. Adapted from [44].	19
Figure 2.9 – Representation of the arm orientation ( $\theta$ ) related to the earth gravity ( $g$ ) and the accelerometer y-axis output ( $y$ ). Acceleration projection is also shown ( $a_y$ ).	20
Figure 2.10 – FSR <sup>®</sup> construction: the sensor has a layer with electrodes and a layer with a semiconductor material that touch each other when force is applied over the sensor [50].	22
Figure 2.11 – FSR <sup>®</sup> resistance vs. force for the range of the sensor [50].	22

Figure 3.1 – Driver to polarize red and infrared LEDs alternately with controlled current. <i>LED_power1</i> and <i>LED_power2</i> define which LED is ON and which is OFF at each time. Adapted from [16].	27
Figure 3.2 – Photodiode operation modes: photoconductive (left) and photovoltaic (right). Adapted from [53].	29
Figure 3.3 – Current-voltage characteristics for a typical photodiode: curve ① shows the response for no incident light; light intensity is increasing from curve ② to ③. Adapted from [54].	29
Figure 3.4 – Basic transimpedance amplifier configuration to measure photodiode output current. Adapted from [56].	30
Figure 3.5 – Differential photodiode current sensing: transimpedance amplifier for pulse oximetry. Adapted from [24].	32
Figure 3.6 – Sample-and-hold basic circuit: two buffers and a capacitor. Adapted from [60].	33
Figure 3.7 – LF398 typical performance characteristics: acquisition time, hold step and output droop rate [62].	34
Figure 3.8 – Butterworth filter: generic Sallen-Key topology [71].	35
Figure 3.9 – DC conditioning circuitry: a digital potentiometer is used to set gain and level shift of each channel.	36
Figure 3.10 – Peak detector: Basic configuration (left) and two stage active peak detector (right) Adapted from [74].	40
Figure 3.11 – Peak and valley detector module. Peak detector circuitry (B), analogue switch used for capacitor discharge (C) and inverting op-amp for signal inversion required for valley detection (A).	41
Figure 3.12 – Differentiator circuit [74].	44
Figure 3.13 – LM311 configuration: positive derivative (left) and negative (right) positive identification.	45
Figure 3.14 – LM555 monostable mode for <i>one-shot</i> signal used for peak detector reset. Adapted from [80].	46
Figure 3.15 – FSR <sup>®</sup> voltage divider. Adapted from [50].	48
Figure 3.16 – FSR <sup>®</sup> current-to-voltage converter. Adapted from [50].	48
Figure 3.17 – FSR <sup>®</sup> conditioning circuit to improve the output linearity. Adapted from [83].	49
Figure 3.18 – Arduino <sup>™</sup> : the Mega 2560 board with the ATmega2560 (left) and the IDE window with a piece of a used sketch (right).	51
Figure 3.19 – Connection of I <sup>2</sup> C-bus devices to the I <sup>2</sup> C bus. Configuration used in the system, with the Arduino <sup>™</sup> acting as master and the accelerometers and digital potentiometers as slaves.	53
Figure 4.1 – LEDs lighting and signal sampling scheme. Timer/Counter increases in normal mode and has 4 interrupt counter values (middle). In the middle and at the end of the Timer/Counter, OCR5B and MAX interrupt flags whose routines are toggling LEDs power lines are triggered (bottom). In the middle of each LED ON period (OCR5A and OCR5C), interrupts whose routines are to sample the respective LED signal are triggered (top).	59

Figure 4.2 – PPG signal reconstruction algorithm. The output of the instrumentation amplifier is sampled and then the signal is reconstructed in order to retrieve its original amplitude.....	62
Figure 4.3 – Calibration procedure in the first version of the system. ....	65
Figure 4.4 – Representation of the desired signal ranges and threshold values for gain and offset shift update.....	67
Figure 4.5 – Flowchart of a possible approach of a controller to adjust signal gain and DC level.....	68
Figure 4.6 –I <sup>2</sup> C interface general description for writing (top) and reading (bottom) operations. Adapted from [81].....	73
Figure 5.1 – Current version of the system. The Arduino™ Mega 2560 is implemented in the board along with all the remaining hardware modules: LED driver, transimpedance, sample-and-hold, AC and DC signal conditioning, peak and valley detector, force sensor conditioning (below the Arduino™) and accelerometer conditioning (below the Arduino™).....	78
Figure 5.2 – System front panel: external sensors connectors, indicating LEDs and reset button. ....	79
Figure 5.3 –Accelerometers are attached to a wristlet and an armlet and placed in a stable manner in the subject forearm and arm. ....	79
Figure 5.4 –Force sensor is attached to the inferior side of the PPG probe centred with its axis. .	80
Figure 5.5 – LED driver signals: MCU digital output (top) pins <i>LED_power2</i> (shown in red) and <i>LED_power1</i> (blue); and voltages across the LEDs terminals (bottom): red anode and infrared cathode (shown in red) and red cathode and infrared anode (blue). Signals were acquired with the NI-USB6210 at 11 kHz. ....	81
Figure 5.6 – Transimpedance module and sample-and-hold: the voltage output of the transimpedance module, corresponding to the photodiode signal is visibly modulated (top). The modulation clearly matches each of the split channels (bottom): red channel (in red) and infrared channel (in blue). ....	82
Figure 5.7 – Sample-and-Hold: the signal from the transimpedance module is split into two channels – red (top) and infrared (bottom).....	83
Figure 5.8 – PPG DC+AC component: red channel acquired with a gain of 20.4 and an offset voltage of 742 mV (top); infrared channel acquired with a gain of 28.8 and an offset voltage of 1.07 V (bottom). Both signals were acquired with the NI-USB 6210 with a sampling frequency of 500 Hz. On top of each channel, a detail of two pulses is shown with the PPG features: systolic peak (SP), dicrotic peak (DP), dicrotic notch (DN) and pulse foot (PF).....	84
Figure 5.9 – Reconstruction of the PPG DC+AC component of red channel (top) and infrared (bottom) from the digipot voltage and resistance values.....	85
Figure 5.10 – PPG AC component of red channel (top) and infrared channel (bottom). Signals were acquired with the NI-USB 6210 at 500 Hz. On top of each channel, a detail of two pulses is shown with the PPG features: systolic peak (SP), dicrotic peak (DP), dicrotic notch (DN) and pulse foot (PF).....	86
Figure 5.11 – Peak detector with intermediate steps. Raw AC component (in blue) and subsequent filtering (in red) used as triggering (row 1); derivative of the filtered	

trigger (row 2); LM311 comparator for valley sampling (row 3) and the corresponding LM555 for peak reset (row 4); LM311 comparator for peak sampling (row 5) and the respective LM555 valley reset (row 6); peak detection input (in black) and output (in blue) for red channel (row 7) and valley detection input (in black) and output (in red) for red channel (row 8); the same for infrared peak (row 9) and valley (row 10) detection. Dashed vertical lines show the instant when the interrupt signal for sampling is given to the MCU (descendant flank of the respective comparator output). .....	88
Figure 5.12 – Peak and valley detection. Peak detection input (in black) and output (in blue) are shown for the red (row 1) and infrared channel (row 3); valley detection input (in black) and output (in red) are also shown for red (row 2) and infrared (row 4) channels.....	90
Figure 5.13 – PPG signal peaks (black stars) and valleys (green stars) used for SpO <sub>2</sub> computation.....	91
Figure 5.14 – SpO <sub>2</sub> values computed from the peak and valley values of the PPG signal. ....	92
Figure 5.15 – PPG DC+AC component for red channel (top) and infrared channel (bottom). Signals were acquired with the Arduino™ with a sampling frequency of 500 Hz. Many oscillations in the level voltage can be observed. ....	93
Figure 5.16 – PPG AC component for red channel (top) and infrared channel (bottom). Signals were acquired with the Arduino™ at 500 Hz. The signal is much more stable than the DC+AC component. ....	94
Figure 5.17 – Accelerometers 3-axis outputs for the forearm (top) and arm (bottom). The Y axis (shown in green) is aligned with the arm and forearm axis and is the best measure for height estimation. ....	95
Figure 5.18 – PPG signal (top and middle) with different finger heights (bottom). Red channel (shown in red) and infrared channel (shown in blue) are illustrated both for the DC+AC (top) and AC (middle) components. Vertical dashed lines show the instants when the height was increased. On top of DC+AC component, one pulse is depicted for each height level. ....	96
Figure 5.19 – SpO <sub>2</sub> values computed from the signal with variations in finger height shown in Figure 5.18. Dashed lines show the instants of height change. Substantial variations are seen, especially after the height change and before signal stabilization in that step. ....	97
Figure 5.20 – FSR <sup>®</sup> dynamic calibration setup using the <i>Zaber</i> T-LA 28A linear actuator. The FSR <sup>®</sup> is fixed and the actuator pushes the spring against it, producing a determined force according to the displacement. ....	98
Figure 5.21 – Schematic of the FSR <sup>®</sup> dynamic calibration experimental setup. The linear actuator moves and compresses the spring against the FSR <sup>®</sup> . As the displacement and force constant are known, the applied force can be calculated. ....	99
Figure 5.22 – FSR <sup>®</sup> output voltage of the calibration procedure for the voltage divider configuration with again resistance R <sub>G</sub> =56 kΩ (left) and with the current-to-voltage converter with a gain resistance R <sub>G</sub> = 5.6 kΩ (right). Values below zero mean that no contact with the sensor was made. ....	100
Figure 5.23 – Mean (black line) and standard deviation (red error bars) for one of the experimental tests (current-to-voltage converter with a gain resistance R <sub>G</sub> = 5.6	



- k $\Omega$ ) with the actuator tip starting always in the same position. Values below zero mean that no contact with the sensor was made..... 101
- Figure 5.24 – FSR<sup>®</sup> response for different misaligned contact positions: the raw results (left) show a large discrepancy; when the activation linear is superimposed for all trials (in 2 mm of step position), better results are achieved (right). The current-to-voltage converter with  $R_G = 7.67$  k $\Omega$  was used. .... 102
- Figure 5.25 – Mean (black line) and standard deviation (red error bars) for one of the experimental tests (current-to-voltage converter with a gain resistance  $R_G = 7.67$  k $\Omega$ ) with the actuator tip at different initial positions. .... 103
- Figure 5.26 – Calibration results for FSR<sup>®</sup> turn-on threshold. The dashed line shows the voltage threshold defined by visual inspection of the results..... 103
- Figure 5.27 – FSR<sup>®</sup> calibration results: mean response (black) and best quadratic fitting curve (red) for the current-to-voltage converter with a gain resistance  $R_G = 5.6$  k $\Omega$ ..... 105
- Figure 5.28 – FSR<sup>®</sup> output of the contact force between the sensor probe and the subject finger: acquired signal (top) and filtered signal to remove signal spikes (bottom). . 106
- Figure 5.29 – PPG signal (top and middle) and corresponding contact force (bottom). Both DC+AC component (top) and AC component (middle) are shown for red channel (illustrated in red) and infrared channel (illustrated in blue). The instants when the contact force is altered are shown with vertical dashed lines. At the top of the DC+AC component, a detail of two PPG pulses for each force step are shown. .... 107
- Figure 5.30 – SpO<sub>2</sub> values computed from the signal with variations in contact force shown in Figure 5.29. Considerable variations are present, which indicate that contact force takes an active role in PPG signal quality. Dashed lines indicate the instants in which force contact changes. .... 108
- Figure A.1 – PPG signal AC conditioning – Results of the DC+AC component high-pass filtering with a cutoff frequency of 0.01 Hz (top), 0.05 Hz (middle) and 0.1 Hz (bottom). DC+AC original component (red), filtered AC component (blue) and the difference between them (black) are shown..... 119
- Figure A.2 – PPG signal AC conditioning – Results of the DC+AC component high-pass filtering with a cutoff frequency of 0.2 Hz (top), 0.3 Hz (middle) and 0.4 Hz (bottom). DC+AC original component (red), filtered AC component (blue) and the difference between them (black) are shown..... 120
- Figure B.1 – PPG extrema identification – Results obtained with a cutoff frequency of 0.5 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed. .... 121
- Figure B.2– PPG extrema identification. Results obtained with a cutoff frequency of 1.0 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed. .... 122
- Figure B.3 – PPG extrema identification. Results obtained with a cutoff frequency of 1.5 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered

signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed. ....123

Figure B.4 – PPG extrema identification. Results obtained with a cutoff frequency of 2.0 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed. ....124

Figure B.5 – PPG extrema identification. Results obtained with a cutoff frequency of 2.5 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed. ....125

Figure B.6 – PPG extrema identification. Results obtained with a cutoff frequency of 3.0 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed. ....126

Figure C.1 – DC conditioning: rules for gain and offset setting. A – decrease gain and shift upwards; B – shift downwards; C – decrease gain and shift upwards; D – shift upwards; E – increase gain and shift downwards; F – increase gain and shift downwards; G – decrease gain and shift upwards; H – increase gain and shift downwards. ....127

# List of Tables

Table 1.1 – Team members of the current work.....	4
Table 2.1 –Molar extinction coefficients for <i>Hb</i> , <i>HbO<sub>2</sub></i> , <i>MetHb</i> and <i>COHb</i> at $\lambda = 660$ nm and $\lambda = 905$ nm [30].....	12
Table 4.1 – Timer/Counter 5 registers used for interrupt routines to control LEDs lighting and signal sampling timing. ....	57
Table 4.2 – ADPS [2:0]: ADC Prescaler Select Bits. These bits determine the division of the system clock to set the ADC clock [92].....	60
Table 4.3 – AD5252 I <sup>2</sup> C 7-bit address, according to AD1 and AD0 pin voltages [73].....	66
Table 4.4 – MMA8451Q I <sup>2</sup> C 7-bit address, according to SA0 pin voltage.....	71
Table 5.1 – Maximum standard deviation of the 25 trials with 400 displacement points for different values of R <sub>G</sub> . Results for both configurations are shown. The FSR <sup>®</sup> response was previously normalized.....	101
Table 5.2 – FSR <sup>®</sup> calibration results: turn-on threshold determination.....	104



# List of Acronyms

ABP	Arterial Blood Pressure
ACK	Acknowledge
ADC	Analogue-to-Digital Converter
ALU	Arithmetic Logic Unit
AREF	Analogue Reference
ATP	Adenosine Triphosphate
CI	Instrumentation Centre
CO <sub>2</sub>	Carbon Dioxide
COHb	Carboxyhaemoglobin
CTC	Clear Timer on Compare Match
Digipot	Digital Potentiometers
FSR <sup>®</sup>	Force Sensing Resistor <sup>®</sup>
GEI	Electronics and Instrumentation Group
GUI	Graphical User Interface
Hb	Reduced Haemoglobin
HbO <sub>2</sub>	Oxyhaemoglobin
HR	Heart Rate
I <sup>2</sup> C	Inter-Integrated Circuit
IC	Integrated Circuit
IDE	Integrated Development Environment
IR	Infrared
LEDs	Light Emitting Diodes
LSB	Least Significant Bit
MCU	Microcontroller Unit
MetHb	Methaemoglobin
O <sub>2</sub>	Oxygen
PCB	Printed Circuit Board
PPG	Photoplethysmography
P <sub>TM</sub>	Transmural Pressure
PWM	Pulse-Width Modulation
RISC	Reduced Instruction Set Computers
RMS	Root Mean Square
R <sub>OS</sub>	Ratio of Ratios
S/H	Sample-and-Hold
SCL	Serial Clock Line
SDA	Serial Data Line
SNR	Signal-to-Noise Ratio
SPI	Serial Peripheral Interface
SpO <sub>2</sub>	Oxygen Saturation
TWI	2-wire Serial Interface
USART	Asynchronous Serial Receiver and Transmitter
UV	Ultraviolet



# Chapter 1

## Introduction

*The first chapter of this thesis aims to introduce this work, explaining the motivations behind it and the goals it purposes to achieve. Afterwards, the work developed previously under this field is referred. Subsequently, the team of Electronics and Instrumentation Group is presented. Finally, the thesis structure is outlined.*

## 1.1 Motivation

The lack of oxygen to the brain or heart is the ultimate reason for a patient death. Therefore, monitoring of oxygen supply has always been a focused field of investigation. This motivated the search for methods for non-invasive and continuous monitoring of oxygen saturation ( $SpO_2$ ), which was achieved with the development of pulse oximetry [1]. Consequently, the technique has spread rapidly from the labs to the hospitals. Currently, it is a standard practice in many clinical fields such as anaesthesia, intensive care and neonatal care units, along with its use in many diagnostic tests. Nowadays, it is also gaining importance in home care. Developments in microprocessors, as well as in light-emitting diodes (LEDs) and photodiodes, now with improved accuracy and reduced size and cost, were a strong driving force to the expansion of pulse oximetry [2].

This work emerged with the request of cardiologists from the *Hospitais da Universidade de Coimbra* (H.U.C.) for a way of non-invasively measure the arterial blood pressure (ABP) over a large period of time (e.g. 24 hours) using a photoplethysmographic (PPG) system. This has been a studied field, especially in an ongoing investigation of the Massachusetts Institute of Technology (MIT) [3-7] but also in other groups [8-10] and moreover, there have been registered patents under this theme [11].

Moreover, a new project called “New Methodology for Hemodynamic Assessment. Clinical Validation” is about to start at the *Electronics and Instrumentation Group* (GEI) and aims to be a new paradigm in hemodynamic assessment. It will consist of a multichannel systems that incorporate some conventional sensors (including pulse oximetry and photoplethysmography) allied with pioneer sensors developed at the group [12-15].

## 1.2 Main contributions

The interest in the multichannel platform defined a new goal for this work: to develop a stand-alone photoplethysmographic system and pulse oximeter that would be incorporated in a multichannel system and that could, in the future, be used to measure the ABP. A commercial probe will be used at finger site, but all the signal condition and acquisition tasks are to be done during the present work. The system will also comprise



the contact force between the finger and the probe and the finger height measurements. As a data acquisition system and control unit, an Arduino™ will be used.

The current work is focused mainly in the hardware. Though, some algorithms for oxygen saturation and heart rate computation will be developed. Those algorithms are to be implemented in a microcontroller unit (MCU) for real time assessment. Nevertheless, data is to be sent to MATLAB for saving and eventually further analysis.

In the future, the system is expected to be implemented in the multichannel system that is being developed and will be taken to clinical environment for hemodynamic studies.

### 1.3 Previous work

The development of a stand-alone pulse oximeter has been an addressed issue at GEI. Namely, two projects were done in this area: “Projecto de um Sistema Digital de Medida para Aplicações Biomédicas” [16] for the Licence degree in Engineering Physics of Rita Pereira in 1995 and “Development of a Stand-Alone Pulse Oximeter” for the MSc degrees in Biomedical Engineering of Ana Rita Domingues [17] and Sérgio Brás [18] in 2009.

These works provided knowledge about the principles of operation of pulse oximetry. However, implementation in a MCU was not fully accomplished. Besides, the developed system was not in a phase of being capable of making  $SpO_2$  calculations. So, the present work aims to acquire all the data with a microprocessor unit (MCU) and reach results that may allow for its use in clinical validation, provided the controller is programmed.

### 1.4 Team

The project was developed at the *Electronics and Instrumentation Group* (GEI), which is part of the *Instrumentation Centre* (CI), at the Physics Department of the University of Coimbra. The main scientific interests of the group are the hemodynamics, bioimpedance and eye-tracking. In the particular case of hemodynamics, pulse wave characterization, augmentation index, pulse wave velocity, pulse oximetry and blood perfusion in microcirculation are the main focus. Piezoelectric, optical and accelerometric sensors are currently employed for these measurements. Table 1.1 shows the team members associated with this work.

Table 1.1 – Team members of the current work.

Team Members	Contribution <i>Main area of research</i>
PhD Professor Carlos Correia PhD João Cardoso	Scientific and Technical advisors
PhD student Vânia Almeida	Scientific and technical supervisor <i>Study of hemodynamic parameters</i>
MSc student Pedro Santos	<i>Pulse Oximetry</i>

## 1.5 Contents by chapter

The present thesis summarises all the work done during this year. It is divided in six chapters.

In Chapter 2 – **Theoretical Background**, the fundamentals for the understanding of photoplethysmography (PPG) technique are described. The mathematical formulas that allow for the  $SpO_2$  computation are also derived. Finally, a summary of contact force and height measurement is also done.

Chapter 3 – **Hardware** presents an overview of the developed hardware, including the PPG circuit, as well as the accelerometric and force sensor module. Special interest will be given to the signal conditioning techniques employed.

In Chapter 4 – **Firmware**, synchronisation between the LEDs switching and the signal acquisition is described. Besides, the control tasks of some parts of the conditioning circuit are also presented. Finally, computations used for  $SpO_2$ , heart rate ( $HR$ ), height and contact force measurements are also referred.

Chapter 5 – **Results** presents the results of the complete system, from details of intermediate steps of signal conditioning to the final  $SpO_2$  computation. Results from the accelerometry module as well as the contact force (and its calibration process) are also shown.

Chapter 6 – **Final Remarks** summarises the work developed and the results achieved. Possible approaches to proceed with this work are also presented.

# Chapter 2

## Theoretical Background

*In this chapter, the theoretical fundamentals needed for understanding the goal and applications of this work are overviewed. First, a physiological introduction will be done, presenting the human body need for oxygen and how it is carried from the atmosphere to the cells. Special focus will then be given to haemoglobin, the protein responsible for oxygen transport in blood stream. An understanding of light attenuation in the tissues, mainly due to haemoglobin, will lead to the concept of pulse oximetry and consequently, to oxygen saturation measurements. These physical concepts and their mathematical derivation are also described. Finally, a general idea of the force sensor and working principles of accelerometers will be given.*

## 2.1 Physiological introduction

The human being has a permanent need for oxygen ( $O_2$ ), as it is the ultimate need for cell functioning. As an extremely complex but yet well adapted “machine”, the human body has developed a way of effectively taking  $O_2$  from its environment and deliver it to the cells, where it is used in metabolic reactions. In the following three subsections, these concepts will be revised.

### 2.1.1 Oxygen needs

Living organisms require energy for cellular movements (e.g. muscle contraction), active transport of molecules and ions and synthesis of biomolecules. As *chemotrophus*, the human being obtains energy from the oxidation of electron donors, such as carbohydrates, lipids and proteins.

Metabolic paths start with glycolysis. Then, if  $O_2$  is available, i.e. in aerobic conditions, pyruvate oxidation, the citric acid cycle (Krebs cycle) and respiratory chain (electron transport chain) occur, being carbon dioxide ( $CO_2$ ) and water the end products. When no  $O_2$  is available, i.e. in anaerobic conditions, the pyruvate is directly metabolized to lactate or ethanol. Both processes transform part of the energy into adenosine triphosphate (ATP), the cell common energy currency. However, aerobic paths produce 18 times more ATP, and that is why  $O_2$  is so essential for metabolism. Without it, life as we know it would not be possible [19-21].

### 2.1.2 Respiration

For cellular respiration to be carried out, a constant  $O_2$  concentration must be supplied to the tissues and the produced  $CO_2$  must be eliminated. This is the role of respiratory and circulatory systems.

Due to lung contraction and expansion, the air flows through the respiratory tree to the alveoli and from there to the outside.

At the alveoli, the  $O_2$  partial pressure of the gas mixture is higher than the partial pressure in blood of alveolar capillaries. This way,  $O_2$  tends to diffuse across the alveolar membrane and dissolve into the blood. On the contrary, the  $CO_2$  partial pressure is higher

in blood, which means that  $CO_2$  tends to cross the membrane from the blood side to the alveoli.

This is the same force that drives gas exchanges between blood and cells. Since the cells consume  $O_2$ , the capillary partial pressure is higher than the tissue partial pressure. This forces  $O_2$  to diffuse from the blood to the tissue. Conversely, the  $CO_2$  partial pressure inside the cells is increased due to metabolism. Therefore,  $CO_2$  diffuses from the tissue into the capillaries [22, 23].

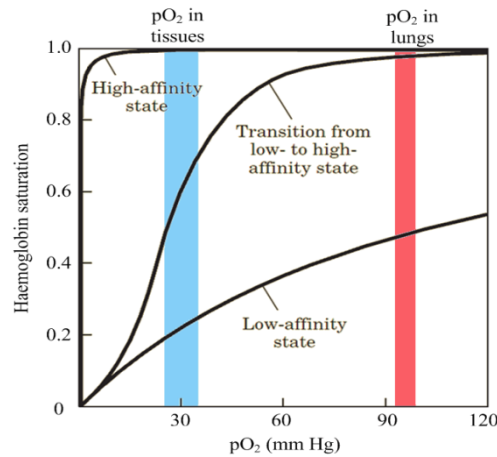
### 2.1.3 Transport of gases in blood – the haemoglobin

Since  $O_2$  cannot diffuse in tissues over distances greater than a few millimetres, it is carried from the alveoli to the tissues in the blood stream. However, the solubility of  $O_2$  in the plasma is very low. Therefore, a method other than plasma dissolution must be used to carry it. That method is an oxygen-binding protein contained within the red blood cells – the haemoglobin [23].

The role of haemoglobin is to bind oxygen in the lungs, where the  $O_2$  partial pressure is  $pO_2 = 95 \text{ mm Hg}$ , and release it in the tissues, where  $pO_2 = 35 \text{ mm Hg}$ . A high-affinity protein would bind  $O_2$  efficiently in the lungs, but would not release a significant quantity in the tissues. In contrast, a low-affinity protein would release the  $O_2$  in the tissues, but would not bind a considerable quantity in the lungs [19].

However, haemoglobin has a way of dealing with this. It has two states. Initially it is in the low-affinity state (called T state) but as more  $O_2$  molecules are bound, it changes to the high-affinity conformation (called R state). This way, haemoglobin has a sigmoid curve for oxygen binding, also called oxyhaemoglobin dissociation curve (shown in Figure 2.1). The first  $O_2$  molecule binds weakly to haemoglobin, because it is in the T state. This binding leads to conformational changes ( $T \rightarrow R$ ) that are communicated to the adjacent subunits, making the next binding easier. This process repeats and when the last (forth) molecule binds, haemoglobin is already in the high-affinity state [19, 20].

Different species of haemoglobin exist in human blood. The most important ones are *oxyhaemoglobin* ( $HbO_2$ ) and *reduced haemoglobin* ( $Hb$ ). The first one is the haemoglobin saturated with oxygen, while the second one is the haemoglobin without bound oxygen.



**Figure 2.1 – Oxyhaemoglobin dissociation curve. Conformational changes allow the haemoglobin to bind  $O_2$  in the lungs where partial pressure is high (red bar) and release it in the tissues, where partial pressure is low (blue bar). Adapted from [20].**

These are called the functional haemoglobin species, since they are the ones that fulfil the haemoglobin main purpose – oxygenation of tissues. Functional oxygen saturation (functional  $SO_2$ ) is a measure used in clinical practice and is defined by equation (2.1).

$$\text{Functional } SO_2 = \frac{HbO_2}{Hb + HbO_2} \times 100 = \frac{c_{HbO_2}}{c_{Hb} + c_{HbO_2}} \times 100 \quad (2.1)$$

where  $HbO_2$  and  $Hb$  are the amount of oxyhaemoglobin and reduced haemoglobin, respectively and  $c_{HbO_2}$  and  $c_{Hb}$  are their concentrations.

In addition to the functional haemoglobin, other species are present in blood, such as methaemoglobin ( $MetHb$ ), which has iron in  $Fe^{3+}$  oxidized state, and carboxyhaemoglobin ( $COHb$ ), that results from binding of carbon monoxide. These two forms are called dysfunctional haemoglobin because they do not bind  $O_2$  [24].

## 2.2 Measuring oxygen saturation

Oxygen saturation is a critical physiological parameter. It has been essential for many clinical applications, such as during anaesthesia, surgery, critical care, recovery or hypoxemia screening. It is also very used in neonatal care. Its most important clinical uses are present in [25].

There have been several methods for monitoring arterial oxygen content. The first one used the Clark electrode either *in vitro* or *in vivo* (implemented in a catheter). Spectrophotometry was also used, allowing for measurements of light absorbed for each wavelength, which is dependent of the molecular extinction coefficient. A known device is the CO-oximeter, a specialized spectrophotometer with four wavelengths to analyse haemoglobin concentrations. As these are analytical methods, they provide information about a determined instant, with no real time follow-up. Some *in vivo* and non-invasive techniques were based on transcutaneous measurements where the skin surface was heated to increase blood flow and then the  $O_2$  content was measured with a Clark electrode. Moreover, chemical sensors based on fluorescent dyes and optical fibres were also employed [26].

Based on spectrophotometry, other techniques arose, such as the *in vivo* optical oximeters. These make use of light to measure the concentration of different types of haemoglobin, in particular *Hb* and *HbO<sub>2</sub>*. Pulse oximeters make use of the pulsatile nature of arterial blood to measure haemoglobin concentrations. By measuring the light absorbance as blood pulsates, it is now possible to measure the oxygen saturation in arterial blood without affecting the measurements with the absorbance of venous blood [24].

### 2.2.1 Optical properties of the tissues

The Beer-Lambert's law describes the attenuation of light travelling through an absorbing medium. It states that the light intensity decreases exponentially with the distance, as equation (2.2) expresses. The equation refers only to the transmission and absorption of a single wavelength and does not account for reflections or scattering.

$$I = I_0 e^{-\varepsilon(\lambda)cd} \quad (2.2)$$

where  $I$  is the intensity for a given optical path length  $d$ ,  $I_0$  is the incident intensity,  $\varepsilon(\lambda)$  is the extinction coefficient or absorptivity of the medium for a particular wavelength and  $c$  is the concentration of the absorbing substance.

The absorbance ( $A$ ) is defined as the product of the extinction coefficient by the concentration and the optical path length, according to equation (2.3).

$$A = \ln\left(\frac{I_0}{I}\right) = \varepsilon(\lambda)cd \quad (2.3)$$

When more than one substance is present, each absorber contributes to the total absorbance, according to equation (2.4).

$$A_t = \varepsilon_1(\lambda)c_1d_1 + \varepsilon_2(\lambda)c_2d_2 + \cdots + \varepsilon_n(\lambda)c_nd_n = \sum_{i=1}^n \varepsilon_i(\lambda)c_id_i \quad (2.4)$$

where  $\varepsilon_i(\lambda)$  is the extinction coefficient of substance  $i$ ,  $c_i$  is its concentration and  $d_i$  is the optical path length through that substance.

With this, concentrations of  $n$  different substances can be measured provided that  $n$  different wavelengths are used and their extinction coefficients and optical paths length are known [24].

For the choice of the emitting wavelengths, some considerations must be taken into account. Namely, the tissue must be reasonably transparent and the absorption must be different for both of them, i.e. the extinction coefficients must be different. Besides, the production and detection of those wavelengths must be easy to accomplish [27].

The absorption spectra of biological tissue from the far ultraviolet (UV) to the middle infrared (IR) is depicted in Figure 2.2. Water is the main absorber in tissues for UV light (up to 100 nm) and also for IR light (from 1  $\mu\text{m}$ ). Since it is the major constituent of all tissues, wavelengths in this range would easily be attenuated and would not reach the photodetector. In the middle UV range (from 200 to 300 nm), some proteins and DNA also absorb a significant amount of light while in the middle IR (from 5  $\mu\text{m}$ ) that is accomplished by collagen. In the visible and near IR, water absorption is minimal and haemoglobin along with melanin are the main absorbers.

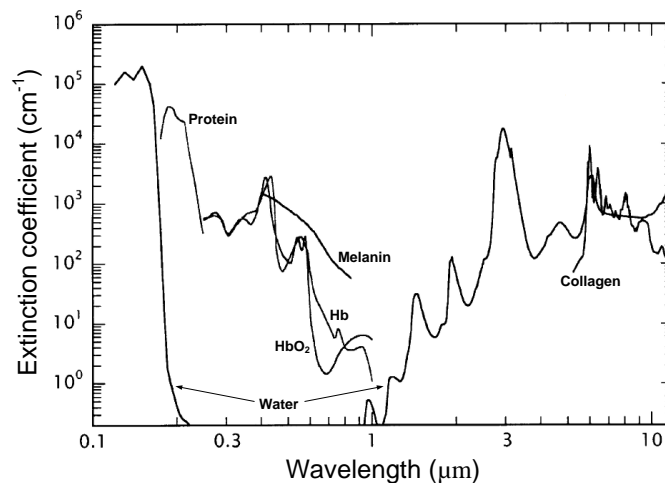
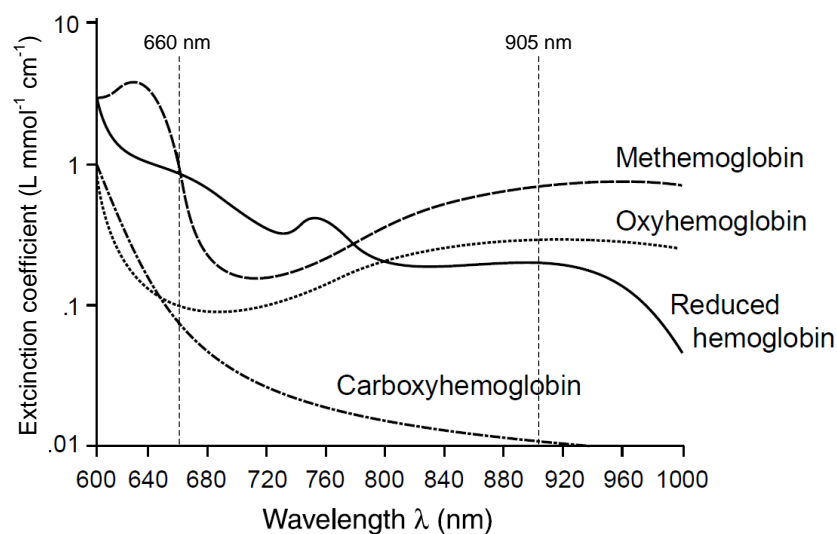


Figure 2.2 – Absorption spectra of biological tissues from 0.1 to 10  $\mu\text{m}$  [28].



Therefore, measurements must be done with visible or near infrared light. Moreover, means for both emitting and detecting these wavelengths are commercially available, with light emitting diodes (LEDs) and silicon photodiodes [27].

Another requirement is the difference between the extinction coefficients for haemoglobin species. Significant differences are found in the region from 600 nm to 1  $\mu\text{m}$  for the functional haemoglobin species. The extinction coefficients for this range and for the 4 major haemoglobin species (*Hb*, *HbO<sub>2</sub>*, *MetHb* and *COHb*) are depicted in Figure 2.3.



**Figure 2.3 – Extinction coefficients of the four most common species of haemoglobin at the wavelengths of interest on pulse oximetry. Adapted from [26].**

As it can be seen, reduced haemoglobin is more absorbed in the red light region than oxyhaemoglobin. At 805 nm, there is a point where both species have the same extinction coefficients – the isobestic point. Thereafter, the oxyhaemoglobin is more absorbed than the reduced haemoglobin. Regarding carboxyhaemoglobin, it has almost the same extinction coefficient of oxyhaemoglobin at the 660 nm, reason why this method is not good for assessing levels of carboxyhaemoglobin. The same happens with methaemoglobin that has an extinction coefficient similar to the one of reduced haemoglobin [24].

Considering all this, manufactures use a wavelength below the isobestic point (typically 660 nm) and one above (905 nm or 940 nm) [29]. The commercial probe used in this system has a 660 nm and a 905 nm wavelength and their extinction coefficients are presented in Table 2.1.

**Table 2.1 – Molar extinction coefficients for *Hb*, *HbO<sub>2</sub>*, *MetHb* and *COHb* at  $\lambda = 660$  nm and  $\lambda = 905$  nm [30].**

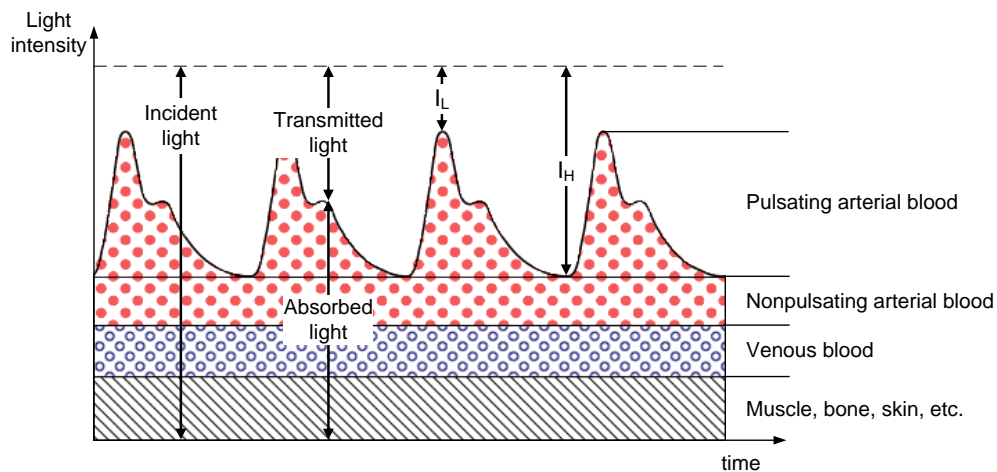
Wavelength (nm)	Extinction coefficient (L mmol <sup>-1</sup> cm <sup>-1</sup> )			
	<i>Hb</i>	<i>HbO<sub>2</sub></i>	<i>COHb</i>	<i>MetHb</i>
<b>660</b>	0.81	0.08	0.06	0.81
<b>905</b>	0.21	0.30	0.01	0.63

### 2.2.2 Photoplethysmography

Plethysmography is a technique used to measure volume changes of a given object or, in clinical practice, in an organ. Those changes are usually caused by inflows and outflows of gases or liquids when the organ shows an elastic behaviour. That is the case of the thorax during breathing, the bladder in micturition or the heart and arteries during the cardiac cycle. Mechanical, impedance and photoelectric techniques are commonly used in various physiological applications. The latter technique – photoplethysmography (PPG) – has particular interest for blood oxygenation measurements. It consists of a light source that is targeted to the tissue and a light detector that receives the transmitted or reflected light [31].

When blood pressure waves are generated by the heart during the cardiac cycle, they travel along the arteries, causing a local increase followed by a decrease of blood volume in tissues. The heart cycle is composed of two phases: diastole and systole. During the diastole, the cardiac chambers are relaxed and blood flows inside the heart, which causes a decrease in the blood vessels pressure. Afterwards, the heart contracts, ejecting the blood to the arteries, causing an increase in blood vessels pressure [23].

Those pressure changes and the consequent blood flow in the arteries cause a variation of blood volume that alters the amount of light that reaches the detector [31]. That variation is depicted in Figure 2.4. There is a strong constant light attenuation that is due to non-pulsatile arterial blood, venous blood and other tissues, such as muscle, bone, skin or fat. That is usually called the DC component. Over that, there is a variable attenuation, which is function of the cardiac cycle. More precisely, it is function of the optical path length changes. In the systolic phase the pressure is maximal and so is the amount of blood. Therefore, the transmitted light is minimal ( $I_L$ ). Conversely, at the end of the diastole, the pressure is minimal and consequently, the blood volume is minimal too. Thus, the transmitted light is maximal ( $I_H$ ).



**Figure 2.4 – Absorbed and transmitted light in pulse oximetry. A constant amount of attenuation is due to the muscle, bone, skin and other tissue plus the venous and the non-pulsatile arterial blood. The varying light attenuation is due to the arterial pulsatile blood: during the systole the light transmitted is minimal ( $I_L$ ) while at the end of the diastole it is maximal ( $I_H$ ). Adapted from [24].**

The varying component is called AC. Although it seems large in the figure, it is only 1-2% of the whole signal amplitude. PPs signal has also some low frequency oscillations due to physiological responses. Sympathetic outflow and local auto regulation impose changes in capillary density, while respiration induces changes in central venous pressure and consequently in venous blood volume. Moreover, temperature, metabolic state changes or drug effects may cause blood flow changes [32, 33].

PPG is useful to non-invasively study a set of physiological conditions. In particular, it is useful in three major fields: clinical physiological monitoring, vascular assessment and autonomic function study. Blood saturation, heart rate, blood pressure and cardiac output measurements, as well as arterial disease and arterial compliance assessment are also possible with this technique. In addition, vasomotor function, thermoregulation, blood pressure and heart rate variability can be studied [33].

### 2.2.3 Pulse oximetry

Pulse oximetry is a technique that combines the basic principles of PPG and spectrophotometry. Its name derives from the fact that pulse oximeters make use of the pulsatile nature of arterial blood. By measuring the light absorbance as blood pulsates, it is possible to determine haemoglobin concentrations and thus estimate the oxygen saturation. Pulse oximeters usually make the assumption that blood has only  $HbO_2$  and  $Hb$  and therefore, make use of only two wavelengths (red and infrared). This introduces

an error in the measurements. However, since those two totalize more than 97% of all haemoglobin species, the error is small [24].

As referred above, during the systole the received light intensity is minimal ( $I_L$ ) because the optical path length is maximal ( $d_{max}$ ). Conversely, at the end of the diastole the received light is maximal ( $I_H$ ) because the optical path length is minimal ( $d_{min}$ ). Grouping venous blood and all tissue (such as bone, muscle and skin) and representing them by  $\varepsilon_{DC}(\lambda)$ ,  $c_{DC}$  and  $d_{DC}$ , the following expressions for the diastolic (equation (2.5)) and systolic (equation (2.6)) transmitted light are obtained.

$$I_H = I_0 e^{-\varepsilon_{DC}(\lambda)c_{DC}d_{DC}} e^{-[\varepsilon_{Hb}(\lambda)c_{Hb} + \varepsilon_{HbO_2}(\lambda)c_{HbO_2}]d_{min}} \quad (2.5)$$

$$I_L = I_0 e^{-\varepsilon_{DC}(\lambda)c_{DC}d_{DC}} e^{-[\varepsilon_{Hb}(\lambda)c_{Hb} + \varepsilon_{HbO_2}(\lambda)c_{HbO_2}]d_{max}} \quad (2.6)$$

Assuming  $d_{max}$  as  $d_{min} + \Delta d$ , the systolic attenuation can be expressed as a function of the diastolic intensity ( $I_H$ ) and the change in optical path length during the cardiac cycle ( $\Delta d$ ), as equation (2.7) expresses.

$$I_L = I_H e^{-[\varepsilon_{Hb}(\lambda)c_{Hb} + \varepsilon_{HbO_2}(\lambda)c_{HbO_2}]\Delta d} \quad (2.7)$$

Light attenuation during the cardiac cycle is schematized in Figure 2.5.

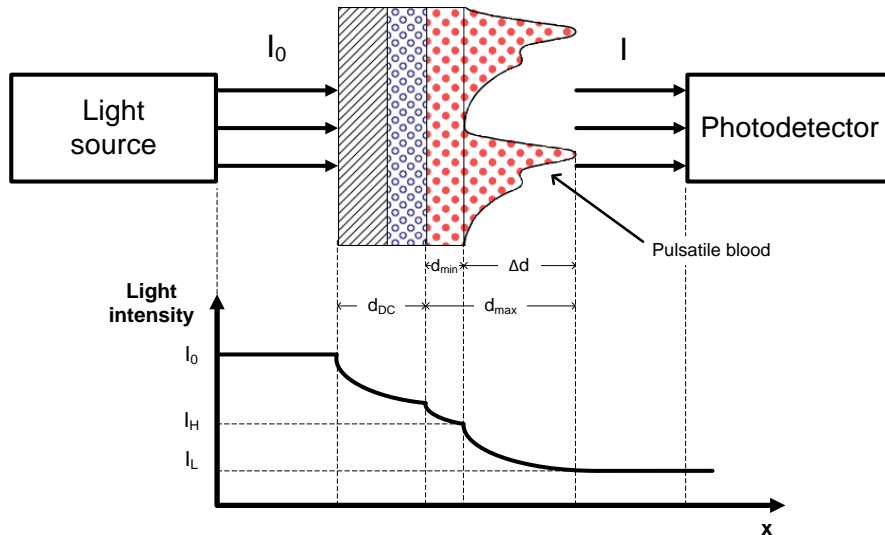


Figure 2.5 – Beer Lambert's law in pulse oximetry. The incident light is attenuated by the DC component (muscle, bone, skin and other tissue plus the venous and non-pulsatile arterial blood) over an optical path length of  $d_{DC}$ . At the end of the diastole, light intensity is  $I_H$  and the optical path length through the arteries is  $d_{min}$ ; during the systole, light intensity decreases to  $I_L$  and the optical path length through the arteries increases to  $d_{max}$ . On air, light absorption is negligible. Adapted from [24].

This manipulation is useful because ‘raw’ signals cannot be directly compared since the LEDs emit light with different intensities. Besides, the DC components have different attenuations for both wavelengths. Moreover, the optical path length differs from patient to patient and with probe positioning. Therefore, it is useful to “normalize” light intensities, making a ratio between the systolic light intensity ( $I_L$ ) and the diastolic ( $I_H$ ) one, as equation (2.8) demonstrates.

$$R = \frac{I_L}{I_H} = e^{-[\varepsilon_{Hb}(\lambda)c_{Hb} + \varepsilon_{HbO_2}(\lambda)c_{HbO_2}]\Delta d} \quad (2.8)$$

This way, the value is independent of LEDs intensity, photodetector spectral response and tissue and non-pulsatile blood attenuation.

Now, to compare the attenuation for both wavelengths, a ratio between the ratio R of red and infrared light intensity is done. That ratio is called the *Ratio of Ratios* ( $R_{OS}$ ) and is computed according to equation (2.9).

$$R_{OS} = \frac{R^R}{R^{IR}} = \frac{\ln\left(\frac{I_L^R}{I_H^R}\right)}{\ln\left(\frac{I_L^{IR}}{I_H^{IR}}\right)} \quad (2.9)$$

In practical terms, to compute  $R_{OS}$ , the maximum and minimum of light intensity reaching the photodiode must be measured. Therefore, this method for oxygen saturation measurement is called the *peak and valley* method. Other methods can be used, some of which are described in [24], although this is the simplest and the most commonly used.

Recalling equation (2.8) one can derive the theoretical meaning of the  $R_{OS}$ :

$$R_{OS} = \frac{[\varepsilon_{Hb}(\lambda_R)c_{Hb} + \varepsilon_{HbO_2}(\lambda_R)c_{HbO_2}](d_{max} - d_{min})_R}{[\varepsilon_{Hb}(\lambda_{IR})c_{Hb} + \varepsilon_{HbO_2}(\lambda_{IR})c_{HbO_2}](d_{max} - d_{min})_{IR}} \quad (2.10)$$

From equation (2.1), we can rewrite  $c_{Hb}$  and  $c_{HbO_2}$  as:

$$c_{Hb} = (1 - SO_2)(c_{Hb} + c_{HbO_2}) \quad (2.11)$$

$$c_{HbO_2} = SO_2(c_{Hb} + c_{HbO_2}) \quad (2.12)$$

Thus, assuming that the optical path length is equal for both wavelengths, and substituting equations (2.11) and (2.12) on equation (2.10), the following expression is achieved for  $R_{OS}$ :

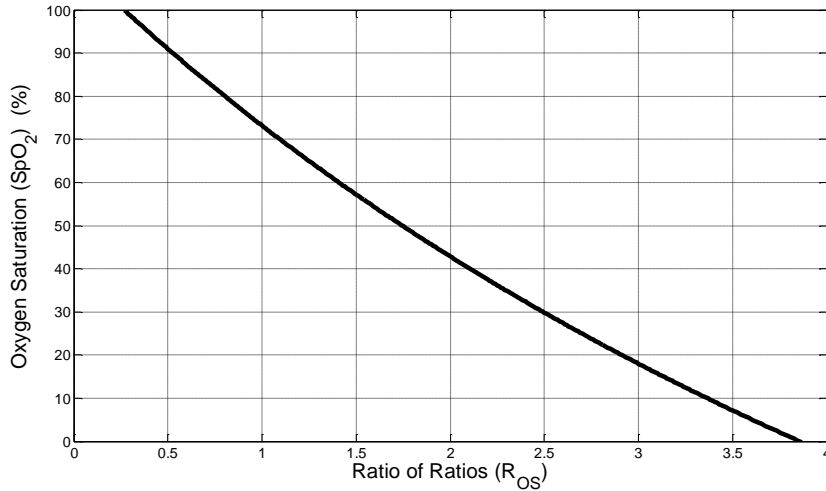
$$R_{OS} = \frac{\varepsilon_{Hb}(\lambda_R) + (\varepsilon_{HbO_2}(\lambda_R) - \varepsilon_{Hb}(\lambda_R))SO_2}{\varepsilon_{Hb}(\lambda_{IR}) + (\varepsilon_{HbO_2}(\lambda_{IR}) - \varepsilon_{Hb}(\lambda_{IR}))SO_2} \quad (2.13)$$

Rewriting equation (2.13), one can compute the indirect measure of oxygen saturation ( $SpO_2$ ) based on the calculated  $R_{OS}$ :

$$SpO_2 = \frac{\varepsilon_{Hb}(\lambda_R) - \varepsilon_{Hb}(\lambda_{IR}) \cdot R_{OS}}{\varepsilon_{Hb}(\lambda_R) - \varepsilon_{HbO_2}(\lambda_R) + [\varepsilon_{HbO_2}(\lambda_{IR}) - \varepsilon_{Hb}(\lambda_{IR})] \cdot R_{OS}} \quad (2.14)$$

Taking the values from Table 2.1, a theoretical curve can be obtained. That curve is shown in Figure 2.6.

The Beer-Lambert's law considered takes into account only the absorption and transmittance of light. Moreover, it assumes a single path between the LEDs and the photodiode. However, red and near IR light is scattered in human tissue and that leads to an ensemble of incident photon paths. The scatter is consequence of discontinuity in the index of refraction at the cell-plasma interface. Besides, the scattering effects are also dependent on red blood cell concentration. As a consequence of scattering, some photons deviate from the straight path. Therefore, some are absorbed in the tissue and some escape without being detected.



**Figure 2.6 – Theoretical curve for pulse oximetry: oxygen saturation ( $SpO_2$ ) vs. Ratio of Ratios ( $R_{OS}$ ).** Curve changes with different wavelengths: 660nm and 905 nm were used.

Due to this, the Beer-Lambert's law should make use of the *effective mean path length* for each wavelength (which is not equal for both) [34]. Some scattering models have been proposed for whole blood interaction. Though, commercial pulse oximeters compensate this by an empirical calibration in studies conducted on healthy subjects. Values of  $R_{OS}$  are computed and blood samples are taken and analysed with the CO-oximeter in order to measure the  $SpO_2$  [34, 35].

## 2.3 Height measurement

The initial purpose of the work was to develop a photoplethysmographic system for arterial blood pressure assessment (ABP). The technique is based in the oscillometric measurement method. It consists in relating the PPG waveform amplitude with the transmural pressure ( $P_{TM}$ ).  $P_{TM}$  is the difference between the pressures in the external and internal sides of the arteries, as equation (2.15) shows.

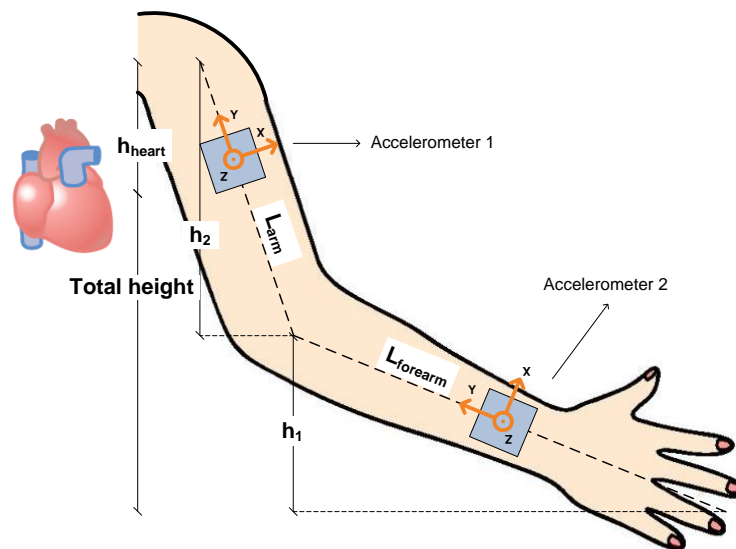
$$P_{TM} = P_{ABP} - \rho \cdot g \cdot h - P_{ext} \quad (2.15)$$

where  $P_{ABP}$  is the proximal aortic pressure,  $\rho$  is the blood density,  $g$  is the acceleration of gravity,  $h$  is the height offset between the measured site and the proximal aorta and  $P_{ext}$  is the pressure created on the outer surface of the blood vessel, as it is the case of a cuff or PPG probe.

It is known that the PPG signal amplitude varies with  $P_{TM}$ , being the greatest amplitude reached when  $P_{TM}$  is zero, i.e. when the internal and external pressures compensate each other. This is the point when the artery is most compliant and that leads to an increased signal amplitude [3]. Therefore, initially, hydrostatic pressure was an important variable to be measured. Despite the change in the course of the work, it was decided that height still was a variable of interest, since it affects PPG signal amplitude and base level.

### 2.3.1 Module architecture

The developed system has an acceleration module composed by two accelerometers: one on the arm and other on the forearm, near the wrist. Their orientation is depicted in Figure 2.7. These accelerometers are used to retrieve the height of the finger relatively to the heart. In a posterior stage, if the system is to be used to measure the ABP, this information will be indispensable, since it strongly influences ABP measurements [3].



**Figure 2.7 – Accelerometers positioning in the arm and forearm and their axes orientation. The arm and forearm lengths are indicated, as well as the increase in height due to the orientation of these segments ( $h_2$  and  $h_1$ , respectively). The distance between the heart and the shoulder is also shown. All this lead to the wanted variable: the total height between the finger and the heart.**

Moreover, the acceleration measurement may also be very useful to handle with motion in the PPG probe. The accelerometer output is a powerful tool that can be used in algorithms for dealing with motion artifacts [36-39].

To facilitate the apparatus, an approximation has been made: the orientation of the hand and the finger is assumed to be the same as of the forearm. If not, one would have to place an accelerometer on the hand and, in the limit, on each phalange of the finger in which the PPG probe is used.

However, as these are segments of small length, one may ignore that they can have different orientations and assign the length of the hand and finger to the forearm.

Assuming 19 cm as the average length of the hand (from the wrist to the end of the middle finger) [40], one can estimate a maximum error of  $\pm 0.38$  m in the height of the finger relatively to the heart. This is the limit situation in which the forearm is vertical and the hand makes an angle of  $180^\circ$  with it (e.g. if the forearm is pointing upwards and the hand and finger is pointing downwards). However, as this position is very unnatural, this error tends to be less than those 0.38 m.

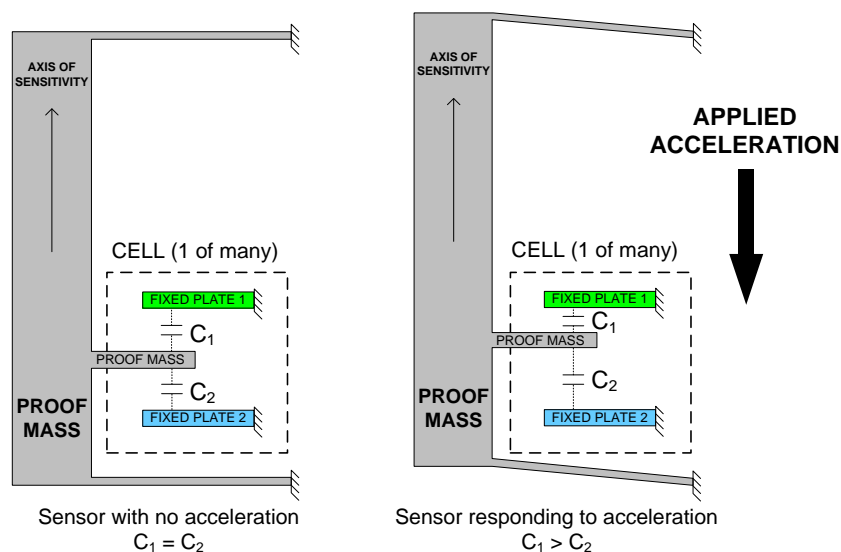


### 2.3.2 Working principles

In this work, MicroElectroMechanical System (MEMS) accelerometers were used. Their major advantage is system integration, since no more external components are needed. All the required apparatus is integrated in the same PCB and all the signal conditioning is internally done [41].

MEMS accelerometers can have different transducer types: electrostatic, with a movable mass attached to springs; electromagnetic, with a magnet attached to a flexible beam; and piezoelectric, with a mass placed over a piezoelectric cantilever [42]. The most used type is the electrostatic, composed of movable proof masses attached to springs and sensing plates. This is the case of the accelerometer used in this work [43].

The proof mass has some “fingers” that are placed between the sensing plates. Each pair of sensing plates with a proof mass in the middle can be seen as a capacitor, as shown in Figure 2.8. The principle behind these devices is that the movement of a proof mass can be measured as a change in capacitance. Whenever there is a gravitational field or a linear acceleration of the sensor package, the proof mass moves to one side. Consequently, there is a change of capacitance due to the change in distance between the capacitor plates. That change is very small, but the association of all the capacitors of the accelerometer makes the value possible to be measured by the internal circuitry [44].



**Figure 2.8 – MEMS Accelerometer: movable and fixed plates make capacitors; acceleration produces a change in distance between plates and thus changes the capacitance. Adapted from [44].**

### 2.3.3 Height computing

An accelerometer can measure the local gravitational field through the projection of the gravity field on each axis. Thus, it is possible to compute the angle the arm is tilted from the accelerometer measurement [43, 45]. Knowing the angle and measuring the arm and forearm lengths, one can determine the height of the PPG sensor relatively to the heart.

The used accelerometers have 3-axis. However, conceptually it would be possible to retrieve results with just one axis. This would be enough provided that the axis is perfectly aligned with the axis of the arm and forearm, as it is the case of the y-axis. The measured output is related with the angle  $\theta$  between the arm and the gravity (vertical direction). Thus, a perfectly vertical orientation pointing downwards gives an output of -1, a horizontal orientation results in an output of 0 and a vertical direction pointing upwards produces an output of +1.

Figure 2.9 schematizes the orientation of the y-axis. Applying basic trigonometry, equation (2.16) can be derived.

$$a_y = -\cos \theta \quad (2.16)$$

where  $a_y$  is the output of the y-axis of the accelerometer and  $\theta$  is the arm orientation relatively to the vertical (gravity field).

In a more convenient way, one can express the orientation of the arm in terms of the accelerometer output:

$$\theta = \arccos(-a_y) \quad (2.17)$$

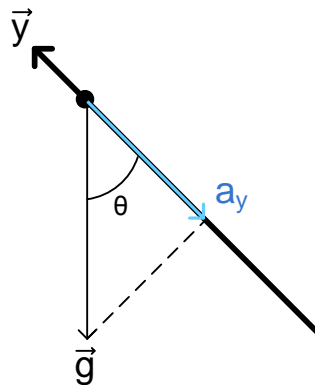


Figure 2.9 – Representation of the arm orientation ( $\theta$ ) related to the earth gravity ( $g$ ) and the accelerometer y-axis output ( $y$ ). Acceleration projection is also shown ( $a_y$ ).

However, this equation has a significant drawback. Due to the derivative of the cosine function, sensor sensitivity is not constant. In fact, the sensor has low sensitivity when the axis is close to the vertical line (either pointing upwards or downwards). When the axis is perpendicular to the gravity field, the sensitivity is maximal [46, 47].

Therefore, 2- or 3- axis accelerometers are used for better accuracy and sensitivity. In 3-axis accelerometers, the three Euler angles are used: pitch ( $\rho$ ) is defined as the angle between the x-axis and the ground; roll ( $\varphi$ ) is the angle between the y-axis and ground and theta ( $\theta$ ) is the angle between z-axis and ground. In our case, the required angle is roll, since it gives the tilt of the subject arm (y-axis of the accelerometer). That angle can be computed by equation (2.18) [48].

$$\varphi = \arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \quad (2.18)$$

where  $a_x$ ,  $a_y$  and  $a_z$  are the accelerometer output of the three axis.

This new expression allows for a constant sensitivity over a 360° rotation [43, 45]. Therefore, it will be used for the height measurements with the accelerometry module.

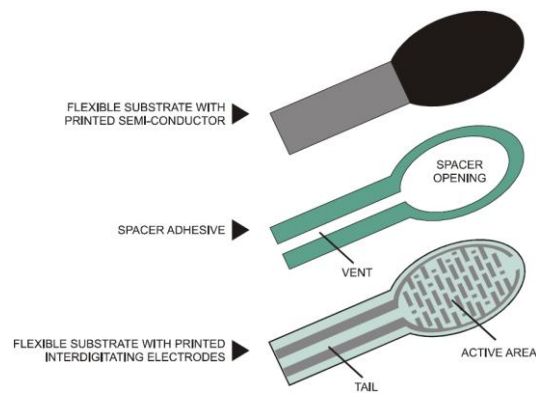
The height  $h$  is then computed using the vertical projection of the angle  $\varphi$  over a known distance  $L$  (the arm or forearm length), as equation (2.19) demonstrates.

$$h = L \cdot \sin(\varphi) \quad (2.19)$$

## 2.4 Contact force measurements

Again, the external applied force measurement is an inheritance of the initial purpose of the system: to measure ABP. Though, as this variable also influences the PPG signal (recall section 2.3) it persisted in the system, which was equipped with a way of measuring externally applied force at the finger tip. In order to retrieve that force, a Force Sensing Resistor® (FSR®) from Interlink Electronics was used [49]. This section presents its working principles.

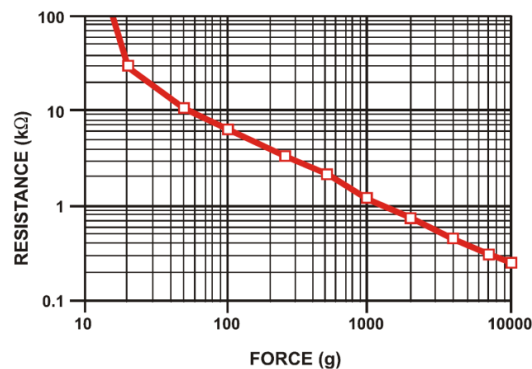
FSR® are made by a thick film of a polymer which responds to an increase in applied force with a decrease in resistance [50]. The sensor is schematised in Figure 2.10. The sensing film is made up of both conductive and non-conductive nanoparticles suspended on a matrix.



**Figure 2.10 – FSR<sup>®</sup> construction: the sensor has a layer with electrodes and a layer with a semiconductor material that touch each other when force is applied over the sensor [50].**

When a force is applied over the sensing area, the particles touch the conducting electrodes and the overall resistance is altered [51]. The advantages of this type of sensors are related with their reduced size, low cost and good mechanical resistance. However, they have relative low accuracy, with  $\pm 5\%$  to  $\pm 25\%$  of full scale error, depending on the consistency of the measurement and actuation systems as well as the use of part calibration [50]. Therefore, this sensor is used just to test the concept and a better sensor will have to be included in the future for better accuracy.

The FSR<sup>®</sup> does not have a linear response to the applied force. Its response is plotted in Figure 2.11. A turn-on threshold (which varies from sensor to sensor) is visible for low forces, with the resistance decreasing from greater than 1 M $\Omega$  to about 30 k $\Omega$ . Before that threshold, the response does not follow any known tendency. Afterwards, it approximately follows an inverse power law ( $1/F$ ). For higher forces, the response deviates from this tendency and eventually saturates. Each individual FSR<sup>®</sup> may not have the same behaviour and to reduce this error, a calibration is recommended for each individual sensor [50]. This calibration will be detailed later on (section 6.2.7).



**Figure 2.11 – FSR<sup>®</sup> resistance vs. force for the range of the sensor [50].**

---

The FSR<sup>®</sup>, as its name suggests, is a resistor that varies its value according to the force that is being exerted on it. Therefore, conditioning circuitry is needed to convert that resistance into voltage, the typical measured parameter. That conditioning will be referred in the following chapter (section 3.3).



# Chapter 3

## Hardware

*As previously referred, the developed system has three main units: the pulse oximeter sensor and its signal conditioning circuitry to retrieve pulsatile blood signal; the accelerometers to compute the finger probe height relatively to the heart; and a force sensor to measure the external force applied by the finger probe. Each of these units has some functional components related to transduction, amplification and acquisition of each variable of interest (light intensity, force and height). In addition, there is a microprocessor responsible for the tasks of controlling the other components and acquiring all the data. Besides, it has to compute some physiological parameters – oxygen saturation and heart rate. Although the computation of these parameters might seem a firmware task, part of the work is done by hardware, namely the signal extrema identification.*

## 3.1 Pulse oximeter

The probe used in this work is a commercial model (S0076B-L from BCI) composed of two light emitting diodes (LEDs) – one red (R) ( $\lambda = 660 \text{ nm}$ ) and another infrared (IR) ( $\lambda = 905 \text{ nm}$ ) – and a photodiode. The reason to choose a commercial device rather than developing one is related to the proved quality of the standard commercial sensors as well as the time saving that this choice allowed.

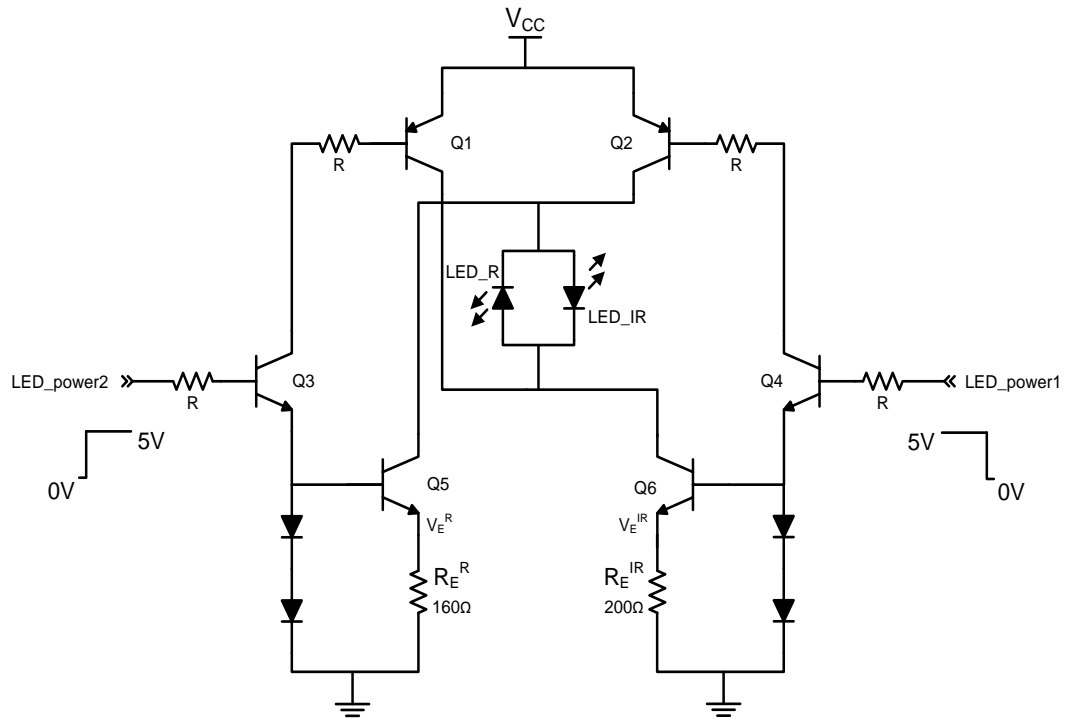
As the commercial sensor has no more hardware than the two LEDs and one photodetector, some circuitry needs to be implemented. Namely, the LEDs have to be correctly polarised and the photodiode signal must be conditioned and acquired. For that, the system has the following 5 modules: LED driver, photodetection, sample-and-hold, signal conditioning and peak and valley detector. Each of these modules will be described in the following sections.

### 3.1.1 LED driver module

Pulse oximetry employs two wavelengths at which the tissue (in this case, the finger) is illuminated. To do so, and since there is only one photodetector, two LEDs are used in opposite phase, i.e. when the red one is ON, the infrared is OFF and *vice-versa*. The LEDs brightness is controlled by their forward current. That control can be done in one of two ways. The first one consists in supplying a fixed voltage across the LED. According to the LED voltage-current curve, this produces a determined current and thus, light power. However, this has a major drawback: little variations in voltage cause major variations in the LED current, causing significant LED power changes. Therefore, another way is used to regulate the LED: a constant current source. By doing this, forward current variations do not occur and consequently, the LED intensity is constant [52]. To achieve constant current, a LED driver was developed in a previous work in the *Electronics and Instrumentation Group* (GEI) [16] and used afterwards in [17, 18]. The same circuit was employed in this work and it is shown in Figure 3.1.

*LED\_power1* and *LED\_power2* are the logic inputs generated by the microcontroller unit (MCU) and used to define which LED is ON and which is OFF. To polarize just one of the LED, these signals are opposite, i.e. if *LED\_power1* is high (5V), then *LED\_power2* is low (0V) and *vice-versa*.





**Figure 3.1 – Driver to polarize red and infrared LEDs alternately with controlled current. *LED\_power1* and *LED\_power2* define which LED is ON and which is OFF at each time. Adapted from [16].**

Although the circuit seems to be complex, it can be easily explained. Diodes D1 and D2 are used to fix a 1.4V potential at Q3 emitter. Thus, Q3 base voltage is 2.1V and when *LED\_power2* is high, a positive base current is established. Therefore, Q3 is in the linear region and draws current from the collector.

By drawing that current, a positive current appears at Q1 base and it becomes active. Consequently, current is drawn from *Vcc* to Q5 collector through the red LED.

Since Q5 base voltage is 1.4V, it is active and in linear mode and an emitter current is established. Assuming the emitter voltage as 0.7V ( $v_{BE}$  is 0.7V) and emitter resistor as  $R_E = 160\Omega$ , from equation (3.1) it follows that the emitter current is 4.4 mA.

$$i_E = \frac{v_E}{R_E} \quad (3.1)$$

As the common-emitter current gain ( $\beta$ ) is high (typically above 100), one can ignore base current and assume that collector current is also 4.4 mA. This means that through the red LED are passing 4.4 mA.

Since  $LED\_power2$  is high,  $LED\_power1$  is low (0V). That means that Q4 is cutoff. And so are Q2 and Q6, because no current flows at their bases, since those currents could only flow to Q4 collector and from Q4 emitter, respectively. Therefore, when digital pin  $LED\_power2$  is high and  $LED\_power1$  is low just one of the LEDs is driven (the red one in the case).

When half of the period of operation is reached,  $LED\_power1$  and  $LED\_power2$  switch. That induces Q1, Q3 and Q5 into cutoff mode and Q2, Q4 and Q6 into active operation. All the analysis is analogous and from equation (3.1) it can be computed that 3.5 mA flow through the infrared LED.

This driver could be simplified but since it was already developed, no alterations were made. In an upcoming version, this module can be improved.

### 3.1.2 Photodetection module

The photodetection module has two main components: the photodiode to sense the light that crosses the finger, and a transimpedance amplifier, to convert the photodiode current into voltage. The photodiode is encased in the finger probe in the opposite side of the LEDs while all the circuitry is in the printed circuit board (PCB).

#### 3.1.2.1 Photodiode

The central sensing element of the system is a silicon photodiode. Other different devices could be used, such as photocells, phototransistors or integrated circuit sensors. However, for reasons related with sensitivity, linearity, speed, size and cost, photodiodes are the most used for this applications [24].

Photodiodes can be operated in either photoconductive or photovoltaic mode (Figure 3.2). Photoconductive mode is reverse biased (i.e. the anode is negative and the cathode positive) and is used for applications where speed is required. However, this configuration introduces noise by adding a dark current. The photovoltaic mode on the other hand, is used for very low light level applications since there is no dark current (and thus the noise is reduced) [53].

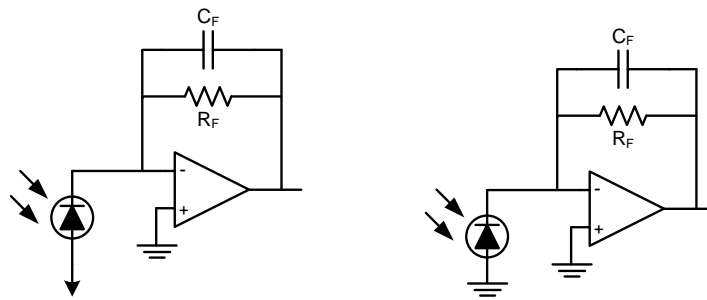


Figure 3.2 – Photodiode operation modes: photoconductive (left) and photovoltaic (right). Adapted from [53].

As it can be seen from the current-voltage characteristics of a photodiode (Figure 3.3), when  $V = 0$  and no light reaches the photodiode (curve ①), no current is produced. When light reaches the photodiode, a photocurrent  $I_s$  directly proportional to incident light is produced [54]. Conversely, when a voltage exists between the photodiode terminals, even with no incident light, a current exists. The current magnitude increases with increasing light, but in a non-linear manner. Therefore, as this application requires no fast measurements but high sensitivity, the photovoltaic mode was used. Moreover, this configuration presents much less variations with temperature [53].

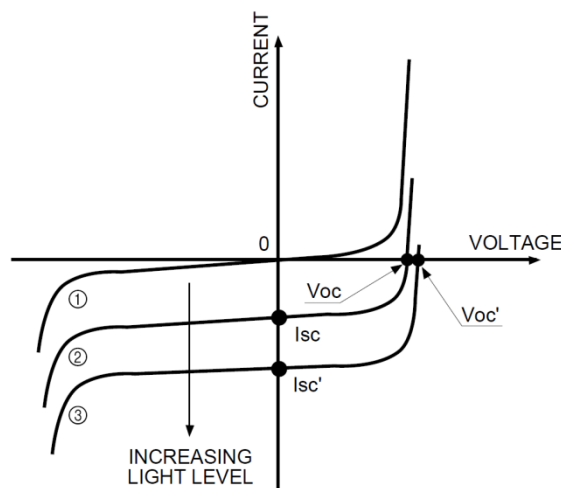


Figure 3.3 – Current-voltage characteristics for a typical photodiode: curve ① shows the response for no incident light; light intensity is increasing from curve ② to ③. Adapted from [54].

Some practical considerations must be taken into account when choosing the photodiode. Namely, the photodiode capacitance should be as low as possible, since it affects bandwidth and noise. Moreover, photodiode active area must be small because photodiode capacitance is proportional to it. Besides, diode shunt resistance must be high to prevent noise, voltage offset and drift that are amplified [55]. As a commercial probe

was used in this work, it is already adapted to this purpose. However, it is important to remember these features when choosing these components, since they will lead to an increase in signal-to-noise ratio (SNR).

Another important characteristic of photodiodes is its spectral sensitivity. However, due to the signal normalisation (seen in section 2.2.3), this is not so important in this context, as the sensitivity will cancel in the peak and valley ratio.

### 3.1.2.2 Transimpedance amplifier

As referred before, when light reaches the photodiode, a photocurrent is produced. However, as typical analogue-to-digital converters (ADC) sample voltages and not currents, a transimpedance amplifier must be used to convert the current into voltage. As seen above, the photovoltaic mode requires zero diode voltage, which is accomplished with the virtual ground [56]. The basic configuration of the transimpedance amplifier is shown in Figure 3.4.

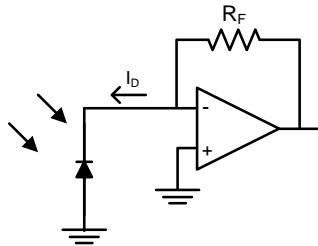


Figure 3.4 – Basic transimpedance amplifier configuration to measure photodiode output current. Adapted from [56].

The feedback resistance ( $R_F$ ) determines bandwidth as well as gain. Since there is ideally no current on the op-amp inputs, current flows through the feedback resistor creating an output voltage given by equation (3.2).

$$v_o = I_D R_F \quad (3.2)$$

Concerning the circuit noise, and since there is no dark current, the main source is the thermal (Johnson–Nyquist) noise which has the following root mean square value:

$$v_n = \sqrt{4 k_B T BW R_F} \quad (3.3)$$

where  $k_B$  is the Boltzmann's constant,  $T$  is the absolute temperature,  $BW$  is the bandwidth and  $R_F$  is the feedback resistance.

Analysing equations (3.2) and (3.3), it can be seen that signal increases with  $R_F$  while noise increases with  $\sqrt{R_F}$ . This means that SNR increases with the square root of feedback resistance. Thus, it should be as high as possible [55].

However, high resistances lead to gain peaking effects and instability so, a feedback capacitor ( $C_F$ ) has to be used [56]. It can be proved [57] that the appropriated feedback capacitance ( $C_F$ ) to ensure stability is given by equation (3.4).

$$C_F = \frac{1}{4\pi R_F f_c} (1 + \sqrt{1 + 8\pi R_F C_I f_c}) \quad (3.4)$$

where  $R_F$  is the feedback resistance,  $f_c$  is the unity gain frequency of the op-amp and  $C_I$  is the total input impedance (photodiode junction and op-amp input capacitances).

The signal bandwidth is calculated by:

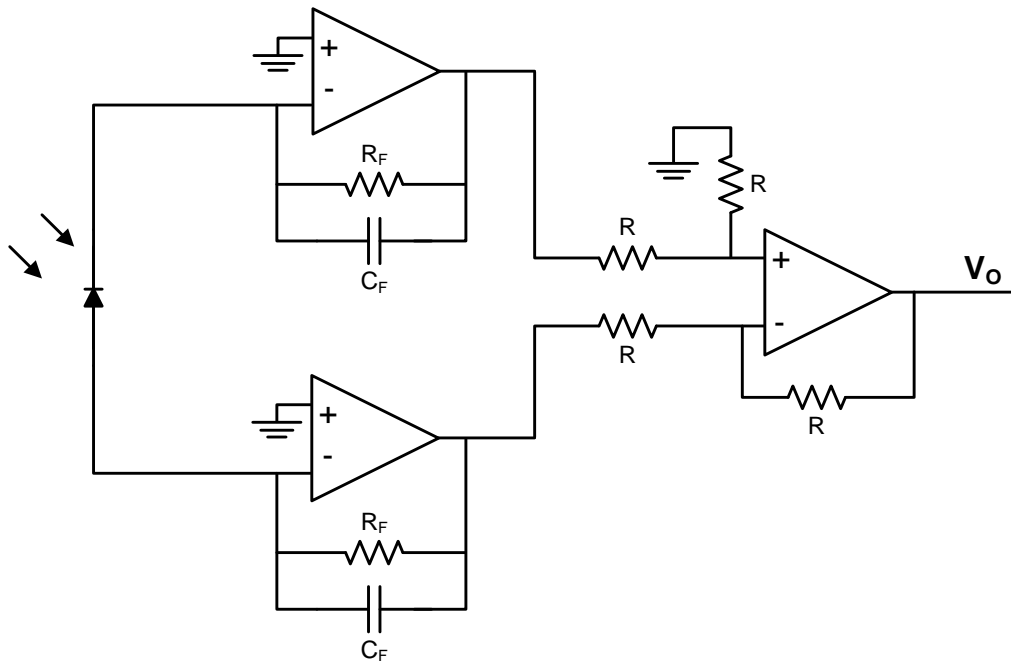
$$BW = 1.4 \sqrt{\frac{f_c}{2\pi R_F (C_I + C_F)}} \quad (3.5)$$

As the pulse oximeter probe is a commercial model and no technical information was given, feedback capacitance was not computed by these formulas. Instead, by experimental tests, a 30 pF capacitor was found to be a good choice.

From equation (3.5) it is obvious the requirement of small photodiode capacitance that was presented in the previous topic, since capacitance affects the bandwidth. Moreover, it also affects stability.

Due to the high feedback resistance, the transimpedance amplifier is very susceptible to noise coupling. Some techniques to reject noise are based on differential measurements. In particular, a very common circuit for pulse oximeter photodiode measurements integrates two transimpedance amplifiers whose outputs are fed into a differential amplifier [24, 56]. The circuit is shown in Figure 3.5.

With this circuitry, photodiode current flows through two equal resistances. Therefore, both will receive equal electrostatic noise coupling. Therefore, there will be a differential output on the resistors, but the noise coupling is common-mode signal and will be cancelled.



**Figure 3.5 – Differential photodiode current sensing: transimpedance amplifier for pulse oximetry. Adapted from [24].**

And as required, there is no voltage across the diode as both terminals are virtually connected to ground [24, 56]. The output voltage is given by equation (3.6).

$$v_o = 2I_D \frac{R_F}{1 + j\omega R_F C_F} \quad (3.6)$$

This configuration also has the advantage of needing half-value feedback resistors to have the same amplification which reduce thermal noise and increase bandwidth [24].

Op-amp input bias currents are a major limitation of the transimpedance module since they cause voltage offset errors with large feedback resistors [55]. Therefore, FET op-amps are generally used since they have less bias currents. In this system, the OPA129 [58] was used. The reason to use this op-amp is that it was previously used in a work at this group [59] and proved to be a good choice for photodiode signal measurements, even after some other op-amps had been tested. It has dielectrically-isolated FET (Difet<sup>®</sup>) inputs to greatly reduce the bias current (typically 30 fA). It also has low voltage (15 nV/ $\sqrt{\text{Hz}}$ ) and current noise (0.1 fA/ $\sqrt{\text{Hz}}$ ) and a low offset ( $\pm 0.5$  mV) [58]. All of these features make it good for amplifying photodiode signals, which have very low currents (at most a couple of hundred nA).

### 3.1.3 Sample-and-Hold

The system operates on a basis of a pulsed LED signal, i.e. red and infrared LEDs turn ON alternately. And since there is only one photodetector, our signal is modulated. Voltage at the transimpedance amplifier output is proportional to the light acquired by the photodetector when the finger is illuminated by the red LED during a time period of  $T/2$  and by the infrared LED during the following  $T/2$ . However, as LEDs have different optical power and the extinction coefficients of the involved tissues are different for both, it is desirable that the two channels are separated. This will allow for different amplification gains, as well as different DC level shifts.

To do so, a sample-and-hold circuit (S/H) is used. A S/H is designed to sample the voltage of an analogue signal and hold that voltage for a certain period of time. The basic components are an input voltage buffer, an interrupt, a capacitor and an output buffer, as Figure 3.6 shows [60].

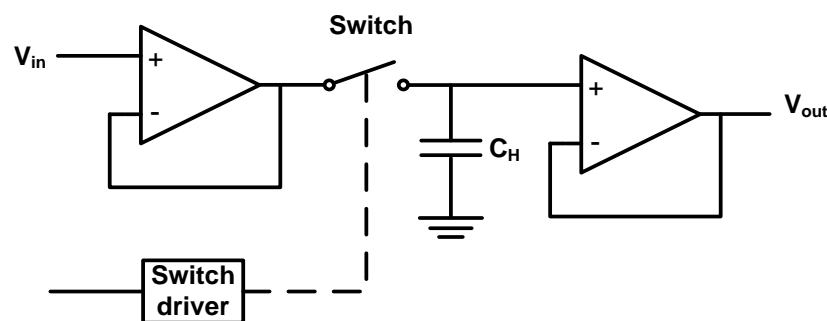


Figure 3.6 – Sample-and-hold basic circuit: two buffers and a capacitor. Adapted from [60].

The choice of the capacitor value is a trade-off. If the value is too small, the leakage current causes the voltage to drop significantly and the system is not accurate. If the capacitor is too large, it may not charge during the time the switch is on [61]. Nowadays, many integrated circuit (IC) S/H are available in the market. In this work, the LF398 [62] was used. The choice of the capacitor value was based on its performance characteristics, namely the acquisition time, hold step and output droop rate, presented in Figure 3.7.

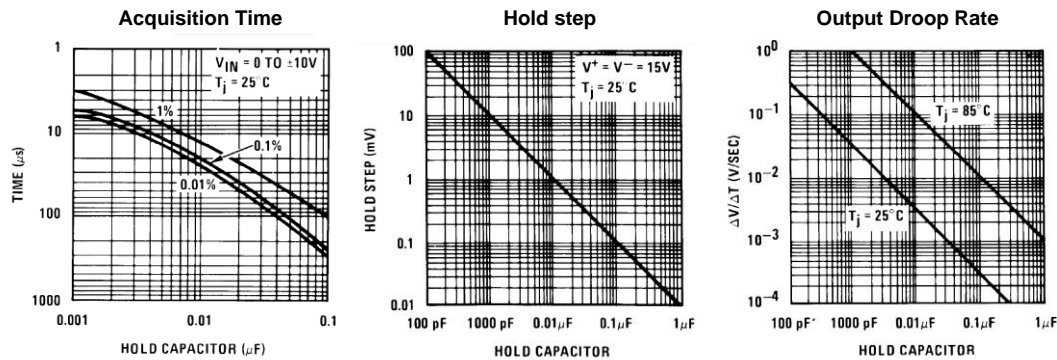


Figure 3.7 – LF398 typical performance characteristics: acquisition time, hold step and output droop rate [62].

Regarding hold step and output drop, for better performance, a big capacitance must be used. As the pulsed signal of the system is expected to run at 500 Hz, each LED will be ON for 1 ms. This way, a capacitor of 100 nF will be a good choice, as it requires less than 300  $\mu\text{s}$  to settle to 0.01%. This will lead to an output droop rate of  $\sim 0.3$  mV/s and a hold step of  $\sim 0.1$  mV. The LF398 has both FET buffer amplifiers with high impedance ( $\sim 10$  G $\Omega$ ) and small input currents ( $\sim 10$  nA), as well as a small gain error ( $\sim 0.004\%$ ) and leakage current during the “hold” phase ( $\sim 30$  pA) [62].

### 3.1.4 Signal conditioning

Now that each signal is separated into different channels, all the signal condition as filtering, amplification and offset removal can be done separately in order to adjust each signal to the dynamic range of the ADC.

As the signal from the transimpedance amplifier was modulated with a square wave of 500 Hz, at that time the signal could not be low-pass filtered. Now that the channels are separated by the S/H, it is possible to filter the signal to remove unnecessary components, typically noise. Common cutoff frequencies present in literature range from 8 to 30 Hz [63-68]. After analysing the PPG signal and testing some different frequencies, 20 Hz was chosen as a good value for cutoff frequency. For that, a second order Butterworth filter was implemented. This is the most common filter design since it provides the maximum pass-band flatness and a good roll-off in frequency response (although not as good as Chebyshev filters) and little overshoot and ringing in step response [69].

For low quality factors ( $Q = \frac{1}{\sqrt{2}}$  for Butterworth filter) and low gains (unitary gain was desired), the Sallen-Key topology is the most used for second-order filters [70].



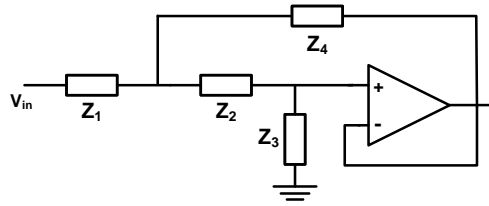


Figure 3.8 – Butterworth filter: generic Sallen-Key topology [71].

The general configuration is shown in Figure 3.8 and it is up to the designer to choose the values of resistors and capacitors to achieve the desired frequency response. Deriving the transfer function of the Sallen-Key, equation (3.7) is obtained.

$$\frac{V_{out}}{V_{in}} = \frac{Z_3 Z_4}{Z_1 Z_2 + (Z_1 + Z_2) Z_3 + Z_3 Z_4} \quad (3.7)$$

For a low-pass filter,  $Z_1$  and  $Z_2$  are resistors (designated  $R_1$  and  $R_2$ , respectively) and  $Z_3$  and  $Z_4$  are capacitors (designated  $C_2$  and  $C_1$ , respectively). Therefore, the low-pass filter transfer function is given by equation (3.8).

$$\frac{V_{out}}{V_{in}} = \frac{1}{s^2 + \frac{R_1 + R_2}{R_1 R_2 C_1} s + \frac{1}{R_1 R_2 C_1 C_2}} \quad (3.8)$$

Making some simplifications ( $R_1 = R_2 = R$  and  $C_2 = \frac{1}{2} C_1 = C$ ), equation (3.9) can be used to set resistor and capacitor values.

$$f_0 = \frac{1}{2\sqrt{2}\pi RC} \quad (3.9)$$

This way, the capacitance  $C$  is chosen and then  $R$  is calculated in order to obtain the desired cut frequency ( $f_0$ ). A more detailed mathematical analysis is present in the literature [70, 71].

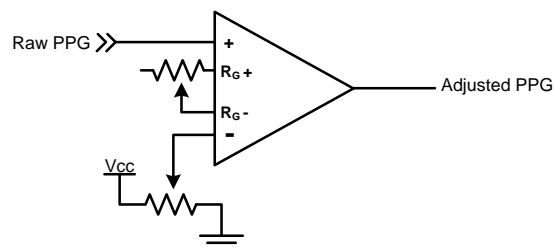
In order to obtain a cut frequency of 20 Hz, the following values were chosen:  $R_1 = 450 \text{ k}\Omega$ ,  $R_2 = 450 \text{ k}\Omega$ ,  $C_1 = 25 \text{ nF}$  and  $C_2 = 12.5 \text{ nF}$ .

From here, it was decided to split the signal in two ways. First, a signal with both DC level and AC variations must be obtained to compute the  $SpO_2$ . This will be referred as the DC+AC component from now on. Then, an AC signal with just the pulsatile component is useful for display, as this is much more stable than the DC+AC signal. Both paths will be explained in more detail in the following two topics.

### 3.1.4.1 DC conditioning

To obtain the  $SpO_2$ , some calculations with the PPG signal maxima and minima from both red and infrared LEDs are required. For that reason, it is fundamental that these signals match the dynamic range of the ADC (0 to 5V) in order to get the best SNR, reducing the quantization error. As the PPG signal is composed by a small AC signal imposed on a large DC baseline, it is not desirable that the dynamic range is almost completely occupied by the DC level. This way, an instrumentation amplifier was used to remove the DC level. Also, that small AC component must be amplified, which can be accomplished with the same instrumentation amplifier, as they have two pins for defining the gain.

The circuit designed to accomplish the DC conditioning is shown in Figure 3.9. To remove the DC level, the original signal is fed to the non-inverting input and a DC voltage is fed into the inverting input. A DC voltage is set by a digital potentiometer (digipot), which allows the system to self-adapt to changes in this DC value that are common in natural physiologic conditions [32]. Another digipot is used to set a resistance between the gain pins, thus allowing for gain control.



**Figure 3.9 – DC conditioning circuitry: a digital potentiometer is used to set gain and level shift of each channel.**

A digipot has two configurations: potentiometer and rheostat. For level shift, a DC potential must be generated, which can be done with the potentiometer mode. It is basically a voltage divider, where one terminal is connected to 5V and the other to the ground (or other wanted potential). This way, when the signal is either rising or falling to values outside the ADC range, the MCU can change the DC level by shifting the digipot wiper position.

Concerning, the rheostat mode is used. It is a variable resistor whose resistance is set by the wiper terminal position. Therefore, gain can be set by changing the wiper value, which, again, is done by the MCU.

The chosen amplifier was the AD623 [72] from Analog Devices, a very used instrumentation amplifier in this type of applications. For its choice greatly contributed the possibility of unitary gain which is not presented in all instrumentation amplifiers. Its gain is defined according to the gain resistor ( $R_G$ ) as equation (3.10) expresses.

$$G = 1 + \frac{100 \text{ k}}{R_G} \quad (3.10)$$

Programmable gain amplifiers were discarded due to their very discrete gain selection. In this work, a more dynamic and continuous gain was desired to better control the system. Furthermore, as the dynamic range of the ADC was from 0V to 5V and all the instrumentation was to be power supplied by 5V from the Arduino™, a rail-to-rail amplifier was valued.

The digipot used for both DC level generation and gain setting was the AD5252 [73]. This is a dual 256-position non-volatile digital 100 k $\Omega$  potentiometer that communicates by I<sup>2</sup>C. Its settling time is 14  $\mu$ s from the value of 0 to 255 [73]. In the rheostat configuration, the output resistance is given by equation (3.11). The 75  $\Omega$  are due to the wiper contact resistance, as it is a non-ideal switch. D is the wiper position, which can vary between 0 and 255.

$$R_{WB} = \frac{D}{256} \times R_{AB} + 75 \quad (3.11)$$

In the potentiometer configuration, if the wiper resistance is ignored, the output voltage on the wiper is given by equation (3.12) [73].

$$V_W = \frac{D}{256} \times V_{AB} + V_B \quad (3.12)$$

The 256 steps allow for a 390  $\Omega$  step in rheostat mode and 19.5 mV steps in potentiometer mode. As the digipot has a total resistance of 100 k $\Omega$ , according to equation (3.10), it is not possible to get a gain lower than 2 on the AD623. With no resistor, the gain would be 1 and with 100 k $\Omega$  rheostat the gain will be 2 (1+100k $\Omega$ /100k $\Omega$ ). A gain higher than 2 is accomplished reducing the rheostat value. The AD623 gain is not linear and thus, there is a much better gain resolution for gains near 2 while resolution gets worst for higher gains. This can be improved in the future with a non linear digipot. A higher resistance range can also be considered, but for the first prototype, this seemed enough.

### 3.1.4.2 AC conditioning

AC filtering produces a good signal for waveform since it intrinsically removes the DC level which constantly changes even under normal physiological conditions [32]. This makes the signal much more stable and easier to present in a waveform monitor. Though, it cannot be used to make calculations, since  $SpO_2$  computations need the absolute voltages for both signals (red and infrared).

To filter the signal, a 2<sup>nd</sup> order Butterworth high-pass filter was used because roll-off was important to reject DC value, but not small AC components. This way, a first-order filter was not suitable. However, higher orders were not used because of the increasing delay they add and that could invalidate the subsequent extrema identification (section 3.1.5). The choice of the Butterworth was the same as before: flatness of passband.

It is worth of notice that a low-pass filter was also tested. The low-pass filtered signal was then subtracted to the original signal. However, in this case, a delayed filtered signal was being subtracted to a real time signal, which could introduce distortions in the PPG signal. Therefore, it was concluded that the high-pass filter would be a better choice.

The high-pass Butterworth Sallen-Key topology is the same as Figure 3.8 but now,  $Z_1$  and  $Z_2$  are capacitors (designated  $C_1$  and  $C_2$ , respectively) and  $Z_3$  and  $Z_4$  are resistors ( $R_2$  and  $R_1$ , respectively). From equation (3.7), and again making some simplifications ( $C_1 = C_2 = C$  and  $R_1 = \frac{1}{2}R_2 = R$ ), the components value can be set according to equation (3.13), which is the same as the low-pass filter.

$$f_0 = \frac{1}{2\sqrt{2}\pi RC} \quad (3.13)$$

After some experimental tests conducted to choose the best cutoff frequency (Appendix A), it was decided that the filter that better conciliated DC removal with little signal distortion was with  $f_0 = 0.2 \text{ Hz}$ .

It was found that for lower frequencies, the output signal still had much DC level variations, indicating that the low frequency oscillations were not fully removed. Conversely, for higher frequencies, the AC component significantly differed in shape from the original signal. Therefore, 0.2 Hz was found to be a good trade-off, not presenting significant oscillations and reasonably preserving the PPG waveform. Thus, filter components were chosen as following:  $R_1 = 170 \text{ k}\Omega$ ,  $R_2 = 340 \text{ k}\Omega$ ,  $C_1 = 3.3 \text{ }\mu\text{F}$  and  $C_2 = 3.3 \text{ }\mu\text{F}$ .

After the signal has been AC filtered, its DC value is ideally 0V, i.e. the signal varies around 0V having both positive and negative voltage segments. As the dynamic range of the ADC is 0V to 5V, it is necessary to shift the filtered signal. To do so, an instrumentation amplifier is used. It has a pin to define the zero output voltage, thus providing a direct means of injecting a precise offset to the output [72]. The same AD623 [72] used before for DC conditioning was employed here. This way, providing 2.5V into the reference pin, the filtered signal will swing between 0V and 5V, provided the right gain is set. This gain is defined, as above, via a digital potentiometer (AD5252 [73]) controlled by the MCU (section 4.5).

### 3.1.5 Peak and valley detector

Maxima and minima voltages of each channel are required to compute the  $SpO_2$ . However, real time identification of extrema is extremely heavy to do recursively in the MCU. It would be much more efficient – and less processor time consuming – to know the temporal location of each extrema. This way, no real time search algorithms would be needed.

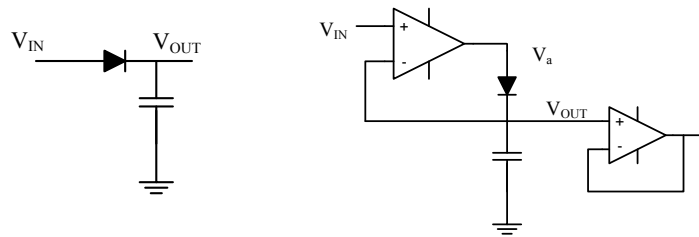
To solve this issue, a circuit was implemented in the system to “tell” the MCU that a maximum or minimum has occurred and to present that voltage for an ADC conversion. For that, two stages are needed: first, the maximum and minimum voltage has to be held in order to the MCU to digitalize it; then, a signal is sent to the MCU to “tell” that the peak or valley voltage is ready for ADC conversion and a reset is done so that a new peak voltage can be stored. This module is based in a circuit previously developed at GEI [13].

#### 3.1.5.1 *Peak and valley voltage holding*

In order to store the peak and valley voltages, a peak and valley detector was implemented. The basic peak detector consists of a diode and a capacitor (Figure 3.10). When the signal increases, the diode conducts and the capacitor is charged. When the input starts to decrease the diode becomes reverse biased and the capacitor remains at the peak voltage until a higher peak is input.

Although being a simple configuration, it yields an important problem: the output is one diode drop below the input. By placing the diode in the negative-feedback loop of an op-amp, this effect can be avoided.

Apparently, the capacitor has no route to ground to discharge so, it should hold the peak voltage forever. However, both the capacitor leakage current and the load of the circuit may discharge it, causing inaccurate voltage readings when the MCU samples the signal. The introduction of a buffer after the first stage reduces this discharge, especially if high input impedance and low input bias current op-amps are used [61, 74, 75].



**Figure 3.10 – Peak detector: Basic configuration (left) and two stage active peak detector (right) Adapted from [74].**

Following the op-amp output voltage equation (3.14), as the input signal  $v_{in}$  (which is equal to  $v^+$ ) raises above  $v_{out}$  (which is equal to  $v^-$ ), the op-amp sees a positive voltage difference between the input terminals and the voltage  $v_a$  increases. As soon as the voltage across the diode is greater than  $v_D$ , it starts to conduct and a feedback path between the op-amp negative input and output terminals is established. Therefore, a virtual short cut appears between  $v^+$  and  $v^-$ , which guarantees that  $v_{out}$  is equal to  $v_{in}$  – the circuit acts as a follower. However, unlike the basic configuration where the input signal had to rise the whole  $v_D$ , with this configuration the input signal has to rise just  $\frac{v_D}{A}$ , as can be seen from equation (3.15). Since the open-loop gain  $A$  is very high, the output voltage  $v_{out}$  follows  $v_{in}$  almost immediately and with virtually no drop [61, 74].

$$v_a = A(v^+ - v^-) \quad (3.14)$$

$$A(v^+ - v^-) > v_D \quad (3.15)$$

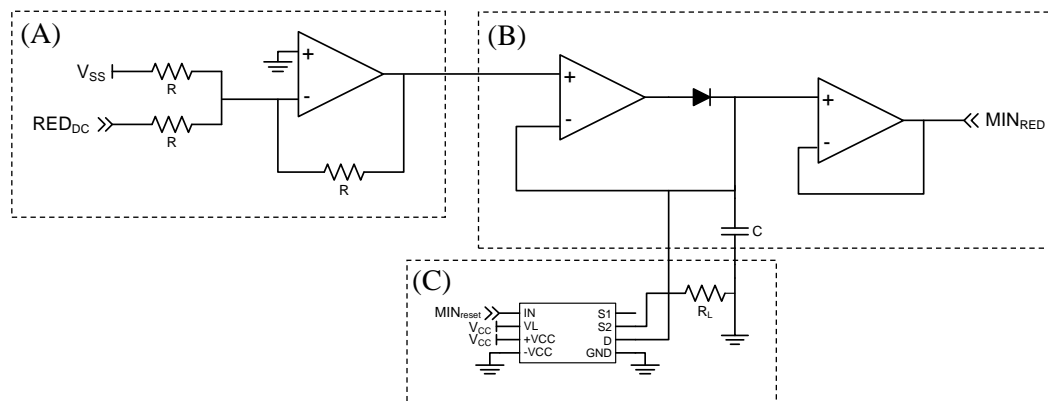
When the input signal  $v_{in}$  decreases, a negative voltage difference between the input terminals ( $v^+ - v^-$ ) is seen and the voltage  $v_a$  decreases, which causes the diode to stop conducting. Therefore, there is neither charge nor discharge of the capacitor and  $v_{out}$  remains the same [61, 74].

The circuitry described above is used to detect signal maxima, i.e. the most positive point of each pulse – physiologically called the systolic peak. However, minima of the signal – the end of the diastolic period – are also required.

Therefore, a similar circuitry was designed for minima detection. It consists of a summing amplifier that subtracts 5V to the signal and then inverts it for positive voltage levels. This transforms minima of the original signal in maxima and the same peak detector configuration described in Figure 3.10 can be used to hold that value. It should be noticed that an instrumentation amplifier like the AD623 [72] used in previous hardware could be employed here. To do that, the non-inverting input would be tied to ground and the signal input to the inverting one; then, the reference pin would be used for level shift.

As said above, the capacitor has no direct route for discharging. Though, it is desired that each peak voltage is sampled so, the capacitor must be discharged after each peak. To do so, a resistor of small value could be placed in parallel with the capacitor. This would give a way for the capacitor to discharge quickly. However, it is desirable that the charge is held until the ADC conversion is done and, just after that, the capacitor can be discharged. Therefore, an analogue switch is used to control when the resistor is connected to the capacitor terminal.

The complete peak and valley circuitry is depicted in Figure 3.11. The first stage is used to invert the signal and is used just for valley detection. The rest of the circuitry is common to both maxima and minima detection.



**Figure 3.11 – Peak and valley detector module. Peak detector circuitry (B), analogue switch used for capacitor discharge (C) and inverting op-amp for signal inversion required for valley detection (A).**

As referred before, bias current discharges the capacitor and causes the voltage to drop. Equation (3.16) gives the voltage drop due to bias input current of the buffer. Also, diode reverse current may discharge the capacitor. To overcome this drawback, more advanced circuits are described in literature [61, 75] but they were not found necessary for this system, at least in this first prototype.

$$\frac{dV}{dt} = \frac{I_{bias}}{C} \quad (3.16)$$

Also, when  $v_{in}$  is lower than  $v_{out}$ , the diode does not conduct and the op-amp feedback loop is broken. This cause the op-amp output to saturate (due to the open-loop gain) and thus, when a new peak appears, the op-amp has to go all the way from negative saturation to the new peak voltage, which takes time. From equation (3.17), this delay can be estimated. To avoid this fact, alternative circuits could be employed but again, this was not found as an important shortcoming [61, 74, 75].

$$\frac{dV}{dt} = \frac{I_{out}}{C} \quad (3.17)$$

Therefore, the choice of the capacitor value greatly depends on these two factors: discharge by the input bias current and system response time. Besides, the period for which it is required that the system holds the voltage also influences the choice – the earlier the sample signal is given, the smaller the capacitor can be. After some experimentation, the ideal value was found to be 5  $\mu\text{F}$ .

For this capacitor, and knowing that for the chosen op-amp – the TL082 — the bias current is typically 20pA [76], the voltage drop due to this current is expected to be 0.008 mV/s. Experimentally, the sample signal delay was found to be around 680 ms which would give an output drop of about 0.005 mV. However, the analogue switch leakage current (0.1 nA [77]), but mainly the diode reverse current (25 nA [78]) significantly increase these effect to 5 mV/s. For the signal delay referred, it gives a drop 3.4 mV. This seems a reasonable value, at least for the first prototypes.

The TL082 output current (40 mA [76]) defines the global slew-rate for the peak detector, which is 8 V/ms, i.e. much lower than the TL082 slew-rate – 13V/ $\mu\text{s}$  [76]. However, since the PPG signal is relatively slow and the time between maxima and sampling signal is of the order of hundreds of milliseconds, this value is adequate for the purpose.

### 3.1.5.2 *Sample and Reset signal*

In the preceding topic, the concept of peak detector reset was introduced. It consists in creating a temporary route for capacitor discharge. That is accomplished by the closing of the analogue switch, creating a path to ground trough the resistor. The reset signal timing, i.e. the signal used to close the analogue switch, must be given only after the peak has been held and sampled. For that, a circuitry was designed to detect when maxima and minimum of the signal occurs.



The essence of that circuitry is to differentiate the PPG signal to identify its extrema – the points where the signal reverses the direction, meaning that its derivative is zero. However, as the signal has noise, a multiple number of zero crossings would be expected. Besides, as the signal usually has two peaks – the systolic and the diastolic – another unwanted zero would appear. To solve that, an initial filtering step is performed.

The PPG waveform has a strong frequency component around the heart rate, i.e. between 1 and 1.6 Hz (60 to 100 BPM). Therefore, a low-pass filter around this frequency would give an approximated sinusoidal wave with this fundamental frequency. This means that information about the diastolic wave is lost and a single sinusoidal period for each heart beat is kept. Since an AC filtered signal is available, this is used as the filter input. The reason to choose these AC signal rather than the DC+AC signal is related to the greater stability of the AC signal since it does not have the DC level changes present in the original signal, as previously referred. This promotes the stability of the sample and reset signals, preventing false signalling.

Again, the filtering was accomplished with a 2<sup>nd</sup> order Butterworth low-pass filter to have a good roll-off but not considerable delay. In fact, a good roll-off is appreciated in order to actually remove the signal components other than the fundamental heart beat. This way, it can be assured that no diastolic peak is left, which would give us unwanted double sample and reset signals. No higher orders were used since signal delay increases with increasing order. A considerable delay would be unusable because the reset signal could be generated so late that the system would not be ready for detecting the new peak, resulting in erroneous peak voltage measurements. In fact, this is even more crucial if one considers that the AC signal used as the input for all this signal generation module has already some delay – due to the 2<sup>nd</sup> order AC filter employed before. Moreover, the low-pass characteristic of the filter itself delays the signal, since the higher frequencies are cut, thus making the output signal to increase and decrease slower than the input signal.

Some different cutoff frequencies were tested to ensure the performance of the system. A few simulations done in OrCAD from Cadence<sup>®</sup> are shown in Appendix B. It was concluded that a cutoff frequency of 0.5Hz produced a reset signal for peak detector later than the subsequent systolic peak for every heart rate. Thus, every two peaks, one would be missed. For a cutoff frequency of 1.0 Hz, this still happened for a signal with a heart rate of 100 bpm and the same almost happened for 1.5 Hz. At the opposite extreme, a cutoff of 3.0 Hz produces multiple reset signals for all heart rates, meaning that the diastolic peak was not fully filtered. With a cutoff of 2.5 Hz, the 60 bpm differentiated

trigger almost crossed the zero multiple times, having a plateau very close to 0V. Therefore, as a matter of safety, and since signals with heart rates close to 60 bpm are more likely to happen than with 100 bpm, a cutoff frequency of 1.5 Hz was chosen. Using equation (3.9), the values for the filter components were computed as:  $R_1 = 750 \text{ k}\Omega$ ,  $R_2 = 750 \text{ k}\Omega$ ,  $C_1 = 200 \text{ nF}$  and  $C_2 = 100 \text{ nF}$ .

If this signal was now differentiated, it could be used to detect extrema of the sinusoidal signal since these are the points where the derivative crosses 0V. A differentiator circuit can be designed placing a capacitor in the feed-in path of an op-amp, as Figure 3.12 shows.

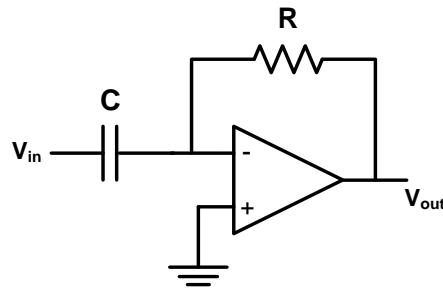


Figure 3.12 – Differentiator circuit [74].

When the input is a time varying function  $v_I(t)$ , and since the inverting terminal of op-amp is a virtual ground, the current that flows through a capacitance  $C$  is given by equation (3.18).

$$i(t) = C \frac{dv_I(t)}{dt} \quad (3.18)$$

This current will flow through the feedback resistor  $R$  producing an op-amp output voltage given by equation (3.19).

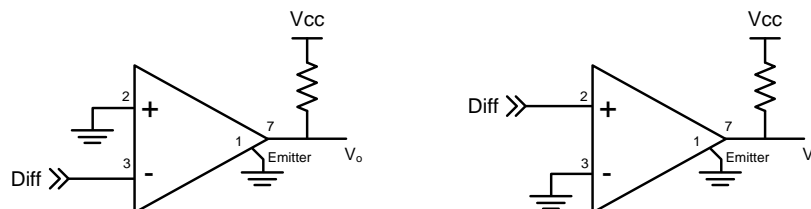
$$v_o(t) = -RC \frac{dv_I(t)}{dt} \quad (3.19)$$

This means that the output is proportional to the derivative of the signal, being the differentiator time-constant  $RC$  the proportionality constant.

A capacitor was connected in parallel with the resistor  $R$  to limit the signal bandwidth as the differentiator is, by definition, a noise amplifier [61]. By practical experimentation, a capacitance of 33 nF was found to be a good choice.

Apart from a slight (and expected) delay, maxima and minima of the sinusoidal wave occur approximately at the same time of those of the original signal, as Appendix B shows. This means that with this circuitry, one can detect where an extrema has occurred by analysing the sign of the differentiator. For that a voltage comparator was used – the LM311 [79].

Comparators usually use “open collector” output with grounded emitter and pulled-up collector (5V in our case), as Figure 3.13 shows. This way, the output swings from 5V to 0V [61]. When the voltage on the non-inverting input is higher than the voltage on the inverting input, there is no current. Therefore, there is no voltage drop across the pull-up resistor and the comparator output is 5V. On the contrary, when the non-inverting input is lower than the inverting one, a current appears and, as the emitter is connected to the ground, the comparator output is around 0V. In fact, that voltage is 230 mV, which is the collector-to-emitter voltage of the transistor. When both inputs voltages are similar, the comparator tends to oscillate, triggering multiple logic changes. This can be avoided by positive feedback, the so called Schmitt triggering that introduces hysteresis in the system [61]. This fact was not observed by the PPG signal, but this technique can be employed in the future, if needed.



**Figure 3.13 – LM311 configuration: positive derivative (left) and negative (right) positive identification.**

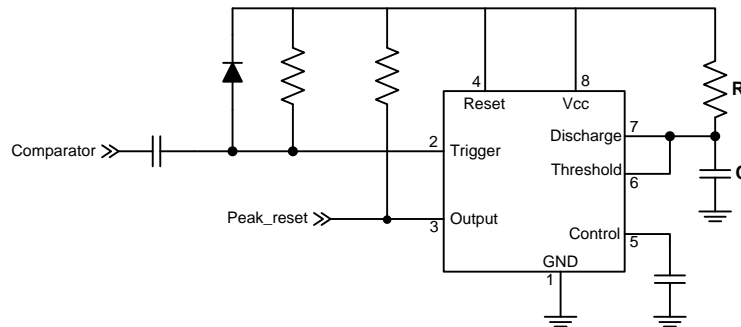
The LM311 [79] has a response time of 115 ns when going from low to high level and of 165 ns when going from high to low level [79]. This makes it suitable for the desired application, since this is a negligible delay for a cardiac wave which has a period of around 1 second.

The output from the comparators can be used to “tell” the MCU that a new peak or valley has been detected and that the digitalization must start. For that, these outputs are used as interrupts that call a sampling routine. This will be explained in more detail in section 4.2.

After the peak or valley voltage has been digitalized, the analogue switch on the peak detector must be closed in order to let the capacitor discharge and then re-opened so that it can charge again to the new peak voltage.

The comparator signal could be a good choice for peak reset as it is synchronised with the peak occurrence. However, it must be considered that in just a fraction of the period (when the signal is decreasing for maxima sampling and while it is increasing for minima sampling) an interrupt has to be generated, the signal must be acquired and the peak detector must be reset. This is particularly important in the case of the systolic peak detection, since the ascending phase of the cardiac wave is much shorter than the descending one. Therefore, a short reset control would be appreciated.

To overcome this, a timer was used. The renowned LM555 [80] was employed to generate a *one-shot* signal each time the LM311 output swings from 5V to 0V (Figure 3.14).



**Figure 3.14 – LM555 monostable mode for *one-shot* signal used for peak detector reset. Adapted from [80].**

In the stable state, the LM555 output is low. When the LM311 output goes from high to low, i.e. when the signal in the trigger pin falls below  $1/3 V_{CC}$ , the LM555 output goes high. A capacitor is used in this pin to differentiate the square wave of the comparator and ensure that the LM555 is triggered. The high state is held for a time determined by the capacitor  $C$  and the resistor  $R$  according to equation (3.20).

$$t = 1.1 \cdot R \cdot C \quad (3.20)$$

This means that choosing the appropriate values of  $R$  and  $C$  one can have the desired pulse duration for closing the analogue switch and thus resetting the peak detector. The chosen values provided a 10 ms pulse width, which is adequate for discharging the peak and valley detectors capacitor.

## 3.2 Accelerometers

As referred before (section 2.3) an accelerometry module is employed in the system to retrieve arm and forearm orientation and thus, offset in height of the finger relatively to the heart.

In this work, the Freescale's digital accelerometer MMA8451Q [81] was used. It is a 3-axis accelerometer with 14-bit resolution and a selectable full-scale of  $\pm 2g$ ,  $\pm 4g$  and  $\pm 8g$ . It communicates through I<sup>2</sup>C (Inter-Integrated Circuit) with an output data rate of 1.56 to 800 Hz. It has some embedded functions as freefall, motion, tap, jolt and orientation detection. These functions are configurable to two interrupt pins which may be helpful in a more advanced stage of this work, allowing power saving and wakeup interrupts to be generated whenever there is a significant change in the accelerometer readout, i.e. if the arm moves significantly [81].

For its choice contributed the high sensitivity and selectable acceleration ranges. Since it is not clear at this point which is the ideal sensitivity, this accelerometer allows great flexibility. Besides, the  $\pm 2g$  range is close enough to the  $\pm 1g$  range needed for static measurements but if greater accelerations are needed in the future (such as for artifacts detection),  $\pm 4g$  or  $\pm 8g$  can be selected. With  $\pm 2g$  range, the sensitivity is 4096 counts/g with an accuracy of  $\pm 2.5\%$ .

The accelerometer has to be interfaced with the MCU. Namely, the logic levels of SDA and SCL lines have to be converted from the 5V logic level of the Arduino™ to the 3.3V of the MMA8451Q. That is done with a voltage-level translator, the PCA 9306 [82]. This is a bidirectional translator adapted for I<sup>2</sup>C SDA and SCL lines. It can be operated to translate voltages from 1.8V – 5.5 V to voltages from 1.2V – 3.3V. The 3.3V are supplied directly from an Arduino™ pin.

## 3.3 Force sensor

As referred before (section 2.4), the PPG signal amplitude depends on the applied force over the measured finger. Therefore, a force sensor was introduced in the system, a Force Sensing Resistor® (FSR®) that has been previously referred. The FSR® is a variable resistance based on the applied force. Therefore, it requires a few signal conditioning, which will be overviewed in this section.

The FSR<sup>®</sup> data sheet [50] presents two basic electrical interfaces: (1) a voltage divider followed by a buffer and (2) a current-to-voltage converter. The first option is schematised in Figure 3.15. With this, the output voltage increases with increasing applied force. However, swapping  $R_{FSR}$  with  $R_G$ , it is possible to have a voltage decrease with force increase. The value of resistor  $R_G$  must be adequately chosen for maximum force sensitivity range. It is also used to limit the current that passes through the FSR<sup>®</sup>, which should not exceed  $1\text{mA}/\text{cm}^2$  of applied force. Further resistances can be used to be used to define the rest voltage, i.e. a zero-force intercept value [50].

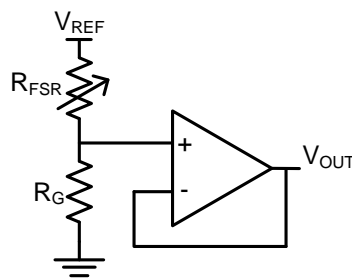


Figure 3.15 – FSR<sup>®</sup> voltage divider. Adapted from [50].

As an ordinary tension divider, the output is related to the FSR<sup>®</sup> resistance ( $R_{FSR}$ ) according to equation (3.21).

$$V_{out} = \frac{R_G}{R_{FSR} + R_G} V_{REF} \quad (3.21)$$

Alternatively, it is possible to use a current-to-voltage converter consisting of an inverting operational amplifier, being the FSR<sup>®</sup> the input resistor. This strategy is schematised in Figure 3.16.  $V_{REF}$  can either be positive or negative. Although, since the ADC inside the Arduino<sup>™</sup> expects a positive voltage, a negative reference voltage is used.

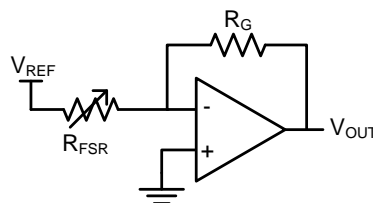


Figure 3.16 – FSR<sup>®</sup> current-to-voltage converter. Adapted from [50].

The output is inversely proportional to the FSR<sup>®</sup> resistance, according to equation (3.22).

$$V_{out} = -\frac{R_G}{R_{FSR}} V_{REF} \quad (3.22)$$

Again, the maximum current through the FSR<sup>®</sup> should be 1 mA/cm<sup>2</sup> and it is also possible to add a resistor in parallel with  $R_{FSR}$  to define the rest voltage [50].

As stated before, the response of the FSR<sup>®</sup> is not linear. However, some configurations may help to make the output more linear with the applied force. Hall et al. [83] proposed the configuration shown in Figure 3.17.

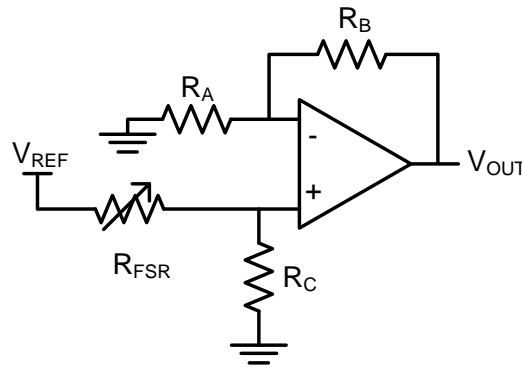


Figure 3.17 – FSR<sup>®</sup> conditioning circuit to improve the output linearity. Adapted from [83].

The output is not linear and is given by the following equation (3.23) [83].

$$V_{out} = \frac{R_C(R_A + R_B)}{R_A(R_{FSR} + R_C)} V_{REF} \quad (3.23)$$

Resistors  $R_A$  and  $R_B$  can be adjusted to the wanted gain of the system. Therefore,  $V_{out}$  is proportional to  $\frac{R_C}{R_{FSR} + R_C}$ . Given that, resistor  $R_C$  is crucial to output linearization.

Two extreme cases can occur: (1) if  $R_C \gg R_{FSR}$ , this equation tends asymptotically to  $\frac{(R_A + R_B)}{R_A} V_{REF}$ , i.e. the output saturates at a constant level which is independent of the applied force; (2) if  $R_C \ll R_{FSR}$ , this equation tends to the hyperbola  $1/R_{FSR}$ . Thus, value of  $R_C$  must be chosen according to the output range one wants to linearize.

This configuration was not tested due to lack of time, but it can be a good choice if the FSR<sup>®</sup> is to be used in further versions of the system. The other two were tested and the best one was chosen to calibrate the sensor (section 5.9).

### 3.4 Microcontroller – the Arduino™

The MCU is the “heart” of the system. It controls all the other components, such as the LEDs ON/OFF, the gain and level shifts of the signal conditioning, the acquisition of the PPG signal, as well as the accelerometer and force sensor readings. Furthermore, the MCU is responsible for the computation of  $HR$  and  $SpO_2$ . Finally, it sends all the information to the computer, where it is displayed in MATLAB. For these tasks, an Arduino™ was employed.

The Arduino™ is an embedded computing platform that consists of a single-board hardware for an Atmel® AVR® microprocessor and a software suite for its programming. The board (Figure 3.18) integrates the microprocessor, a crystal that works as a clock, a 5-volt regulator for power supply, a USB socket to connect it with a computer and an I/O interface with the pins exposed to facilitate integration with other hardware [84].

The Arduino™ programming is done in a computer in an Integrated Development Environment (IDE) where programs (called *sketches*) are written in C/C++ language and then upload to the microprocessor. Moreover, it has a serial monitor to visualize information sent by the serial port of the Arduino™ [84]. Figure 3.18 shows the aspect of the Arduino™ IDE window.

Both Arduino™ hardware and software are open source which makes it a very used platform for all kind of applications. That leads to a huge amount of information, hardware schematics and firmware sketches available on the internet. Also, the *Arduino™ language reference* has a variety of built-in functions for communication, ADC reading, timer configuration, interrupt routines and so on, which makes it real easy to use. Moreover, the ease of developing stand-alone devices or connecting it to a computer or network for further processing makes the Arduino™ an ideal choice for the developmental versions of the system.

The first two versions of the system had an Arduino™ Duemilanove with the ATmega 328. However, for reasons related with the number of analogue input ports (ADC channels), this was replaced. Thus, the current version uses an Arduino™ Mega 2560 with the ATmega2560. With this, the number of input pins increased from 6 to 16.



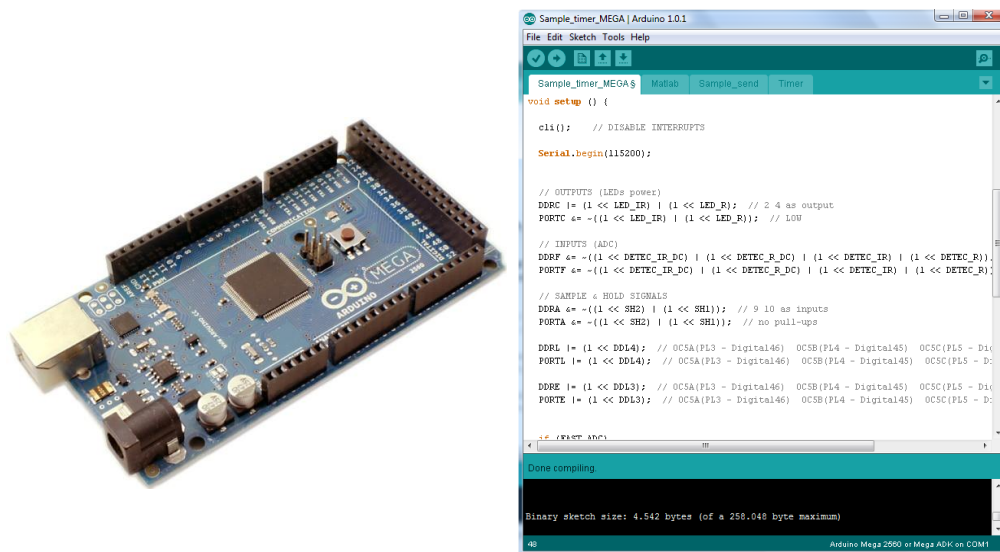


Figure 3.18 – Arduino™: the Mega 2560 board with the ATmega2560 (left) and the IDE window with a piece of a used sketch (right).

### 3.4.1 Processor

The Duemilanove and the Mega 2560 are very similar. Both are 8-bit Atmel® AVR® microcontrollers with RISC (reduced instruction set computers) architecture and 16 MHz of system clock. They have 32 8-bit working registers directly connected to the Arithmetic Logic Unit (ALU) which can be accessed in one single instruction executed in one clock cycle [85, 86].

### 3.4.2 I/O Ports

The significant difference between them is the number of I/O pins. The Duemilanove has only 20 I/O pins, being 6 of them analogue inputs for the ADC. Conversely, the Mega 2560 has a total of 70 I/O pins and 16 of them are analogue. However, both devices have the same 10-bit ADC, i.e. there are 1024 quantization levels [85, 86].

Analogue ports are used as inputs from AC and DC+AC components of both LEDs. The photodiode output is also sampled as a control signal. Maxima and minima voltages held by the peak and valley detector are also sampled for posterior  $SpO_2$  calculation. Besides, the FSR® is acquired via an analogue pin. Digital pins are used to control LEDs ON/OFF as well as the S/H. A digital pin is also available for synchronisation purposes, as all the sensors of the multichannel system will need to be temporally matched. Some digital pins are also used as external interrupts, namely from the peak and valley detector (to receive the sampling command) and eventually from the accelerometers (to indicate movement).

### 3.4.3 Timer/Counters

The Mega 2560 has 2 8-bit and 4 16-bit Timer/Counters with separate prescalers and Capture and Compare modes. Currently, a 16-bit timer is being used to control the LEDs timing sequence, as well as the signal sampling. This way, the synchronisation between finger illumination and signal detection is assured. Another 16-bit timer is used to determine the signal periodicity, and thus the *HR*.

### 3.4.4 Interrupts

The Arduino™ has two types of interrupts: *external* and *pin change*. There are six external interrupts on Mega 2560, although two of them are also dedicated to I<sup>2</sup>C. These trigger an interrupt on rising edge, falling edge, level change or low state. The number of pin change interrupts is bigger, with the Mega 2560 having 24 grouped into 3 ports. However, there are only 3 interrupt vectors (one for each port). Whenever one of the 8 pins of the port changes its state, the corresponding interrupt vector is triggered. Thus, it is up to the programmer to figure out which pin has changed and to what state [85, 87].

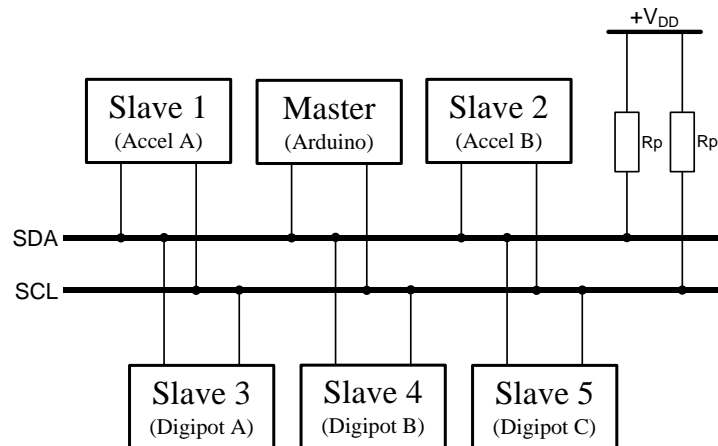
In the designed system, four external interrupts were used to trigger the peak and valley detector sampling – maxima and minima signal of each wavelength. Moreover, two external interrupts are connected to accelerometers to eventually detect movement.

### 3.4.5 Communication

The Arduino™ has an Universal Synchronous and Asynchronous Serial Receiver and Transmitter (USART), a master and slave Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I<sup>2</sup>C) [85, 86].

The simplicity of I<sup>2</sup>C makes it very useful for connecting I/O devices since it just requires two wires. This is the reason why I<sup>2</sup>C is also called 2-wire Serial Interface (TWI). Those two wires are the serial data (SDA) and serial clock (SCL) lines. They are bidirectional and are connected to positive power supply by pull-up resistors. Each device is recognised by its unique address and can act as a master or as a slave. The number of allowed devices depends only on the bus capacitance, which must not exceed 400 pF, and of course the number of allowed addresses (usually, 7-bit addresses are used). The master sets the clock and initiates the data transfer. I<sup>2</sup>C allows for transfer rates of 100 kbit/s in the standard mode or 400 kbit/s in the fast mode [88].

I<sup>2</sup>C was used to interface all the digital peripherals (accelerometers and digipots) due to the ease to connect many devices on the same bus. The configuration is schematized in Figure 3.19.



**Figure 3.19 – Connection of I<sup>2</sup>C-bus devices to the I<sup>2</sup>C bus. Configuration used in the system, with the Arduino™ acting as master and the accelerometers and digital potentiometers as slaves.**

SPI is a full duplex bus, thus allowing for faster transfer rates (up to 12 MHz). Unlike I<sub>2</sub>C, SPI makes use of three wires common to all the devices and one wire dedicated to each device. The common wires are the Serial Data Clock (SCLK), the Master Output/Slave Input (MOSI) and Master Input / Slave Output (MISO). The dedicated line is the Slave Select (SS), which is only active (low) for one device to which the communication is intended. This makes the SPI much more complex if many devices are wanted, due to the number of SS needed [89]. Some configurations allow for the use of a single SS for all devices, creating a daisy-chain topology [90]. However this is a much more complex strategy than the use of I<sup>2</sup>C and is not needed for this system.

### 3.4.6 Shields

Some extra modules (called *shields*) are available for Arduino™. They include other devices that can be used for easily extend its capabilities (as the GPS or the LCD display shields), communication (as the Ethernet and Wi-Fi shields) or data storing (as the SD card shield) among many others. Shields are simply plug in the top of the Arduino™ and are ready to use. Still, they leave all the pins in sight, allowing for the normal use of I/O pins [84]. When the system evolves for the stand-alone arterial blood pressure (ABP) storing, the SD card shield may be useful to store the acquired signals and parameters.



# Chapter 4

## Firmware

*For the system to be a stand-alone pulse oximeter, some requirements must be fulfilled. Accordingly, it must compute oxygen saturation and present the photoplethysmographic curve in real time. Therefore, it is obvious the need for some component with calculus capability. That component is the microcontroller unit. Besides, for the pulse oximeter to present correct values, all the components must be synchronized, mainly the light emitting diodes switching and the signal sampling. Thus, a component capable of communicating and even controlling the remaining ones adds immeasurable value to the system. Nevertheless, for all of this to work properly, the microcontroller needs to be programmed for the desired tasks. In this chapter, the firmware employed in the Arduino™ is explained.*

## 4.1 LEDs timing

Since the system employs two light emitting diodes (LEDs) and just one photodiode, the LEDs must work in opposite phase. This means that while one is ON, the other should be OFF. In the previous work [17], a LM555 timer was used. A square wave with a duty cycle close to 50% and a frequency equal to the sampling rate was used. Though, in this work, the potentialities of microcontroller units (MCUs) were exploited. So, a Timer/Counter was used to generate the squared waves needed to feed the LEDs driver (section 4.2) in order to have periodic and alternated LEDs lighting.

The Arduino™ Timer/Counters have several modes of operation: normal mode, diverse Pulse-Width Modulation (PWM) modes and Clear Timer on Compare Match (CTC). It has also several interrupt flags: 3 Compare Match and 1 Overflow. Theoretically, the CTC mode would be the most appropriated, as one could define the desired TOP value in which the Timer/Counter would be reset, thus allowing for fixed frequency operation. However, this mode does not allow for the Compare Match interrupts, which will be useful for sampling command [91]. Therefore, the normal mode was chosen.

Normal mode is the simplest mode of operation. It counts in the up direction (incrementing) and, unlike CTC, has no reset. This way, the Timer/Counter starts at BOTTOM (0x0000) and increments until MAX (0xFFFF). In the following timer clock cycle, it restarts from BOTTOM again and the *Timer/Counter Overflow Flag* (TOV) is set.

The Timer/Counter allows for the choice of the clock frequency via 5 different prescalers ( $f_{CLK}/1$ ,  $f_{CLK}/8$ ,  $f_{CLK}/64$ ,  $f_{CLK}/256$  and  $f_{CLK}/1024$ ) to divide the CPU clock frequency or via an external clock.

The desired frequency for the LEDs sampling was 500 Hz, i.e. the Timer/Counter should restart each 2 ms. The prescaler value must be chosen according to equation (4.1).

$$prescaler = \frac{f_{CLK}}{TOV_{CLK} \cdot MAX} \quad (4.1)$$

where  $f_{CLK}$  is the clock frequency (16 MHz),  $TOV_{CLK}$  is the desired overflow frequency and  $MAX$  is the maximum value of the Timer/Counter (0xFFFF on 16-bit timers).

Computing the prescaler value, one gets 0.49, i.e. the  $f_{CLK}/1$  prescaler must be used. The overflow frequency is given by equation (4.2).

$$TOV_{CLK} = \frac{f_{CLK}/prescaler}{MAX} \quad (4.2)$$

With the  $f_{CLK}/1$  prescaler, the overflow frequency is  $TOV_{CLK} \approx 244$  Hz. However, 500 Hz are desired. To achieve this specification, the Timer/Counter value register (TCNT) must be set to a value higher than 0 on each restart. That value is computed according to equation (4.3).

$$TCNT = 2^{16} - \left( \frac{f_{CLK}/prescaler}{TOV_{CLK}} - 1 \right) \quad (4.3)$$

With timer clock frequency of 16 MHz, a 500 Hz cycle requires  $16\,000\,000 / 500 = 32\,000$  counts. Therefore, for each Timer/Counter restart (after TOV flag) TCNT must be set to 33537 in order to reach an overflow each 500 Hz, i.e. with a period of 2ms from that value to MAX.

Since each LED should be ON for half of the period (i.e. 1 ms), the MCU must toggle two digital pins each millisecond. That is accomplished by two interrupt routines: one triggered by a Compare Match B (OCRnB) in the middle of the Timer/Counter range and the other triggered by the TOV. On each routine both pins are toggled. Those pins are the *LED\_power1* and *LED\_power2* inputs for the LED driver described in section 3.1.1.

The values set for the registers to configure interrupts are shown in Table 4.1.

**Table 4.1 – Timer/Counter 5 registers used for interrupt routines to control LEDs lighting and signal sampling timing.**

Register	Value
TCNT5*	33537
OCR5A	41537
OCR5B	49536
OCR5C	57535
TOV5**	65535

\* TCNT5 is set on each restart (when TOV5 flag is set)

\*\* TOV5 is not set, it is the MAX value and was presented just for information

## 4.2 Photoplethysmogram

As far as signal sampling is concerned, two major aspects are controlled by the MCU firmware: the sampling rate (i.e. when the signal is sampled) and the sampling routine (i.e. the instructions to actually sample the signal). In this section, questions related to the signal acquisition are discussed. Aspects related with signal conditioning are not considered as they will be addressed later (section 4.5).

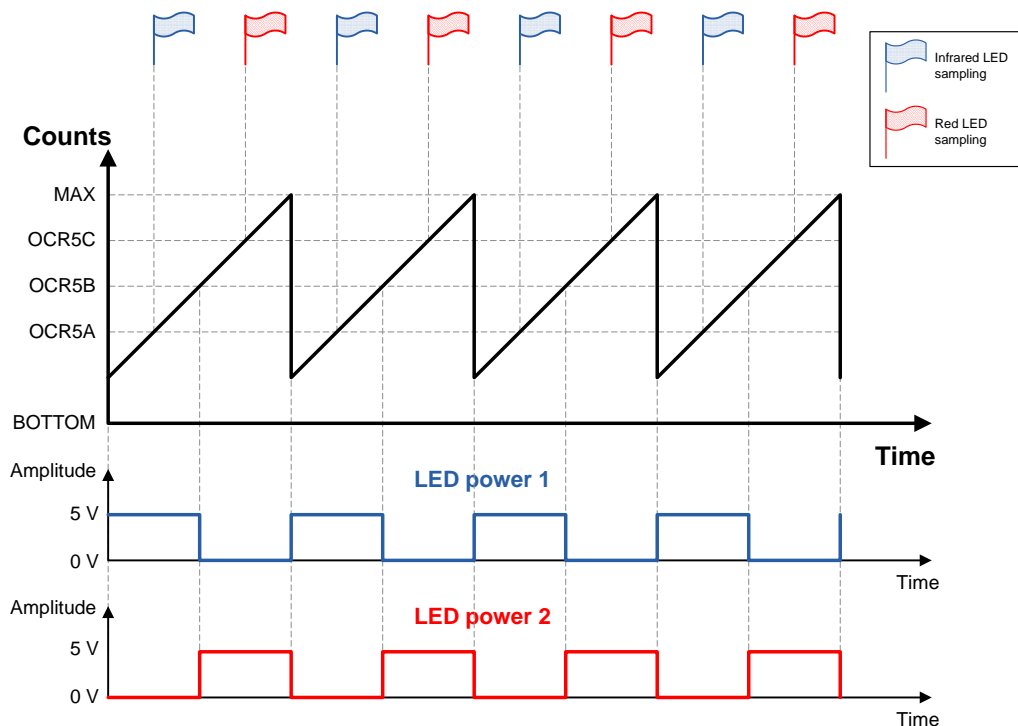
In order for the system to work properly, the signal must be acquired when each LED is ON. Therefore, synchronisation between signal sampling and LED lighting is critical. Since a Timer/Counter was used to control LEDs ON/OFF timing, and given that it is possible to set interrupt flags during the timer running cycle, it is logical that one takes benefit of that. This way, two interrupt flags are used to trigger sampling routines (one for each wavelength).

As deduced in section 2.2.3, oxygen saturation ( $SpO_2$ ) measurements require just the extrema of the signal. Its identification is accomplished by hardware with the peak and valley detector (section 3.1.5). For displaying purposes, it was decided to use also the AC component *per si*, since it is more stable and easy to display in a waveform monitor. Therefore, continuous sampling of the DC+AC component is not needed and just the AC component would be sampled at Timer/Counter interrupt flags. However, as a matter of system testing, in these first prototypes, the DC+AC component will be sampled. Therefore, when an interrupt routine is called, it first samples the AC channel and then samples the DC+AC channel of the respective LED.

As a matter of stability, the signal must be sampled at the middle of ON period of each LED. This avoids transitory regime of the LEDs and the sample-and-hold (S/H) which could introduce noise. As seen above, the Timer/Counter has 3 Compare Match registers (OCRnx) that can be used to trigger interrupts. OCRnB was already used to toggle LEDs power lines. Thus, OCRnA and OCRnC can be used here to trigger the sampling routines. Table 4.1 shows the register values chosen in order to assure correct timing.

Figure 4.1 presents an overview of the Timer/Counter operation, evidencing the usefulness of this kind of devices in synchronization. As it can be seen, on each restart and at the middle of the operation range, *LED\_power1* and *LED\_power2* are toggled, thus inverting the LED polarity. This ensures that the when one LED is ON the other is OFF and no mixed signals are obtained.





**Figure 4.1 – LEDs lighting and signal sampling scheme.** Timer/Counter increases in normal mode and has 4 interrupt counter values (middle). In the middle and at the end of the Timer/Counter, OCR5B and MAX interrupt flags whose routines are toggling LEDs power lines are triggered (bottom). In the middle of each LED ON period (OCR5A and OCR5C), interrupts whose routines are to sample the respective LED signal are triggered (top).

Then, in the middle of each half of period, interrupt flags are set in order to sample the respective LED signal. This guarantees that the signal is sampled in a stable region of operation, avoiding noise from transient regimes.

The code flow starts when the OCR5A flag is set. It will trigger the sampling routine for infrared signal. First, infrared AC component is read by the ADC. Afterwards, data is stored in a variable for posterior calculations and finally, it is sent to the computer. Then, the infrared AC+DC signal is sampled, stored and sent. On OCR5C flag, red sampling is triggered. Likewise, red AC channel is read, stored and sent and the same is done to the AC+DC channel.

The sampling command is done by a built-in function of Arduino™, the *AnalogRead*. The ATmega2560 has a 10-bit successive approximation Analogue-to-Digital Converter (ADC) of 16 channels. Therefore, every conversion is done by comparing the analogue input with the possible  $2^{10}$  (i.e. 1024) quantization levels. The AREF (Analogue Reference pin for the ADC) indicates the input range for the ADC. It can be defined

externally by applying a voltage to the AREF pin. By setting the REFSn bits in ADMUX Register, external or internal 1.1 V / 2.56 V may be selected.

As data sampling is not the main goal of Arduino™, it has a relatively low default sampling rate. As any ADC, the maximum reachable sampling rate is limited by the time the ADC takes to digitalize a continuous waveform (analogue variable) to a digital code discrete time (digital variable). The ATmega2560 ADC requires a clock frequency between 50kHz and 200 kHz for maximum resolution and accuracy [92]. However, from 200 kHz to 1MHz, ADC resolution is not significantly affected [46].

The ADC clock frequency is generated by a prescaler from the CPU clock frequency. Different ADC clock frequencies can be defined, setting the ADC Prescaler Select Bits (ADPS) in ADC Control and Status Register A (ADCSRA) according to Table 4.2 [92]. Considering that the default prescaler is set to 128 and the Arduino™ clock frequency is 16 MHz, the ADC clock frequency is set to 125 kHz. Now, taking into account that a normal conversion takes 13 ADC clock cycles, a conversion will take 104  $\mu$ s to be completed. Whenever a shorter conversion period is required, one may change the prescaler. If the prescaler is set to 16 for example, the ADC clock frequency is 1 MHz and a conversion takes 13  $\mu$ s, thus 8 times faster. However, it is recommended to keep this clock frequency as low as possible in order to maximize the ADC accuracy and resolution. Furthermore, prescaler values under 16 would give ADC clock frequencies above 1 MHz, which were not characterized by Atmel® and thus are not recommended [92].

**Table 4.2 – ADPS [2:0]: ADC Prescaler Select Bits. These bits determine the division of the system clock to set the ADC clock [92].**

ADPS2	ADPS1	ADPS0	Division factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

### 4.3 $SpO_2$ computation

For  $SpO_2$  to be computed, maxima and minima of the PPG signal must be known. In this section, some approaches for their computation will be discussed, namely the previous developed and the currently implemented.

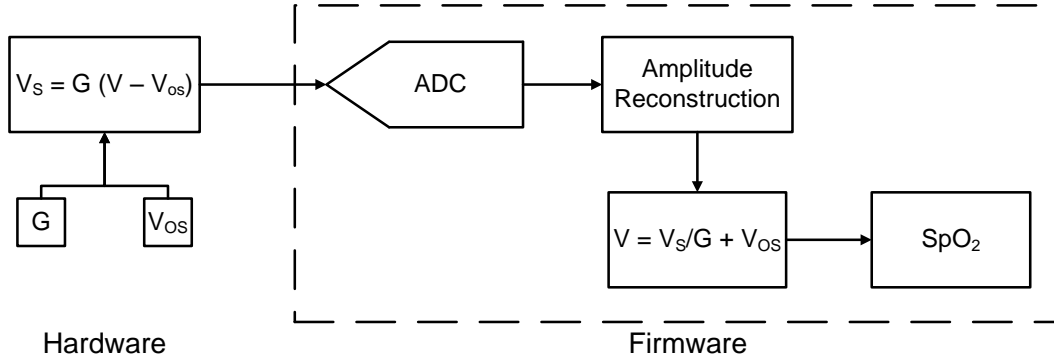
In the previous work of Brás [18], an algorithm was developed to search for maxima and minima through iterative and continuous search of higher and lower values in the signal. First, the algorithm computed the signal mean. Then, it searched a maximum during the time in which the signal was higher than the mean. Likewise, a minimum was searched when the signal was lower than the mean. It also incorporated a time threshold to discard false peaks due to noise or even to the dicrotic peak and valley.

However, this task was done offline, with the algorithm running upon stored data of prior acquisitions. As the purpose of this work is to have a real time oximeter, this solution was not acceptable. Besides, the iterative search of a new extrema is too time consuming and thus, it is not suitable for the current work. This way, another approach was used and the peak and valley were identified via hardware (section 3.1.5).

As far as the MCU is concerned, it “knows” when a new extrema has occurred by an interrupt signal provided by the hardware. The interrupt is generated on the rising edge of each LM311 output signal, which occurs once in each cardiac cycle.

The LM311 output signal is connected to two external interrupt pins. One signals an available maximum while the other signals an available minimum. This way, the Arduino™ knows exactly if there is a maximum or a minimum ready to sample. Once their voltages are sampled, they are stored in the MCU memory and used to compute  $SpO_2$ . For now, just the last value is being used. However, some algorithms are based in prior values to deal with periods of increasing or decreasing oxygen saturation, which are misestimated when just instantaneous readings are used [24].

Recalling the Ratio of Ratios ( $R_{OS}$ ) equation (2.9), it can be noticed that any gain prior to DC shift is irrelevant to the computation, since it cancels in the numerator and denominator. However, when a voltage is subtracted to the signal,  $R_{OS}$  is changed since a subtraction is not cancelled. Therefore, when the offset value is generated by the digipot, it must be stored. The same happens with the subsequent gain. Those values are used for posterior signal reconstruction, in order to get an accurate  $R_{OS}$ . The procedure is depicted in Figure 4.2.



**Figure 4.2 – PPG signal reconstruction algorithm.** The output of the instrumentation amplifier is sampled and then the signal is reconstructed in order to retrieve its original amplitude.

As seen in section 2.2.3, commercial pulse oximeters have a calibration step that requires blood samples to be taken. However, this was not possible to do during this work. An *in vitro* calibration procedure may be done using finger models. Blood with variable oxygen content may be used to retrieve a calibration curve. This would be extremely useful to improve accuracy and can be done in the future.

For now, as no calibration was done, the best that can be done is using the theoretical Beer-Lambert's law. Assuming the signal reconstruction,  $R_{OS}$  is computed according to equation (4.4).

$$R_{OS} = \frac{\ln\left(\frac{I_{min}^R/G^R + V_{OS}^R}{I_{max}^R/G^R + V_{OS}^R}\right)}{\ln\left(\frac{I_{min}^{IR}/G^{IR} + V_{OS}^{IR}}{I_{max}^{IR}/G^{IR} + V_{OS}^{IR}}\right)} \quad (4.4)$$

where  $G^R$  and  $G^{IR}$  are the instrumentation amplifier gain for the red and infrared channels,  $V_{OS}^R$  and  $V_{OS}^{IR}$  are the offset shifts,  $I_{min}$  are the systolic voltage levels and  $I_{max}$  are the diastolic values.

That computation is done after each minimum has been sampled. Subsequently, the  $R_{OS}$  value is used to compute  $SpO_2$  following equation (4.5).

$$SpO_2 = \frac{0.81 - 0.21 \cdot R_{OS}}{0.81 - 0.08 + [0.30 - 0.21] \cdot R_{OS}} \quad (4.5)$$

## 4.4 Heart Rate computation

Technically, Heart Rate ( $HR$ ) is the heartbeat frequency. Therefore, an obvious source of information to compute  $HR$  is the signal from extrema identification. In fact, the peak and valley detector (section 3.1.5) generates an interrupt signal on each systolic peak and diastolic valley. This means that if one makes use of that signal, the cardiac period can easily be determined. This way, the signal from the LM555 that is used to signal the PPG maximum is used to restart a Timer/Counter. That Timer/Counter measures the time interval between systoles, thus measuring the cardiac period. Maximum signal interrupt was used in detriment of the minimum interrupt, since this is the one that requires less computation associated to its routine. It only has to sample the maximum, reconstruct the amplitude and store it. The diastolic interrupt, on its side, besides those functions, has to compute  $R_{OS}$  and  $SpO_2$  (section 4.3).

It is expected that the cardiac cycle period in resting conditions lasts between 0.67 seconds (for 90 bpm) 1.2 seconds (for 50 bpm) [93]. Therefore, the overflow time should happen within these values. Using again one of the available 16-bit Timer/Counters, and following equation (4.1), it comes that the only prescaler that allows for 1.2 seconds of overflow timer is the 1024.

Therefore, the interrupt routine triggered by the LM555 includes reading the TCNT4 register (Timer/Counter 4 count) and restart it. The TCNT4 is then used to compute the  $HR$  according to equation (4.6).

$$HR = \frac{TCNT4 - 1}{f_{CLK}/prescaler} = \frac{TCNT4 - 1}{15625} \quad (4.6)$$

The  $HR$  value is then store. For now, just the last value is used. Nevertheless, previous values are useful to compute a weighted average in order to handle with eventual incorrect readings. Also, it would be useful to have thresholds for heart cycle period. That would allow for automatic rejection of incoherent values, such as very large values (driven by a missed pulse) or very small (driven by noisy multiple pulses).

## 4.5 Gain and level shift control

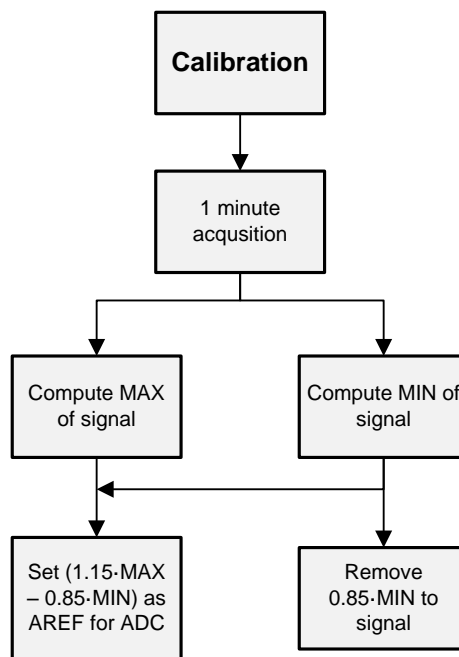
All the previous sections assumed a well behaved signal. However, this is not an easy presuppose to achieve. In fact, some conditioning circuitry addressed before (section 3.1.4) requires some control from the MCU, namely, the signal gain and level shift. In this section an overview of the tried approaches will be addressed. Then, a possible approach to design an automatic controller to the systems is discussed.

### 4.5.1 Approaches

None of the previous developed works had controllable level shift or gain. The reason is that the work of Pereira [16] was just a project and did not reach practical tests. The work of Domingues [17] and Brás [18], on its turn, did reach experimental tests but the acquisition was done with a NI USB-6009 from National Instruments™, a 14-bits data acquisition system. As it has a very good resolution, this means that the need for sampling just the AC component was not discovered. In this work, however, as the Arduino™ has just 10 bits, it is required that the DC level is removed in order to have acceptable signal quality. The remaining AC component must then be amplified in order to match the ADC range.

The first version of the current system did not have a manner of systematically adapt the signal level. Instead, it had an initial calibration process which recorded the signal amplitude over a period of time (for instance, 1 minute). After, the signal maximum and minimum were computed. A fraction of the minimum was used to subtract to the original signal, thus allowing for DC removal trough an instrumentation amplifier. The fact of using a fraction (e.g. 0.85) and not the entire value was a safety margin to account for eventual signal variations. In an equivalent manner, a factor greater than 1 was also applied to the maximum (e.g. 1.15). The value  $1.15 \cdot max - 0.85 \cdot min$  was used as a measurement of signal amplitude and was input to the AREF pin to set the ADC dynamic range. Figure 4.3 shows the flowchart of this calibration procedure.

However, the PPG signal has slow fluctuations due to physiological conditions [32, 33]. In practice, these fluctuations were very pronounced and this calibration was not enough. In fact, after the signal calibration the signal rapidly went outwards the defined range, making it invisible to the Arduino™ ADC. Therefore, a different approach was then implemented.



**Figure 4.3 – Calibration procedure in the first version of the system.**

In the current version, the system is provided with several AD5252 [73] digital potentiometers (digipot) that are used in signal condition (section 3.1.4). Namely, they are used to shift the signal in order to remove the strong DC component that would fill almost the entire range of the ADC and to set the gain for AC component in order to match it to the that range.

To set the DC level shift, a constant DC voltage must be generated and fed to one of the inputs of an instrumentation amplifier. Therefore, the digipot is used in the potentiometer mode, with one terminal at GND and the other at +5V. To change the DC level, the digipot wiper must be moved towards GND terminal towards the +5V terminal, depending on the desired voltage. The wiper position is digitally controlled by the RDAC registers. The AD5252 is a dual digipot, so it has two RDAC registers: RDAC1 (0x01) for channel 1 wiper and RDAC (0x03) for channel 2 register. Since this is an 8-bit digipot, a value between 0 and 255 must be set. Recalling equation (3.12), and since the potential across the digipot is 5V, the output voltage is given by equation (4.7).

$$V_o = \frac{D}{256} \cdot 5 \quad (4.7)$$

where  $D$  is the wiper value, between 0 and 255.

To set gain, the digipot is used in the rheostat mode, whose resistance is fed between the pins 1 and 8 of the instrumentation amplifier (section 3.1.4). To define the resistance value, the wiper is moved in the same way used for the potentiometer mode: changing the RDAC1 and RDAC3 registers. Using equation (3.11) and knowing that the total resistance is 100 k $\Omega$ , the resistance is given by equation (4.8) according to the wiper position  $D$  (between 0 and 255).

$$R = \frac{D}{256} \cdot 100k + 75 \Omega \quad (4.8)$$

Digipots are controlled by I<sup>2</sup>C. So, the first thing to do is to write the corresponding address in the Serial Data line (SDA). That address is unique and is used to specify for which digipot is the information intended for. The AD5252 has 4 different I<sup>2</sup>C addresses configurable via 2 external pins that can be held at 0V or 5V, allowing for 4 digipots on the same bus (Table 4.3). This is useful as the system has 3 dual digipots: a potentiometers for DC level shift and a rheostat for DC+AC gain setting of red channel; the same for infrared channel; and two rheostats for AC gain of both channels [73].

**Table 4.3 – AD5252 I<sup>2</sup>C 7-bit address, according to AD1 and AD0 pin voltages [73].**

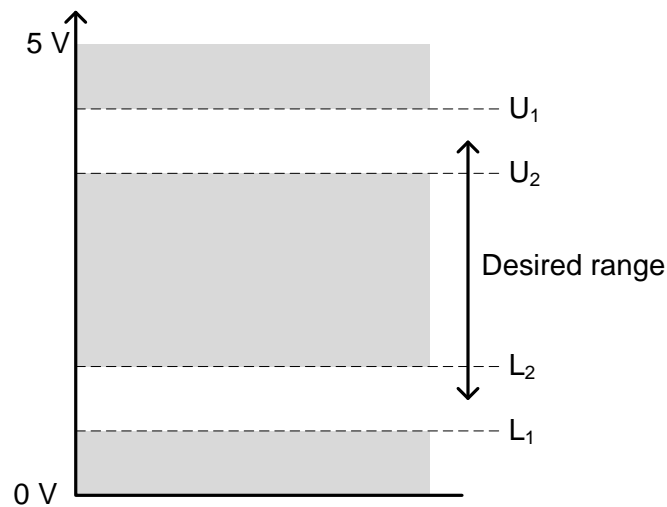
Device	AD1	AD0	I <sup>2</sup> C Address
DC+AC IR conditioning	0V	0V	0 1 0 1 1 0 0
DC+AC R conditioning	0V	5V	0 1 0 1 1 0 1
AC conditioning	5V	0V	0 1 0 1 1 1 0
–	5V	5V	0 1 0 1 1 1 1

Now that each device is uniquely identified, information can be exchanged between the digipots and the MCU. The communication commands will be referred in section 4.8.1.

## 4.5.2 Controller

This later conditioning approach allows for real time system adaptation. However, it requires a much more complex controller. Signal must be kept between constant levels to ensure good resolution and to avoid saturation. This controller is not implemented yet. Nevertheless, a theoretical solution was already idealised and will be described in this section. This possible approach is to define a few thresholds to control the signal amplitude. Figure 4.4 shows those thresholds.





**Figure 4.4 – Representation of the desired signal ranges and threshold values for gain and offset shift update.**

The ideal condition is when signal maximum is between  $U_1$  and  $U_2$ , while its minimum is between  $L_1$  and  $L_2$ . When the signal exceeds  $U_1$ , it is too close of the ADC upper limit, which means that it is dangerously close to fall outside ADC range and the digital output will be saturated. Conversely, when it decreases below  $L_1$ , it is dangerously close to reach the ADC lower limit. Therefore, periodic checks must be done in order to detect when the signal falls outside the ideal ranges. Otherwise, comparators and logic gates may be combined and connected to the interrupts pins of the MCU to automatically detect these events without spending processor time. A possible controller to accomplish this is the one schematised in Figure 4.5.

When the signal exceeds  $U_1$ , three possible conditions can occur: (1) the signal minimum is below  $L_1$ , meaning that the signal amplitude is too high – therefore, the gain must be reduced and the signal shifted upwards; (2) the minimum is above  $L_2$  and the signal has to be shifted downwards; (3) the minimum is inside the desired range, indicating a large amplitude and thus the gain must be reduced – however, this will lead to a drop in the signal voltage and it must be shifted upwards.

If the signal maximum decreases below  $U_2$ , other three situations may happen: (1) the signal minimum is below  $L_1$  so, it must be shifted upwards; (2) the minimum is above  $L_2$ , meaning that the amplitude is too low and the gain must be increased – to account for signal amplification, it must be shifted downwards; (3) the minimum is in the expected range, implying that the gain must be increased and consequently, the signal shifted downwards.

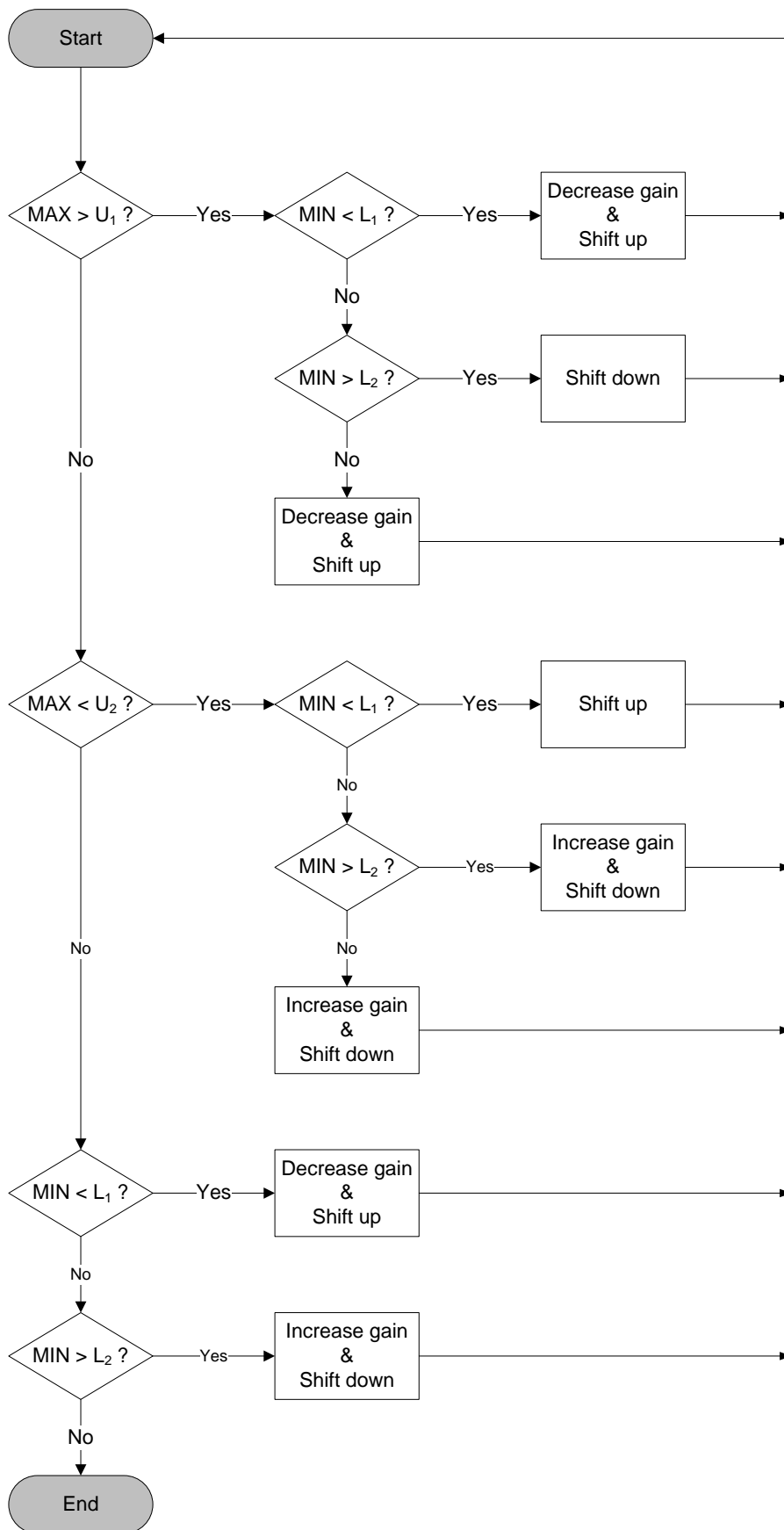


Figure 4.5 – Flowchart of a possible approach of a controller to adjust signal gain and DC level.

When the signal falls below  $L_1$ , just one non-verified condition must be happening: (1) the maximum is inside the desired range, meaning that the amplitude must be decreased and hence, the signal must be shifted upwards. The case of having a maximum above  $U_1$  or below  $U_2$  was already accounted for.

If the signal minimum rises above  $L_2$ , one condition must be occurring: (1) the maximum is inside the desired range and thus, the gain must be increased and a downwards shift must be imposed. The cases of having the maximum outside the desired range, either above or below, were already referred.

All the cases detailed above are illustrated in Appendix C. It is worth to notice that some of these rules are not unique. As an example, when both gain and level shift are changed, it may be better to accomplish just one of the tasks at a time. Or when one of the extrema is inside the desired range, a gain adjustment may be enough, instead of both gain and offset change. Different strategies may allow for different system reactions, allowing for a more flexible or a harder control. Moreover, quantization of the rules must also be defined for the desired corrections.

It must be taken into account that no practical implementation of this controller was done. This scheme is simply a sketch of the controller. Much work is still to be done, being this an important task for the future work. For now, the digipots settings are done by software. Six sliders exist in a graphical user interface (GUI) that was designed and is explained in more detail in section 5.10. It is up to the user to regulate the values of the digipot looking to the signal and changing the sliders, which requires some practice and knowledge about the system.

## 4.6 Height measurement

A parameter that this system proposes to measure is the finger height. That is accomplished by the use of two accelerometers to register the arm and forearm orientations and, knowing the length of those segments, compute the height variation (section 2.3).

To start, the accelerometers must be configured to accomplish the needs. Since speed is not an important issue, Output Data Rate of 1.56 Hz is selected in the CTRL\_REG1 register. The same register is used to set the Low Noise option. Moreover, High-Pass Filtering can be activated with different cut-frequencies if it proves to be useful (although

it was not tested yet). Then, to work in the High Resolution mode (14 bits), CTRL\_REG2 is written accordingly. In addition, as a static acceleration is required to know the orientation, the  $\pm 2g$  range is selected in the XYZ\_DATA\_CFG register. The communication with the accelerometers (reading and writing) is done by I<sup>2</sup>C, according to the commands that will be referred later in section 4.8.

Taking into account that the used accelerometers have 3 axis of 14 bits each, a total of 6 bytes must be read, i.e. the value of six registers must be read: OUT\_X\_MSB (0x01), OUT\_X\_LSB (0x02), OUT\_Y\_MSB (0x03), OUT\_Y\_LSB (0x04), OUT\_Z\_MSB (0x05) and OUT\_Z\_LSB (0x06). As these registers have sequential register addresses, a multi-byte read can be done, with the MMA8451Q automatically incrementing the register address after each reading [81].

One can retrieve the three measurements following equations (4.9), (4.10) and (4.11).

$$x = (\text{OUT\_X\_MSB} \cdot 256 + \text{OUT\_X\_LSB}) \cdot (1/4096) \quad (4.9)$$

$$y = (\text{OUT\_Y\_MSB} \cdot 256 + \text{OUT\_Y\_LSB}) \cdot (1/4096) \quad (4.10)$$

$$z = (\text{OUT\_Z\_MSB} \cdot 256 + \text{OUT\_Z\_LSB}) \cdot (1/4096) \quad (4.11)$$

Since the accelerometer has 14 bits and the range is  $\pm 2g$ , the sensitivity is 4096 counts/g, i.e. the reading must be multiplied by 1/4096.

As referred in section 2.3, the orientation can be determined through the roll angle ( $\varphi$ ) according to equation (4.12).

$$\varphi = \text{atan2} \left( a_y, \text{sqrt}(a_x^2 + a_z^2) \right) \quad (4.12)$$

To compute the height, the length of the arm and forearm segments (including the hand and finger) segments must be measured for each subject as well as the distance between the shoulder (the rotation point of the arm) and the heart. Those values are then used to compute the finger offset relatively to the heart according to equation (4.13).

$$h = L_{arm} \sin(\varphi_{arm}) + L_{forearm} \sin(\varphi_{forearm}) + L_{heart} \quad (4.13)$$

where  $L_{arm}$  and  $\varphi_{arm}$  are the length and roll of the arm segment,  $L_{forearm}$  and  $\varphi_{forearm}$  are those variables for the forearm and  $L_{heart}$  is the vertical distance between the heart and the shoulder.

As the digipots, the MMA8451Q has 2 different I<sup>2</sup>C addresses that are configured by the SA0 pin. That addresses are expressed in Table 4.4. This way, two accelerometers can coexist in the same I<sup>2</sup>C bus, as it is required.

**Table 4.4 – MMA8451Q I<sup>2</sup>C 7-bit address, according to SA0 pin voltage.**

SA0	I <sup>2</sup> C Address
0V	0 0 1 1 1 0 0
5V	0 0 1 1 1 0 1

Unlike PPG signal, which requires high sampling rate, or the  $SpO_2$  and  $HR$  computation, which must be computed once in a second, the accelerometers reading may require much less sampling rates. As the accelerometers are used to measure the finger height, it is reasonable to consider that it does not change with high frequency. However, a means of periodically acquire the accelerometers data is still required.

For now, the signal is being acquired once a maximum of PPG is detected. The reason for this is strictly related to the fact that this occurrence was already signalled by an interrupt (section 3.1.5). New sampling routines may be developed in order to not overload these interrupt routines. In particular, the MMA8451Q interrupt flags might be useful to detect movement and read the new arm and forearm orientations. Also, as data sending to MATLAB is not synchronized, the height is computed and stored in the MCU and not sent. Further work still needs to be done in this field.

## 4.7 Contact force measurement

Another variable measured by the system is the contact force between the finger and the probe. That is done using a Force Sense Resistor<sup>®</sup> (FSR<sup>®</sup>), as seen in section 2.4. In opposition to the accelerometers, which give a calibrated output and communicate digitally with the MCU, the FSR<sup>®</sup> is an analogue sensor. Therefore, calibration and ADC conversion are required.

The calibration procedure will be detailed in section 5.9.1. Since the ADC reads the output voltage, its transfer function must then be used to retrieve the applied force. That transfer function is derived in section 5.9.2 and is expressed in equation (4.14).

$$F = 0.072 V_o^2 + 0.376 V_o - 0.086 \quad (4.14)$$

where  $F$  is the applied force on the finger probe and  $V_o$  is the output voltage of the FSR<sup>®</sup> conditioning circuitry.

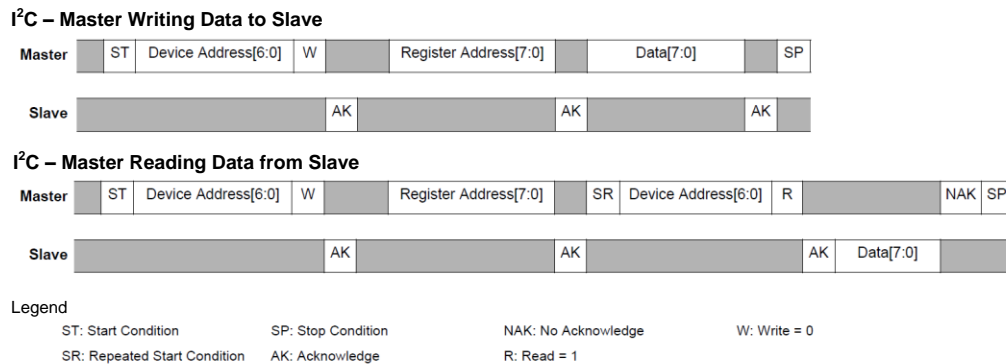
The timing control for the sampling routine is, as in the case of the accelerometer, still not ideal. Further experimentation is required to define the best sampling frequency or the best way to detect the need for a new sampling. Again, for now the sampling order is given by one of the existent interrupt from the peak detector (the dirotic valley in the case). Besides, a manner of sending data to MATLAB in the middle of the PPG data is still required, since at this time, it is not possible to distinguish both data. Therefore, the contact force is computed by the MCU but not sent to MATLAB.

## 4.8 Communication

One of the advantages of having a MCU is the capability of communication with other devices, might that communication be to control them or to transfer data. In the case of communication for devices control (accelerometers and digipots), that is done by I<sup>2</sup>C. In the case of data transfer between the Arduino<sup>™</sup> and MATLAB, that is done by serial communication.

### 4.8.1 I<sup>2</sup>C

As referred in section 3.4.5, the I<sup>2</sup>C requires two wires: serial data (SDA) and serial clock (SCL) lines. While the SCL is a pulsed wave that dictates the transmission rate, the SDA has a series of changes in voltage according to a defined set of commands that follow reading and writing protocols. This reading and writing protocols are schematized in Figure 4.6.



**Figure 4.6** –I<sup>2</sup>C interface general description for writing (top) and reading (bottom) operations. Adapted from [81].

For the writing operation, the communication protocol initiates with a START condition written by the master, followed by the 7-bit slave address and the  $R/\bar{W}$  bit set to 0. Then, the designated slave answers with an ACK (acknowledge) and the master sends the register address byte. Afterwards, another ACK is sent by the slave and the data byte is sent by the master. If the communication is succeeded, slave sends another ACK and the master ends the communication with a STOP.

In the Arduino™, the I<sup>2</sup>C communication is done by the *Wire* library. First, the *Wire.beginTransmission* function is used to send the START condition and the device address. Then, the instruction and data bytes are sent using the *Wire.write* function. Finally, the STOP condition is sent by the *Wire.endTransmission* function.

The reading operation starts, once again, with the START command sent by the master followed by the slave address and  $R/\bar{W}$  bit set to 0. Then, the slave sends and ACK and the master sends the register address that it wants to read. The slave signals the reception with an ACK and the communication is restarted, now with the  $R/\bar{W}$  bit set to 1. Subsequently, the slave sends the data from the requested register. In this case, the master do not sent and ACK, but sends the STOP command to end data transfer.

In Arduino™, this is done by the *Wire.requestFrom* and *Wire.read* functions. The first one is used to send the slave address and define the number of bytes to read; the second one is used to read the data sent by the slave.

## 4.8.2 Serial

The Arduino™ board communicates with the computer (or eventually, other devices) through Serial communication via the USB cable. A few built-in functions are used for

the communication. First, the *Serial.begin* sets the data rate (also called *baud rate*) for communication. Typical values range from 300 to 115200 bits per second (bps). The sampled data has 10 bits, which are transmitted as two bytes: the high byte and the low bytes. Therefore, as there are 4 photoplethysmographic channels acquired with a sampling rate of 500 Hz, at least 32 000 bps are required. The selected baud rate was 115 200. At the other end of the communication, MATLAB connects with the serial port through the function *serial*.

To transmit data, the function *Serial.write* is used. This function writes binary data, i.e. one byte of 0s and 1s, to the serial port. MATLAB receives these data using the function *fread*.

## 4.9 Data storage / sending

When an analogue conversion is made, a digital value between 0 and 1023 is returned by the ADC. Although this value might be used to perform calculations within the MCU, it is useful to store it on a physical memory other than the Arduino™ Flash memory, which has only 248 Kbytes for variable storage. Two principal routes are of special interest in this context: (1) sending this information to a computer for processing and/or storage (e.g. for MATLAB); (2) keeping the information in the system for later visualization and/or processing (e.g. in a SD card).

The first route is of special interest during all process of designing, building and testing the device. Moreover, since the system is to be integrated as a module of a multichannel device, sending information to the computer is crucial. That data can afterwards be analysed and synchronized with other modules in order to extract additional information that comes from that synergy.

The second route is useful if the system is required to work as a stand-alone device to record the patient signal over a large period (e.g. 24 hours), since permanent connection to a computer is unsuitable. That was the initial objective of the project and some tests with the Arduino™ SD card shield were done. However, with the introduction of the multichannel concept, that idea was dropped.



# Chapter 5

## Results

*The system hardware and firmware were already presented in Chapter 3 and Chapter 4, respectively. In this chapter, the results of those electronic circuits and microcontroller code will be overviewed. This will show if the system fulfils the requirements described in Chapter 2. First, a summary of the system versions will be done. Then, the signals from each module will be presented, including the accelerometers and force sensor. The resulting PPG signal acquired by the Arduino™ will be present at the end of the chapter.*

## 5.1 System versions

The system has had some developments due to practical results and to new requirements. Therefore, three major changes led to 3 system versions, from a breadboard to a PCB module of the multichannel system. In this topic a quick overview of those versions will be made, focusing its generic modules and the drawbacks of each previous version.

### 5.1.1 Version 1

The first version of the system consisted on the hardware implemented in a breadboard with the control being done by the Arduino™ Duemilanove. This version was focused on the photoplethysmographic (PPG) signal and thus, no force sensor or accelerometer conditioning nor complex firmware was implemented. Besides, it was power supplied by voltage sources.

This version had no real time DC conditioning as the initial approach was to acquire the signal during a determined period of time for each subject (e.g. 1 minute) to determine the baseline. From that acquisition, two pulse-width modulation (PWM) waves were generated and filtered to yield their DC value. One of these signals was used as DC level shifting (to subtract to the signal in an instrumentation amplifier) while the other was used as external analogue reference (AREF pin) to limit the ADC range to the PPG signal amplitude. Moreover, there was no sample-and-hold (S/H) module and both channels were separated only by the Arduino™ sampling timing. However, this introduced a considerable limitation. As the signal was modulated over a squared pulse of 500 Hz (LEDs powering), an efficient low-pass filtering could not be implemented and the signal was too noisy.

There was also no AC conditioning since it was assumed that the DC component with the constant level removal would lead to stable signals and no further hardware processing would be needed. However, this proved to be a false premise.

The system also lacked the peak and valley detector module. Their determination and the subsequent  $SpO_2$  computation were not done by the MCU. Instead, they were done offline by in MATLAB.

### 5.1.2 Version 2

For the second version, a printed circuit board (PCB) was designed and a new version of the system was developed. The Arduino™ Duemilanove was again the controlling unit.

With the first version, it became obvious that the DC calibration of the first system was neither reasonable nor useful, since PPG signal continuously changed its DC level during acquisition. Therefore, a new approach was implemented. The idea of an initial calibration was left and the conditioning was done by AC filtering. This way, the level changing was no longer a problem since it was intrinsically removed by high-pass filtering. The PPG signal was now ready for display and the results were very good.

However, with AC filtering, the DC level was no longer available. Thus,  $SpO_2$  computation was not possible, as a manner of faithfully knowing the DC level was not implemented. Consequently, a new way of condition the DC level was still required.

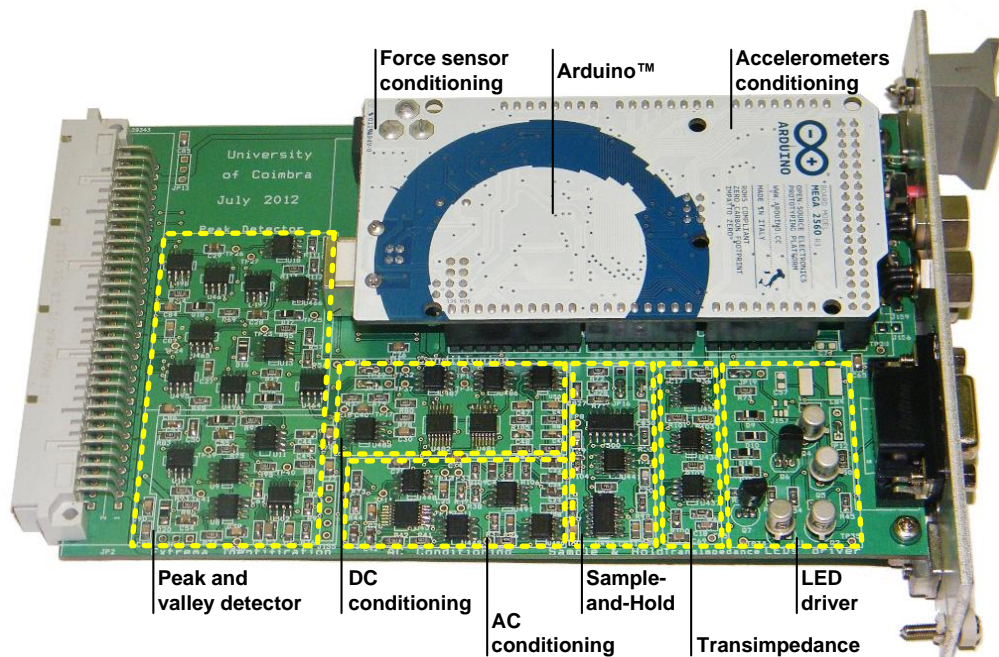
This new version already incorporated the force sensor and the accelerometers. Therefore, it had the FSR® conditioning module and the I<sup>2</sup>C communication module, required to convert the 5V logic level of the Arduino™ to the 3.3 V of the accelerometers. Firmware was also implemented to acquire these variables.

Peaks and valleys were still determined by software in an offline processing. However, it was desired that this could be done online. Nonetheless, constant comparison between the new sample and the stored peak and valley voltages was not desired to be done by firmware, given its huge processor time consume. Recall that there are 4 values to compare each two milliseconds. That means that in each second, 2000 comparisons are made but at the end just 4 points are needed, i.e. the processor waste is enormous.

Besides, the system employed a power supply module with the +5V and GND supplied by the Arduino™ USB cable, making it energetically autonomous. That fed the circuitry with +5V, GND and -5V required for integrated circuits (ICs) operation. The +3.3 V required by the accelerometers were directly supplied by the Arduino™ 3.3 V pin.

### 5.1.3 Version 3

The current version is the third large modification of the system. It is also implemented in a PCB board, but the new goal of implementing it in a multichannel system led to the incorporation of the Arduino™ in the PCB itself. The multichannel module is shown in Figure 5.1.



**Figure 5.1** – Current version of the system. The Arduino™ Mega 2560 is implemented in the board along with all the remaining hardware modules: LED driver, transimpedance, sample-and-hold, AC and DC signal conditioning, peak and valley detector, force sensor conditioning (below the Arduino™) and accelerometer conditioning (below the Arduino™).

This new version has some improvements relatively to the previous one. To begin, the new requirements in the analogue sampling channels, dictated that the Duemilanove should be upgraded to the Mega 2560. Besides, the MCU is implemented directly in the PCB and not interfaced with wires, as before.

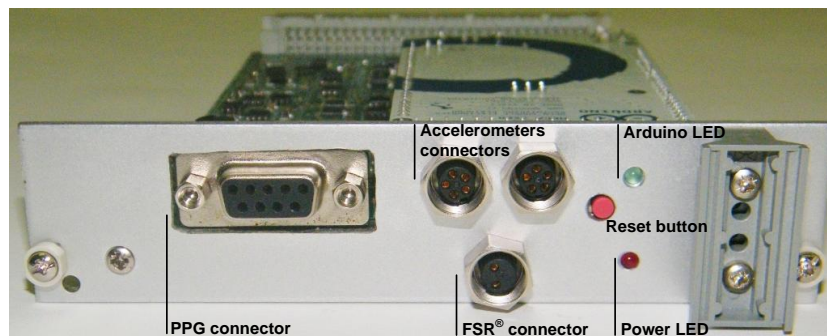
All but the signal conditioning modules are similar to the ones in the preceding version. In this version, it was clearly divided into two routes: AC and DC (in truth, AC and DC+AC). Both AC and DC conditioning are done with the support of digital potentiometers (digipots) in order to allow for online control of the system.

AC conditioning is conceptually the same as before, although the signal gain is given by digipots in order to be automatically adjustable to different subjects.

DC conditioning is accomplished by a DC level shift, which is accomplished by creating a constant voltage with a digipot and input it into an instrumentation amplifier. Then, a gain is set, also with a digipot. Again, this allow for subject adjustment and, more interestingly, for DC level and amplitude corrections during acquisition procedure. This allows for the system to adjust the signal to best fit in the ADC dynamic range.

In addition, the power supply module was no longer needed since the system is to be supplied by batteries that are common to all multichannel modules.

The system has now some connectors for external sensors, such as the PPG sensor, the two accelerometers and the force sensor (Figure 5.2). The first one is connected with a DB-9 and the others have circular connectors. The panel has also two LEDs, one that indicates power supply and the other indicates that Arduino™ is connected to the computer through the USB cable. A reset button for the Arduino™ is also present.



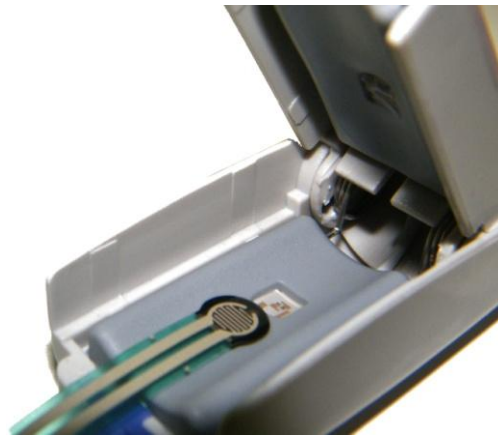
**Figure 5.2 – System front panel: external sensors connectors, indicating LEDs and reset button.**

The two accelerometers are attached to a wristlet and an armband, respectively, in order to maintain it fix in the subject forearm and arm respectively, but in an easy way to be placed and removed (Figure 5.3).



**Figure 5.3 – Accelerometers are attached to a wristlet and an armband and placed in a stable manner in the subject forearm and arm.**

The force sensor is attached to the PPG finger probe, over the inferior side and centred with the probe axis in order to be as normal to the applied force as possible (Figure 5.4).



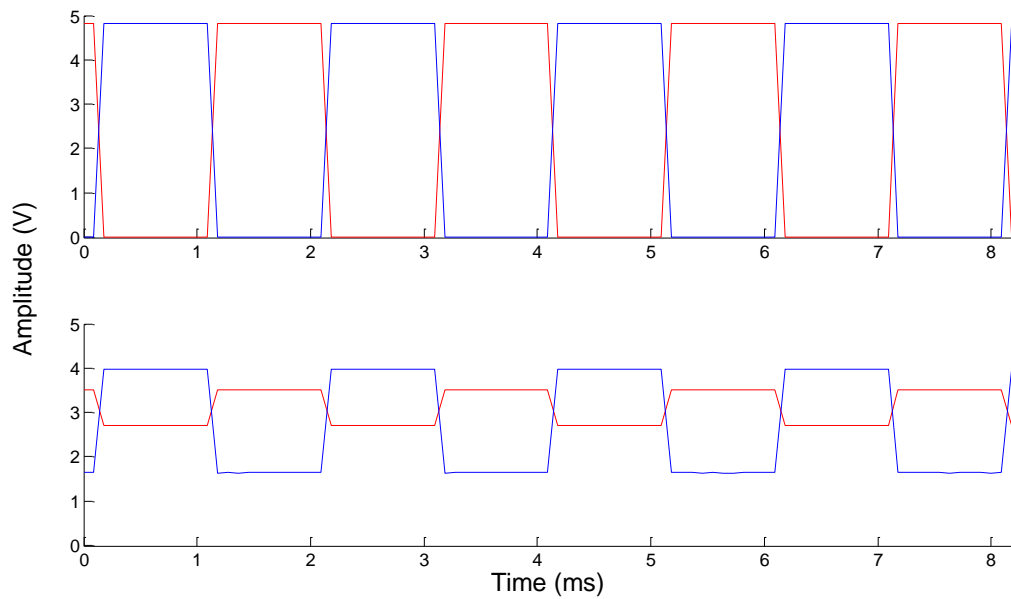
**Figure 5.4** –Force sensor is attached to the inferior side of the PPG probe centred with its axis.

This last version seems to be adequate to the purpose, which is reinforced by the results acquired with it and that will be presented in the following sections. First, results of each module will be presented. In these intermediate results, data was acquired with an acquisition platform from National Instruments™, the NI-USB 6210 since central modules are not connected to the Arduino™ ADC. Besides, some signals have very low amplitudes and the 16-bits of the NI device are more adequate to the purpose. Then, final PPG signal will be acquired with the Arduino™ to prove that the system is functional.

## 5.2 LED driver

The first module of the system is the LED driver. Its intent is to give controlled current to the LEDs in an alternated way, i.e. when one LED is ON, the other should be OFF. This is done in order to not mix both attenuation signals in the photodetection module. Figure 5.5 shows the signals generated by the MCU and supplied to the LEDs.

As expected, both digital output signals from the MCU are square waves with 5 V of amplitude and 2 milliseconds of period: 1 ms ON and 1 ms OFF. They are in opposite phase, as they should for sequential LED supply. The slope of the transition is exaggeratedly slow because of the low acquisition rate. In practice, the rising time is about is about 0.125 V/ns.



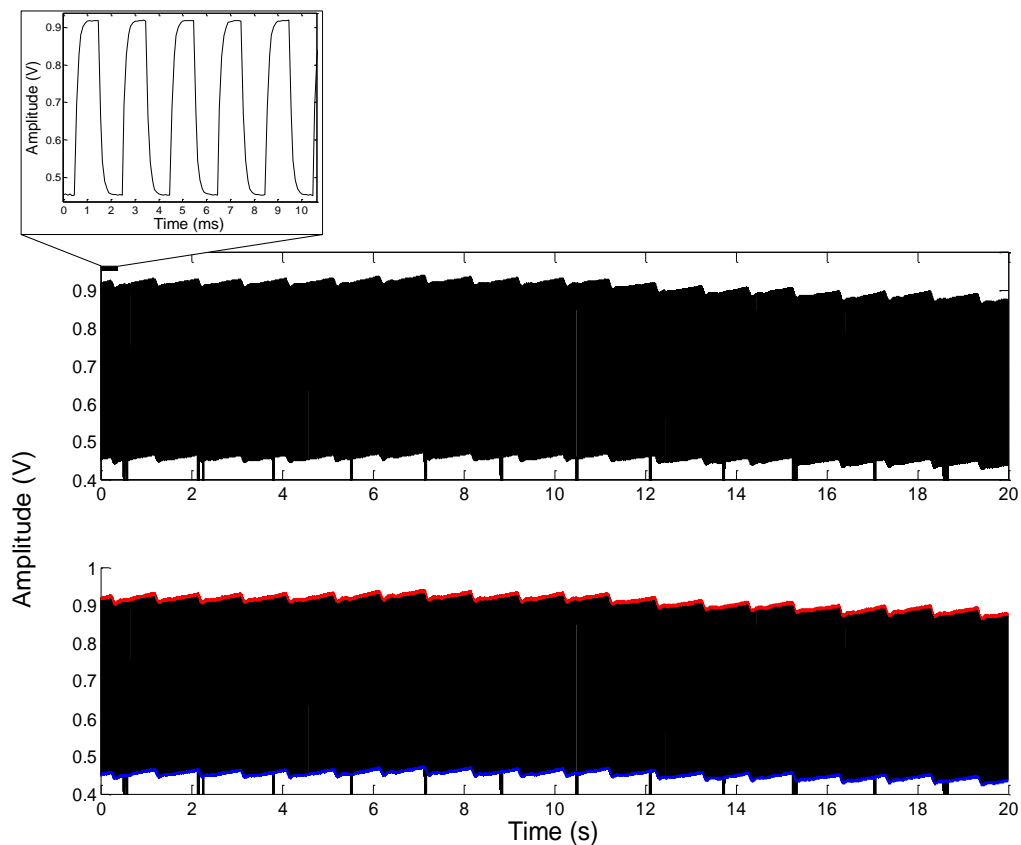
**Figure 5.5 – LED driver signals: MCU digital output (top) pins *LED\_power2* (shown in red) and *LED\_power1* (blue); and voltages across the LEDs terminals (bottom): red anode and infrared cathode (shown in red) and red cathode and infrared anode (blue). Signals were acquired with the NI-USB6210 at 11 kHz.**

Across the LEDs terminals, the voltage is also a square wave. Again, as expected, they are in opposite phase: for 1 ms red anode is higher than red cathode, meaning that the red LED is ON and the infrared is OFF; for the other 1 ms the positions are inverted, being the infrared LED ON and the red OFF. The voltage difference across both terminals is 1.86 V when the red LED is ON and 1.27V when the infrared LED is ON. These voltage differences are no more that the LEDs voltage drop, which is different for both components.

### 5.3 Photodetection and Sample-and-Hold

The following module is the photodetection and S/H module, which has the role of converting the PPG probe photodetector current to voltage and separate the signal into two channels: red and infrared. Figure 5.6 shows the voltage signals at the end of transimpedance and S/H modules.

The voltage at the transimpedance module is modulated. Both edges of the signal show an evident PPG waveform. However, as it can be seen in the figure detail, the signal is almost a square wave.



**Figure 5.6 – Transimpedance module and sample-and-hold: the voltage output of the transimpedance module, corresponding to the photodiode signal is visibly modulated (top). The modulation clearly matches each of the split channels (bottom): red channel (in red) and infrared channel (in blue).**

In fact, the signal stays at the lower edge when one LED is on and rises to the upper one when the other LED is switched ON. This is due to the LEDs intensity, which is different from both and to the efficiency of the photodetector, which changes with wavelength, as well as the attenuation of light in tissues that is also distinct.

Imposing both red and infrared channels over that signal, it is possible to see that the transimpedance signal edges perfectly match the split channels coming from the sample-and-hold module. The red channel is the upper envelope, while the infrared is the lower one.

When the signals are split by the S/H circuit the square modulation disappears and the PPG waveform becomes visible on both channels, as Figure 5.7 confirms. Nevertheless, it has very small amplitude and much noise. For these reasons, subsequent conditioning is needed, namely filtering and gain adjustment.



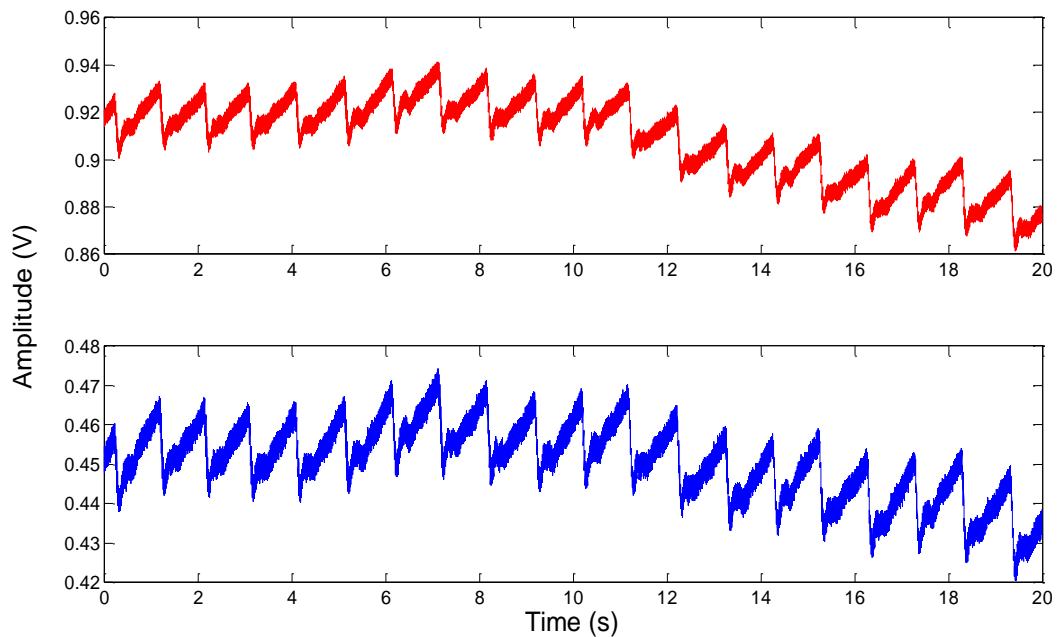


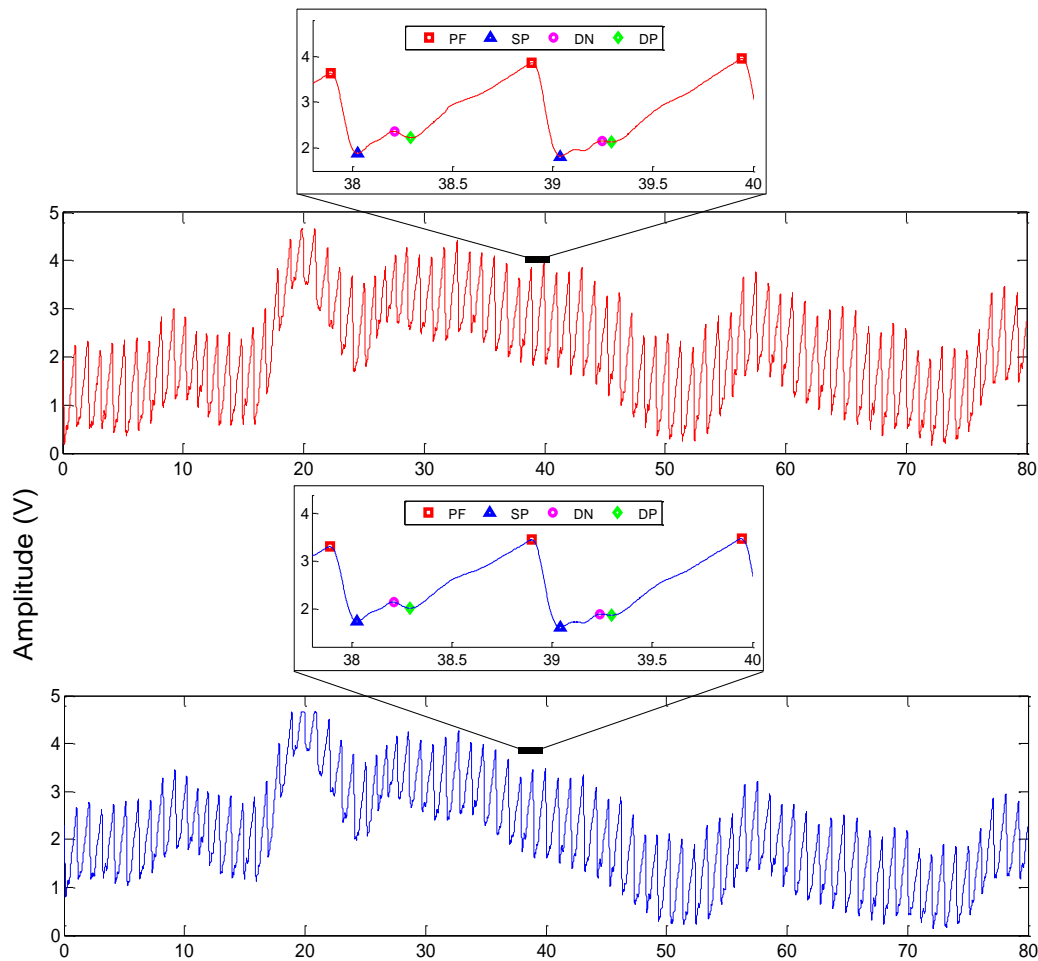
Figure 5.7 – Sample-and-Hold: the signal from the transimpedance module is split into two channels – red (top) and infrared (bottom).

## 5.4 Signal conditioning

As referred before (section 3.1.4), two different ways were developed for the signal conditioning. One consists of the DC+AC component, which is required for  $SpO_2$  computation, and the other is the AC component. Both components were sampled with the chosen 500 Hz of sampling rate and are presented below.

### 5.4.1 DC conditioning

As said before, the signal after the S/H module had small amplitude and large noise, so further filtering and amplification was needed. That is accomplished with the low-pass filter and instrumentation amplifier described in section 3.1.4. Signal was acquired after the instrumentation amplifier output and before the last inverting op-amp. This will make the sampled signal inverted when compared with the usual representation in commercial devices. The following inverting op-amp will invert the signal to be display in the conventional way. Figure 5.8 shows the acquired PPG signal for both channels.



**Figure 5.8 – PPG DC+AC component: red channel acquired with a gain of 20.4 and an offset voltage of 742 mV (top); infrared channel acquired with a gain of 28.8 and an offset voltage of 1.07 V (bottom). Both signals were acquired with the NI-USB 6210 with a sampling frequency of 500 Hz. On top of each channel, a detail of two pulses is shown with the PPG features: systolic peak (SP), dicrotic peak (DP), dicrotic notch (DN) and pulse foot (PF).**

The first thing to notice is the persistent change in voltage level. It is this change in the signal level that makes the PPG signal retrieval difficult. If the signal has high gain, it will rapidly fall outside the ADC range if no DC level shifting is done. If it has low gain to stay inside the ADC range, its resolution will be much lower and the SNR higher. That is why an automatic controller to account for these variations is crucial.

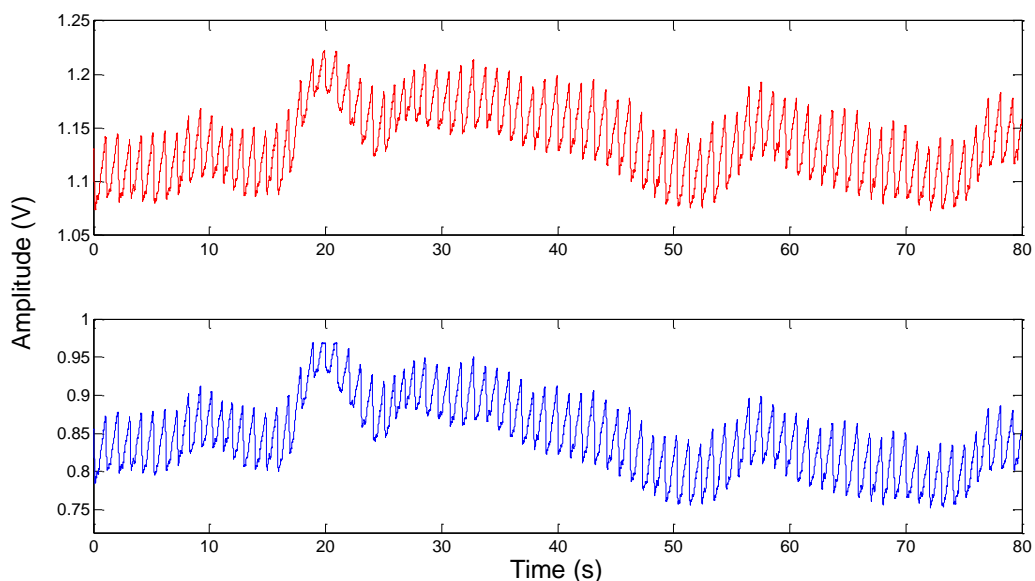
The level changes led to the temporary saturation of the infrared channel near the 20 seconds and the reduction in amplitude of both channels. However, that is not important for the current analysis.

To examine the PPG waveform, two pulses are detailed above each channel. As it can be seen, the main features such as the systolic peak, diastolic peak and valley and the pulse foot are well defined. Moreover, the general shape is consistent with the described in literature [33].

As exploited in section 2.2.3, these sampled signals must be reconstructed, i.e. the offset level and the amplification gain that are different for both channels must be backpropagated in order to retrieve a faithful signal for  $SpO_2$  computation. The offset voltage is retrieved from the digipot potentiometer wiper position according to equation (4.7) while the gain is computed from the digipot rheostat wiper (equation (4.8)) according to equation (3.10).

For the signal shown in Figure 5.8, these values are the following. For the red channel, the wiper position was 13, which means a 5.15 k $\Omega$  resistance, i.e. a gain of 20.4. For the offset level, the wiper position was 38, which means 742 mV. As far as the infrared channel is concerned, the gain was 28.8 (as the wiper position was 9, i.e. the resistance was 3.59 k $\Omega$ ) and the offset level was 1.07 V (wiper position of 55). Using these values, the signal before the AD623 can be reconstructed. The result is shown in Figure 5.9. Before that point in the circuit, the gains are the same for both channels, which means that they cancel out in the Ratio of Ratios ( $R_{OS}$ ) formula (equation (4.4)).

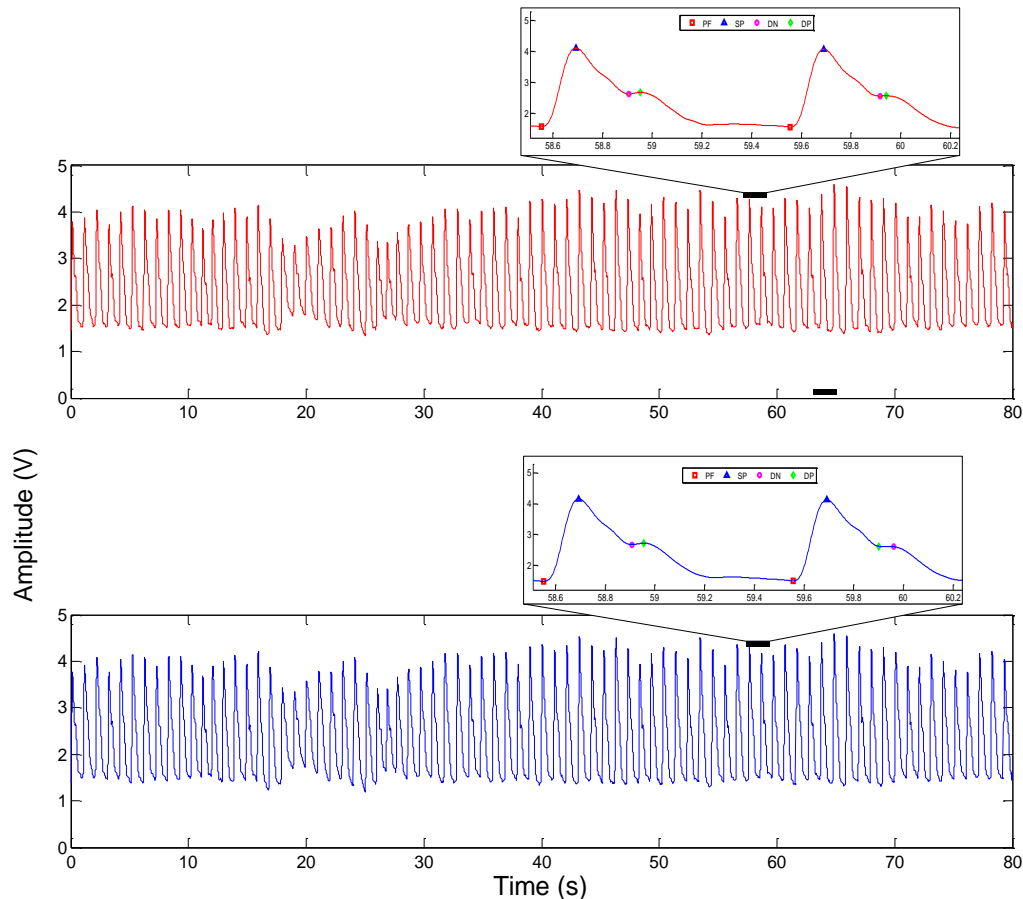
This signal can now be used to compute the  $SpO_2$ , as it will be seen in the section 5.6.



**Figure 5.9 – Reconstruction of the PPG DC+AC component of red channel (top) and infrared (bottom) from the digipot voltage and resistance values.**

## 5.4.2 AC conditioning

For the analysis of the AC conditioning, the AC component corresponding to the signal depicted in Figure 5.8 is shown in Figure 5.10.



**Figure 5.10 – PPG AC component of red channel (top) and infrared channel (bottom). Signals were acquired with the NI-USB 6210 at 500 Hz. On top of each channel, a detail of two pulses is shown with the PPG features: systolic peak (SP), dicrotic peak (DP), dicrotic notch (DN) and pulse foot (PF).**

Analysing the results, it becomes obvious that the AC component of the signal is much more stable than the DC+AC component. This is why this component is used for display and for the peak and valley detector. Even around 20 seconds, where the signal increased until saturation and then decreased (see Figure 5.8), the AC component stayed within the ADC dynamic range and with reasonable amplitude.

As far as the PPG waveform is concerned, the results are similar to the ones obtained for the DC+AC component. All the features are well identified in the signal. However, it seems that the dicrotic notch and wave are not so marked, possibly as a result of the AC filtering that slightly distorts the signal.

In the future, cutoff frequency or filter order may be adjusted in order to have a higher roll-off. This may prevent important components from being removed. Though, alterations are not very significant and the waveform is still accurate enough for display.

AC gain is not important since that component is used just for PPG display and is not used for computations.

## 5.5 Peak and valley detector

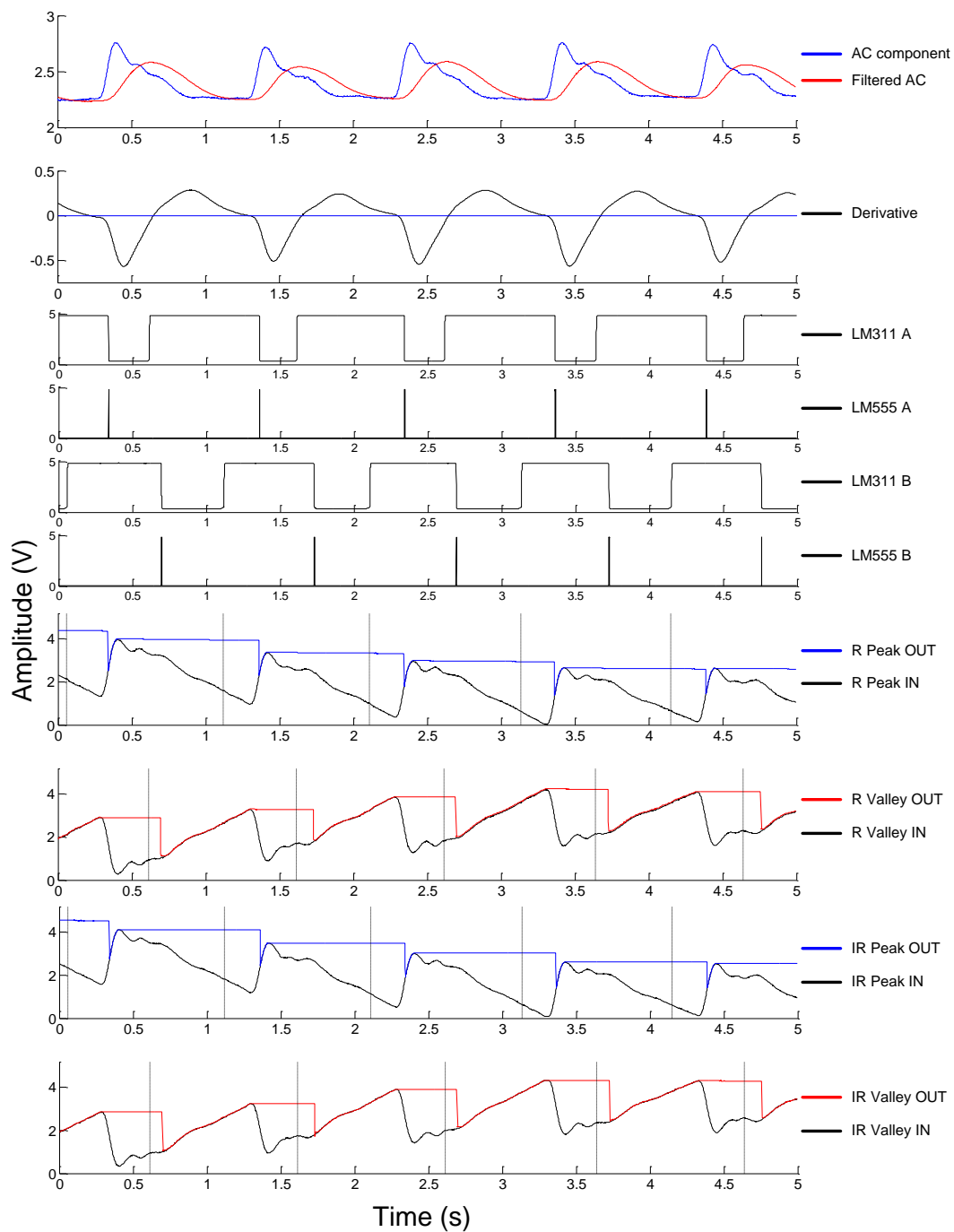
As referred before, the peak and valley detection by firmware is very heavy. So, hardware implementation was done to not overload the MCU (section 3.1.5). In this topic, the results of the peak detector will be analysed.

In order to realise if what is happening in intermediate points is synchronised, 5 seconds of signal are depicted in Figure 5.11. All intermediate steps of peak detector are shown.

First, the triggering signal, i.e. the AC component of the red channel, is depicted in row 1. In the same row, the filtering of this signal is also shown. Filtering was done in order to remove the higher components of the signal, as it is the case of the dicrotic peak and valley that would produce false reset and sampling signals. It is visible the effect of filtering in the trigger signal. First, it clearly removes the dicrotic peak, yielding only one peak per pulse, as it was desired. This way, it is assured that the dicrotic notch will not trigger any sampling or reset signal. Besides, it is also notorious the delay that the filtering imposes, which was  $0.224 \pm 0.015$  s for the acquired signal.

As the module needs the instants where the PPG signal inverts its tendency, the derivative of the filtered trigger is used and is shown in row 2. As seen, the derivative is negative when the filtered trigger increases and positive when it decreases. This is a consequence of the inverting topology of the hardware filtering. However, this has no practical effect on the module, since there are two comparators and both sides will be needed.

This derivative is used to trigger the two LM311 comparators. One of the comparators (row 3) is responsible for generating the interrupt for valley sampling by the MCU in the rising edge and also triggers the LM555 timer on the falling edge. This LM555 is used for peak reset and its output is represented in row 4. Likewise, the other comparator (row 5) is used to generate the peak sampling interrupt signal on the ascendant flank and trigger the LM555 (row 6) on descendant flank for valley reset.



**Figure 5.11 – Peak detector with intermediate steps. Raw AC component (in blue) and subsequent filtering (in red) used as triggering (row 1); derivative of the filtered trigger (row 2); LM311 comparator for valley sampling (row 3) and the corresponding LM555 for peak reset (row 4); LM311 comparator for peak sampling (row 5) and the respective LM555 valley reset (row 6); peak detection input (in black) and output (in blue) for red channel (row 7) and valley detection input (in black) and output (in red) for red channel (row 8); the same for infrared peak (row 9) and valley (row 10) detection. Dashed vertical lines show the instant when the interrupt signal for sampling is given to the MCU (descendant flank of the respective comparator output).**

As shown, the descendant flank of the comparator signal does trigger the respective timer one-shot pulse. These one-shot pulses are used to temporarily connect the capacitor of the peak detector to ground and thus discharge it, making it ready to sample the subsequent pulse. The pulses of the first LM555 (row 4) are used to reset the peaks signal, as it can be seen by the abrupt decrease in voltage of the peak detector output (shown in blue in rows 7 and 9). The pulses of the second LM555 (row 6) are used to reset the valleys signal, as they are synchronized with the down step of the valley detector (red in rows 8 and 10).

Both reset signals occur before the following peak or valley, as it is required. In the case of the valleys, the reset signal is given approximately in the middle of the pulse, which means that the module has plenty of time to sample the new valley. Conversely, in the case of peaks, the reset signal is given almost at the time of the new peak, letting less time for its sampling. This is a consequence of the asymmetry of the PPG signal, or rather, of the cardiac cycle. It has a fast pressure increase when the chambers close and a slower decrease in pressure as the blood propagates through the arterial tree [22]. However, this time interval seems to be adequate, at least for the acquired signal which had a heart rate (*HR*) of 58 bpm. Simulations done and shown in Appendix B reinforce the fact that the system is reliable. Though, some adjustments can be done. Namely, if the comparator 1 is used for both peak and valley sampling instead of just valley sampling, comparator 2 rising edge could be used to reset the peak detector. Nevertheless, these alterations must be studied if further improvements are required, namely with different *HR* and pulses of different subjects.

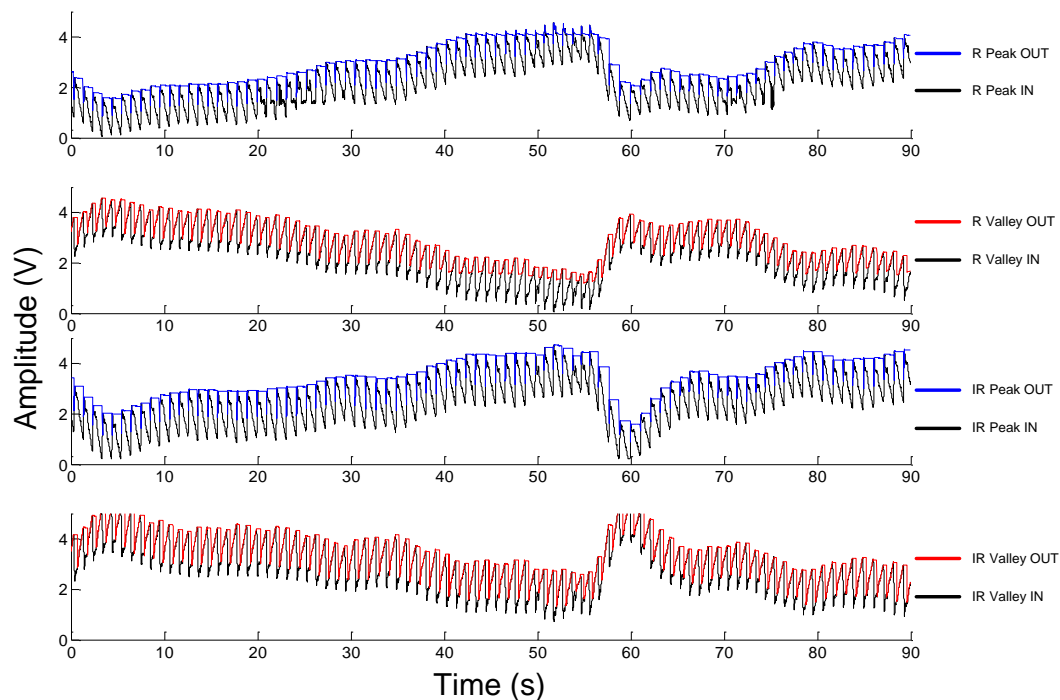
The 4 outputs of the peak detector are shown below. The original red channel signal is shown in black in row 7 and the peak detector output is shown in blue. It is visible the reset signal immediately prior to a new peak, as the voltage rapidly decreases to a low value. Then, the voltage increases with the PPG pulse until a maximum has been reached. From there, the PPG signal decreases but the peak detector output is held to wait for signal sampling. The sampling routine is run by an interrupt signal coming from the rising edge of the comparator. That instant is shown as a dashed vertical line. When a new reset signal is given, the voltage decreases again in order to sample the new peak. The same is applied for infrared channel in row 9).

The inverted PPG signal is depicted in black in row 8 and the respective valley detector output is shown in red. The procedure is similar to the one referred above: after a reset signal occurs the voltage increases with PPG inverted signal and is held at its maximum

for sampling (dashed line) before a reset signal is sent again. The same is done for infrared signal (row 10).

It is notorious a slow decrease in peak and valley voltage steps. This is due to the capacitor discharge and cannot be eliminated. It is possible to adjust the capacitor to decrease that drop (increasing capacitance), but that will lead to slower slew rate of the peak and valley detectors. The decrease in voltage is around 4 mV but depends on peak or valley voltage and, of course, on pulse duration. For the acquired signals, the drop in 4 mV is consistent with the value predicted in section 3.1.5.1. This value leads to a change of less than 0.1 % in  $SpO_2$ , but again this change is dependent both on signal DC level and amplitude. Therefore, if an improvement of this system is required, the capacitor value can be altered and the sampling signals can be given earlier.

In order to realise that the system consistently detects the peaks and valleys, an acquisition of 100 seconds is shown in Figure 5.12. As it can be noticed, the signal does not have a constant level. In fact, it significantly changes over time. However, the system always follows the peaks and valleys, proving that it works well and with the desired synchronisations. Therefore, it is useful for peak and valley detection by hardware, eliminating the need to be done by firmware at the cost of processor time consume.



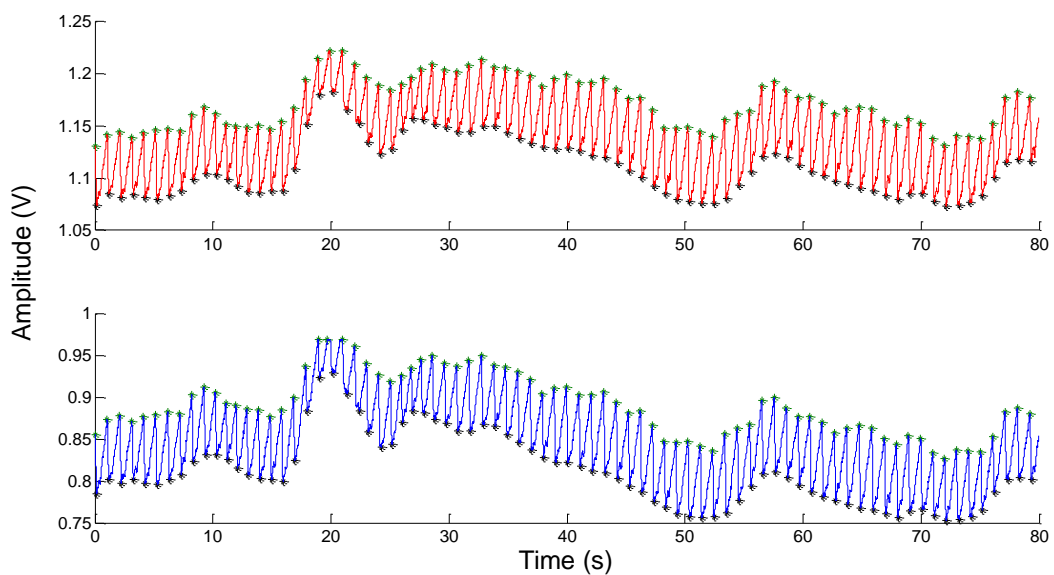
**Figure 5.12 – Peak and valley detection. Peak detection input (in black) and output (in blue) are shown for the red (row 1) and infrared channel (row 3); valley detection input (in black) and output (in red) are also shown for red (row 2) and infrared (row 4) channels.**



## 5.6 Pulse oximetry

The  $SpO_2$  computation relies on the reconstruction of the DC+AC component peaks and valleys. Those are to be sampled from the peak and valley detector described above. For now, and as a proof of concept,  $SpO_2$  will be computed from the reconstructed signal in Figure 5.9. This is due to the lack of online DC level adjustment and to the fact that the dynamic range of the Arduino™ ADC is not the best choice, having less SNR. Thus, maxima and minima of the signal are computed by the MATLAB function *peakdet* and the results are shown in Figure 5.13.

Attention must be paid to the fact that the signal is inverted, i.e. the peaks (systolic peak) are marked with black stars, while the signal valley (the end of the cycle) is shown with green stars.



**Figure 5.13 – PPG signal peaks (black stars) and valleys (green stars) used for  $SpO_2$  computation.**

This peak and valley voltages can now be used to compute the  $R_{OS}$  and consequently  $SpO_2$  according to equations (4.4) and (4.5), respectively. The resulting values are depicted in Figure 5.14.

The first thing that comes with the analysis of the  $SpO_2$  evolution is the peak around 20 seconds, which corresponds to a significant decrease in  $SpO_2$  – to about 80%. This is due to the saturation that occurred in the infrared channel and that was previously referred.

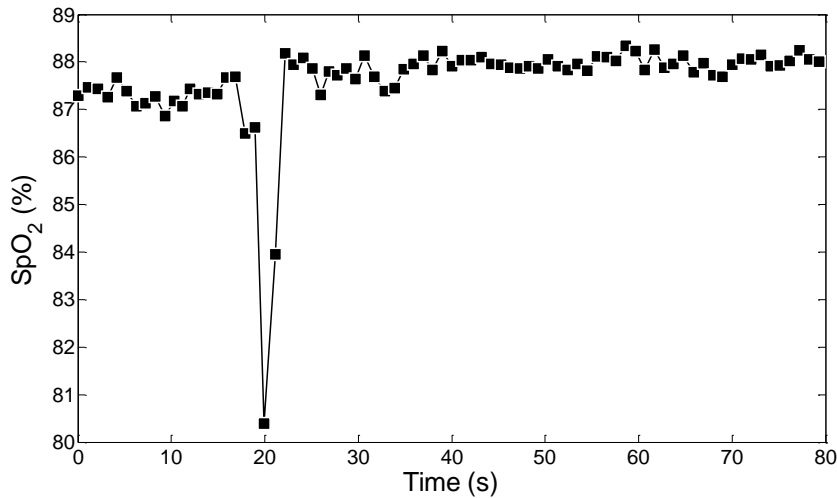


Figure 5.14 – SpO<sub>2</sub> values computed from the peak and valley values of the PPG signal.

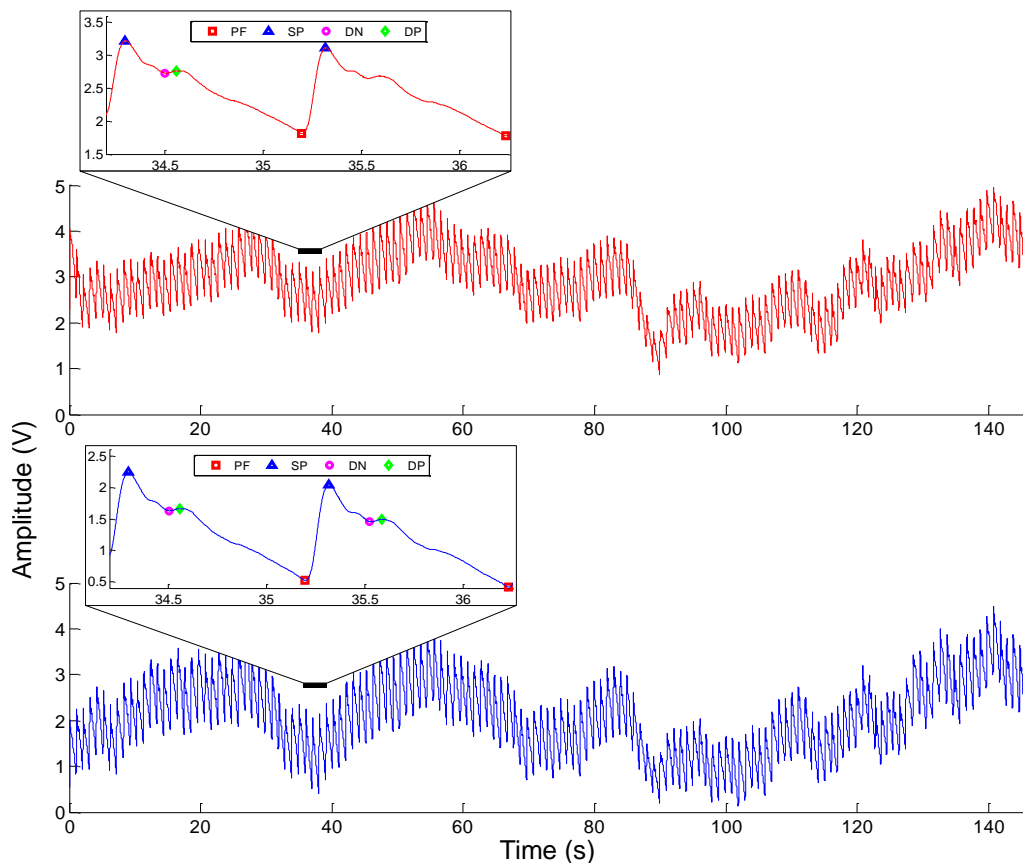
Recalling that the DC+AC component is inverted, it is clear that the valley value of infrared channel is increased (since the signal did not reach its minimum value as it saturated). Therefore, its amplitude is assumed as lower than its actual value and that leads to a smaller  $R_{OS}$  according to equation (4.4). From Figure 2.6, it is notorious that this fact leads to a decrease in  $SpO_2$ . Therefore, 4 pulses around 20 seconds will be discarded. That results in a  $SpO_2$  value of  $87.8 \pm 0.3$  % for the remaining 75 pulses. This leads to two main conclusions: first, the value is lower than the expected and second, it is fairly stable.

The low  $SpO_2$  value may be explained by the fact that no sensor calibration was performed. Actually, as referred before (section 2.2.3), when light scatter and nonlinear absorption are considered,  $SpO_2$  tends to increase. Moreover, the oscillations in the signal may somehow affect the  $SpO_2$  values since signal amplitude is wrongly overestimated as the signal decreases and underestimated when the signal increases. If these miscomputations are not similar in both channels,  $SpO_2$  values can be distorted. Furthermore, the results were taken just from one subject. Acquisitions with more subjects must be done to realise if this deviation is systematic.

Although the value may be inaccurate, it remains quite constant, even with large fluctuations of the DC+AC component (which can exceed the 100% of signal amplitude), provided there is no signal saturation. This seems to mean that the system has good precision and repeatability, although still lacking accuracy.

## 5.7 Arduino™ acquisition

All the results shown above were acquired with the NI-USB 6210 because intermediate points were not connected to the Arduino™ ADC. However, the system is functional and both AC and DC+AC components can be acquired with the MCU. Figure 5.15 shows 25 seconds of the DC+AC PPG component acquisition done by the Arduino™ with a sampling frequency of 500Hz.



**Figure 5.15 – PPG DC+AC component for red channel (top) and infrared channel (bottom). Signals were acquired with the Arduino™ with a sampling frequency of 500 Hz. Many oscillations in the level voltage can be observed.**

As it can be seen, the signal has some oscillations in the DC level. That confirms the early statement and clearly demonstrates the need for an efficient controller to regulate offset level and gain. Since that controller is not developed yet, an ideal gain was not set. Therefore, for the signal to not saturate too often, gain was reduced. With this, a significant part of the ADC range is unused and the SNR is lower.

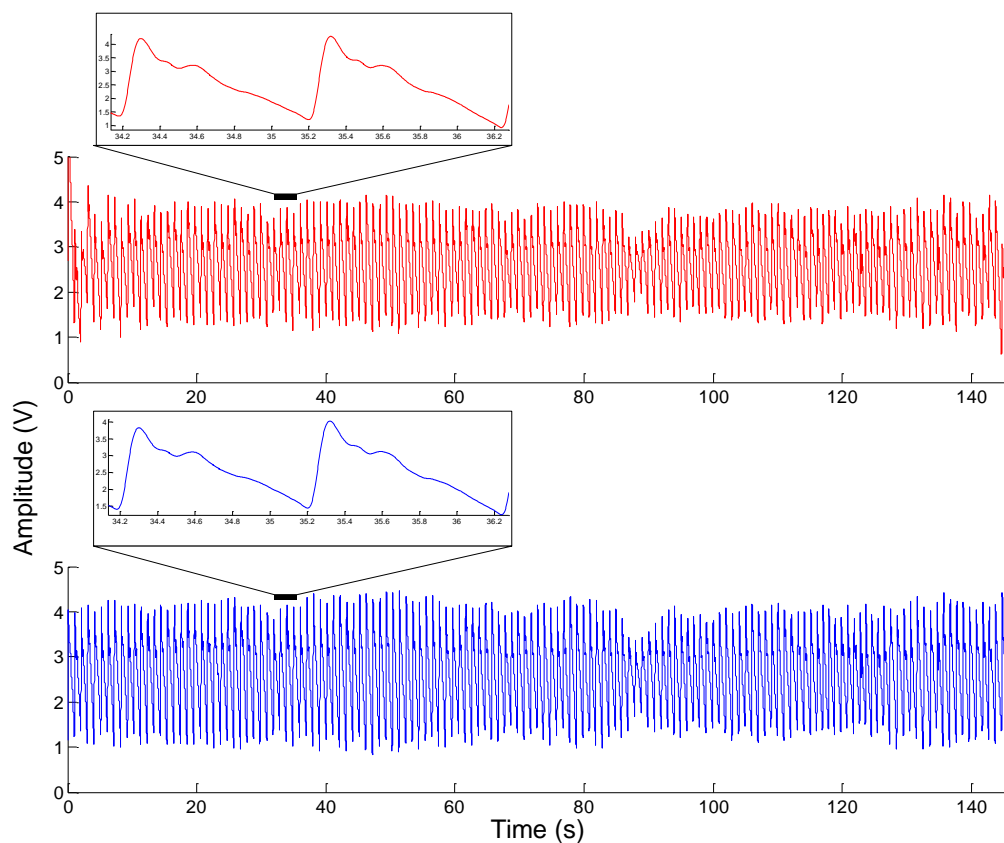
Analysing each individual pulse, the signal quality is unexpectedly good. Each pulse has an amplitude around 1 V, which is just one fifth of the total range of the ADC. However,

the waveform is “clean” with no visible noise and with all features present. This seems to indicate that this amplitude may be reasonable and that the controller may not have to be very aggressive. If it had to adapt the signal conditioning too often, the peak and valley voltages would be compromised since the DC level would change too much times at the middle of a cardiac pulse, thus invalidating the amplitude results.

In part of the pulses, a second peak is observed for both the Arduino™ and NI acquisitions. That was not expected and can be owed to two factors. First, it can be due to some reflection at the patient finger arterial ramifications. The second explanation may be hand or finger tremor, which produces alterations like these in the signal [33]. To confirm this, signals need to be acquired for more subjects.

As far as the  $SpO_2$  is concerned, the computed value was  $91.5 \pm 1.1$  %, which means a better result than the previously computed, but still away from the typical healthy value (97% [47]). The lack of calibration may account for this deviation.

The Arduino™ also samples the AC component. Figure 5.16 depicts the AC component corresponding to the PPG signal of Figure 5.15.

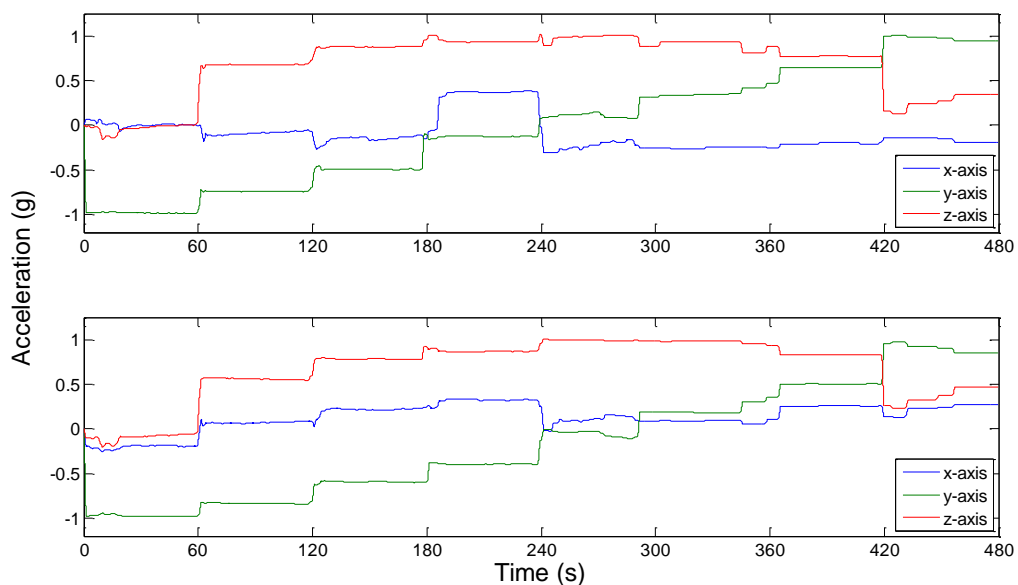


**Figure 5.16 – PPG AC component for red channel (top) and infrared channel (bottom). Signals were acquired with the Arduino™ at 500 Hz. The signal is much more stable than the DC+AC component.**

Again, it can be noticed that this component is much more stable than the DC+AC signal. That corroborates the idea of using this component for PPG display. No gain change is usually required, which allows for a better exploitation of the ADC dynamic range. Consequently, the SNR is increased, as the quantization error is less significant.

## 5.8 Height measurement

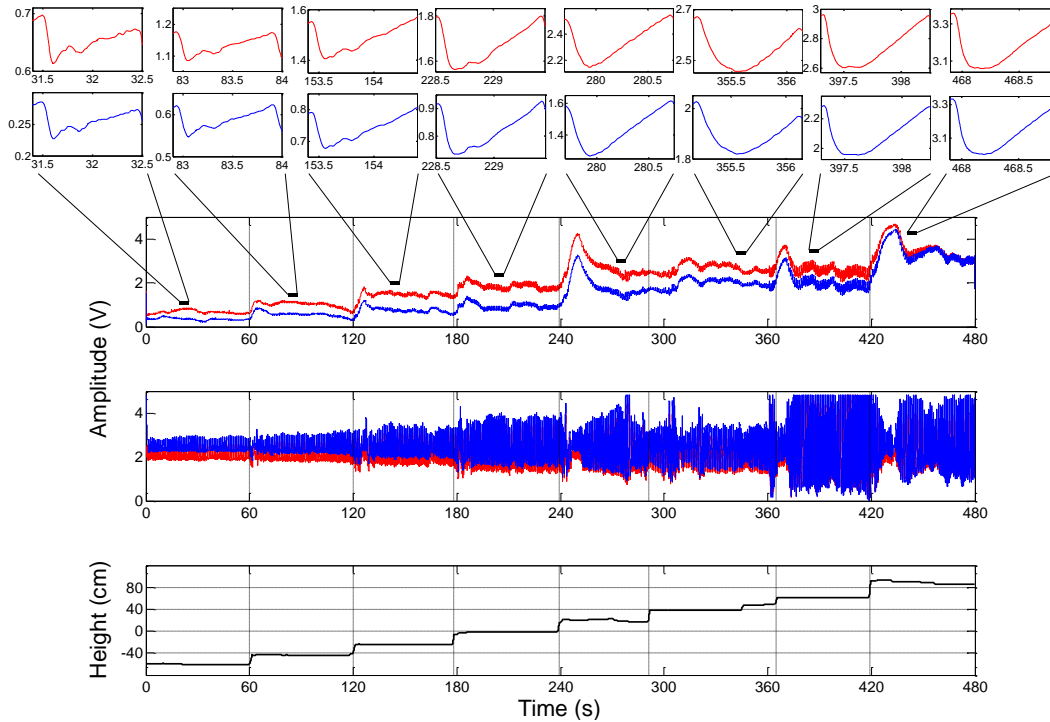
To test the height measurement module, acquisitions of the PPG signals were done with varying finger heights relatively to the heart. The experimental protocol started with the subject holding his arm completely pointed downwards and then, increases of 20 cm were performed each 1 minute, up to the arm was completely vertical pointing upwards. The tested heights ranged from  $-0.6\text{m}$  (60 cm below the heart) to  $+0.8\text{m}$  (80 cm above the heart). The accelerometers output is depicted in Figure 5.17.



**Figure 5.17 – Accelerometers 3-axis outputs for the forearm (top) and arm (bottom). The Y axis (shown in green) is aligned with the arm and forearm axis and is the best measure for height estimation.**

As it shows, the y-axis of accelerometer 1 is aligned with the arm axis while the y-axis of accelerometer 2 is aligned with the forearm axis. Therefore, these are the best axis to compute the height, since this is the one that define the increase or decrease in height when the subject moves the arm. Analysing Figure 5.17, it is possible to distinguish the different height levels as the y-axis output increases from  $-1\text{ g}$  (vertical pointing downwards) to  $+1\text{ g}$  (vertical pointing upwards). Both x- and z-axis are related with arm rotation and change as the subject rotated the arm when changing the height level.

To realise the alterations in the PPG signal as the finger height changes, both DC+AC and AC components were acquired during the procedure. Results are shown in Figure 5.18.



**Figure 5.18 – PPG signal (top and middle) with different finger heights (bottom). Red channel (shown in red) and infrared channel (shown in blue) are illustrated both for the DC+AC (top) and AC (middle) components. Vertical dashed lines show the instants when the height was increased. On top of DC+AC component, one pulse is depicted for each height level.**

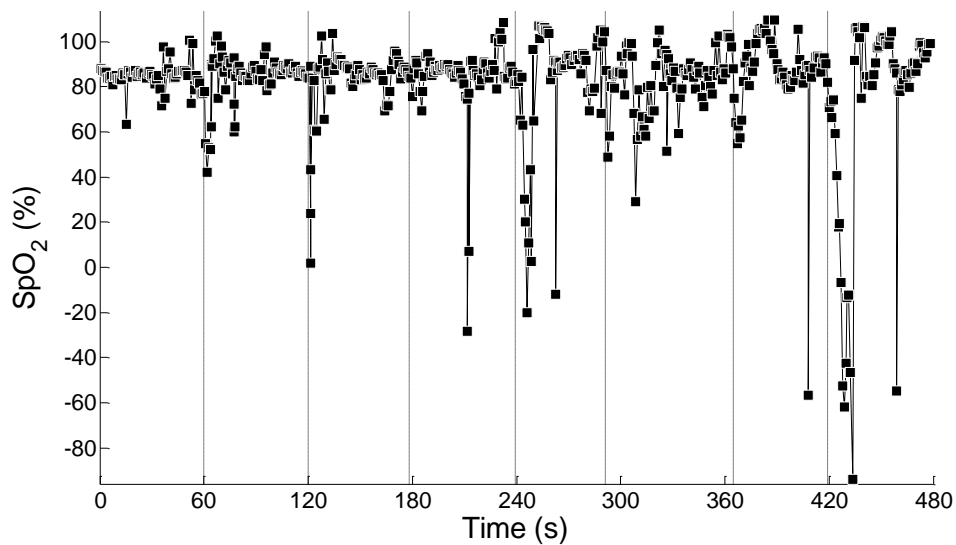
Looking at Figure 5.18, it is notorious that the PPG signal significantly varies with finger height. The most notorious alteration is the DC level, which rises as the height increases. This can be explained, at least partially, by the decrease in the venous blood that occurs when the finger is lifted. In fact, the lower the finger, the more blood tends to accumulate in the veins, as well as in the arteries as non-pulsatile blood. When the arm is raised, the hydrostatic pressure decreases and less blood accumulates. This way, the light is less attenuated and the signal is stronger.

Furthermore, it is possible to see that the signal amplitude also changes. This can be explained by the transmural pressure ( $P_{TM}$ ), as seen in section 2.3. The change of height influences the hydrostatic pressure and with that, the  $P_{TM}$ . PPG amplitude is maximal when  $P_{TM}$  is zero and decreases when it leaves that region. The maximum amplitude was reached at +60 cm of finger height. Although, it is not possible to affirm that this is the

point of zero  $P_{TM}$ , only, that probably, it is the closest one. However, that is not the goal of this work.

As expected, this change in amplitude is also seen in the AC component. Besides, the PPG waveform also changes its morphology, namely being the dicrotic notch less prominent. When the arm is totally lifted, this feature is almost imperceptible.

To realise the effect of height changes in  $SpO_2$ , its value for all pulses is depicted in Figure 5.19.



**Figure 5.19** –  $SpO_2$  values computed from the signal with variations in finger height shown in Figure 5.18. Dashed lines show the instants of height change. Substantial variations are seen, especially after the height change and before signal stabilization in that step.

As it can be seen, considerable fluctuations in  $SpO_2$  occur, especially after a step in height is changed and the signal has not established yet. It is even possible to get values of -60% of  $SpO_2$ , which clearly do not have any clinical significance. The value of  $SpO_2$  for all the 522 detected pulses is  $81.6 \pm 24\%$ , showing a low accuracy and a low precision. However, if pulses below 60% are assumed as miscomputed due to height change and discarded, the results improve to  $87.3 \pm 9\%$ . This shows a value very consistent with the one obtained for static height (Figure 5.14) although with comprehensive 30 times more variations.

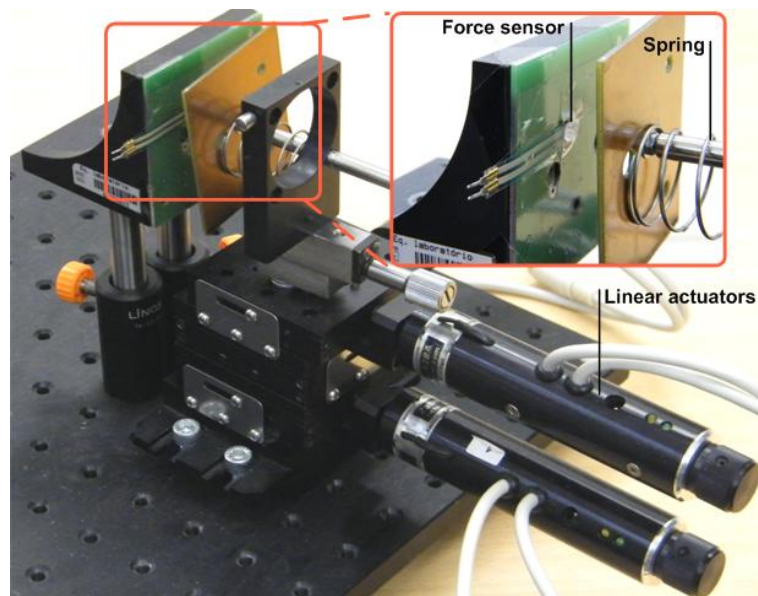
This means that height measurement is important to understand the values of  $SpO_2$ , since these values may be erroneous after a considerable change in height. This makes the accelerometers useful for the pulse oximetry system, in addition to their importance for an eventual blood pressure monitoring (as seen in section 2.3).

## 5.9 Contact force measurement

The force sensor output strongly depends on the configuration and on the gain resistor. Moreover, part calibration is required if accuracy is needed [50], as it is the case of this system. Therefore, an experimental setup was developed to calibrate the sensor. Subsequently, a curve fitting was done in order to obtain a transfer function. That curve is stored in the MCU and used afterwards to compute the contact force. This section will overview this calibration and curve fitting, and will, at the end, present results related with the PPG signal evolution when contact force is altered.

### 5.9.1 Calibration setup

To test the repeatability of the sensor, a calibration setup was developed, using a computer controlled linear actuator. The actuator was a *Zaber T-LA 28A*, a linear actuator with  $0.1\mu\text{m}$  resolution and 60 mm of maximum travelling distance. The actuator was placed over a linear stage. In order to obtain a force instead of a displacement, a spring with a known force constant was coupled to the end of the actuator. The other end of the spring was in touch with the FSR<sup>®</sup>, which was rigidly attached to a fixed platform. The calibration setup is shown in Figure 5.20.

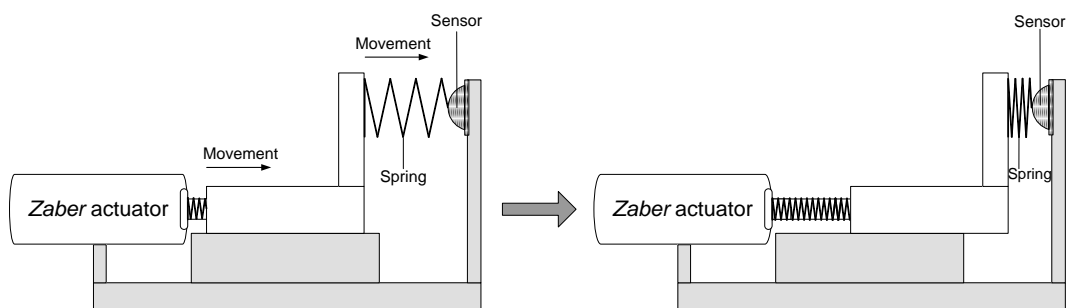


**Figure 5.20 – FSR<sup>®</sup> dynamic calibration setup using the *Zaber T-LA 28A* linear actuator. The FSR<sup>®</sup> is fixed and the actuator pushes the spring against it, producing a determined force according to the displacement.**



Via the RS 232 port, a command is sent by the computer to the actuator and the screw is turned to reach the desired position. As the linear actuator travels over the stage, the spring is compressed between the actuator and the FSR<sup>®</sup>, as Figure 5.21 shows. If the spring is compressed within its elastic limit, it is possible to compute the applied force by the Hook's law given by equation (5.1).

$$F = -k x \quad (5.1)$$



**Figure 5.21 – Schematic of the FSR<sup>®</sup> dynamic calibration experimental setup. The linear actuator moves and compresses the spring against the FSR<sup>®</sup>. As the displacement and force constant are known, the applied force can be calculated.**

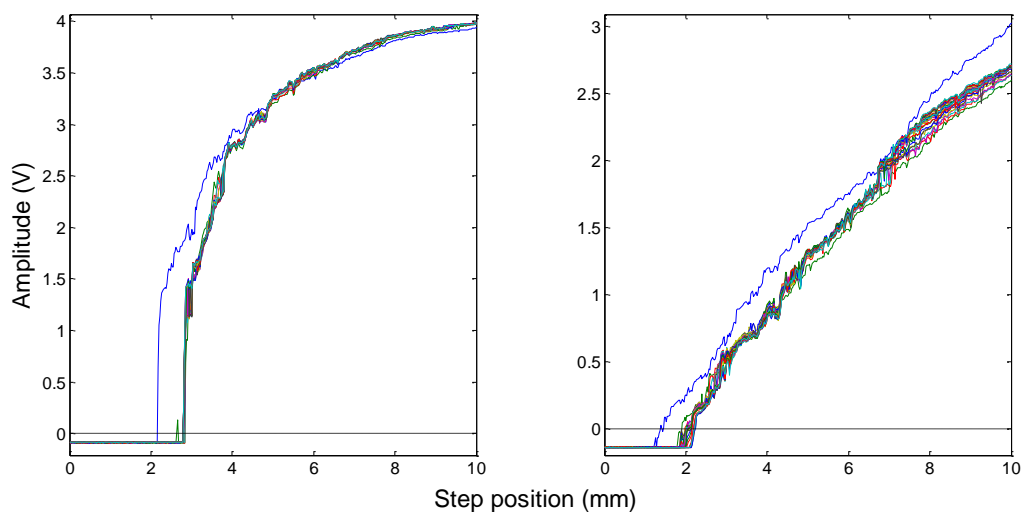
In order to obtain a better distribution of the applied force, a rigid coat with the diameter of the sensing area was added over the sensor [48]. That coat can be seen in Figure 5.20.

As mentioned before (and seen in Figure 2.11), the FSR<sup>®</sup> has a turn-on force, i.e. an activation threshold below which the sensor does not respond. Since that threshold is different from part to part, it must be determined. However, the previous experimental setup does not allow for its accurate determination. This is due to the difficulty of determining the exact point at which the actuator contacts with the sensor.

In order to determine that threshold, another experimental setup was developed. The sensor was placed horizontally and small spheres of lead (with approximately 0.1 g) were placed over its active area. Thus, knowing the weight of the spheres (measured with a weighting scale), one can compute the force needed to activate the sensor. As before, a small coating was placed over the active area to have a uniform application of the force over that entire region.

## 5.9.2 Calibration curve

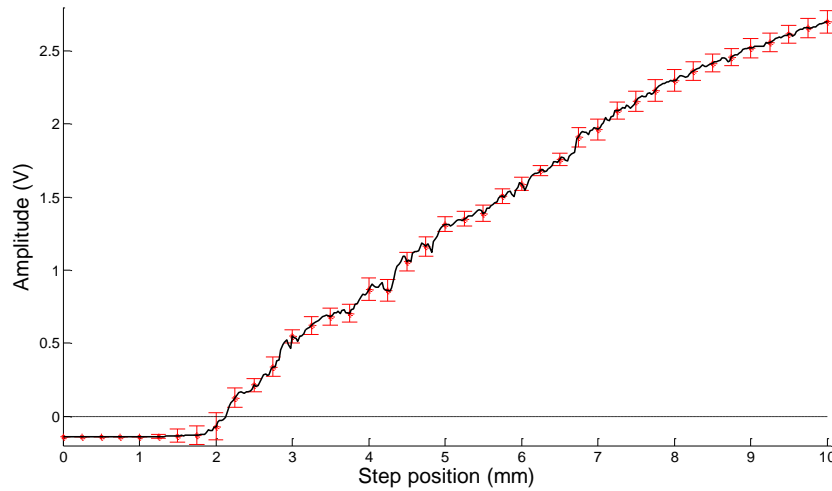
To obtain a calibration curve, the FSR<sup>®</sup> was tested with several values of resistor  $R_G$  for each configuration. For each resistor, 25 repeated tests were performed. Each test consisted of 400 sequential steps of the linear actuator with 25 $\mu$ m each. This totalized 10 mm of travelled distance, i.e. of spring compression. Figure 5.22 shows two of those tests, one for the voltage divider conditioning, and the other for the current-to-voltage converter.



**Figure 5.22 – FSR<sup>®</sup> output voltage of the calibration procedure for the voltage divider configuration with again resistance  $R_G=56\text{ k}\Omega$  (left) and with the current-to-voltage converter with a gain resistance  $R_G = 5.6\text{ k}\Omega$  (right). Values below zero mean that no contact with the sensor was made.**

To test the repeatability of the FSR<sup>®</sup> response and the conditioning circuitry, the responses of the 25 trials were compared and a mean and a standard deviation for each displacement were computed. Figure 5.23 shows an example of that computation. As it can be seen, in general, the responses are consistent, although the accuracy of the sensor is known to be low.

Table 5.1 shows the maximum standard deviation of all the 25 trials and all the 400 different displacements. As different values of  $R_G$  may or may not saturate the response, i.e., may or may not reach approximately 5 volts, to obtain a more truthful value, the FSR<sup>®</sup> response was normalized between 0 and 1.



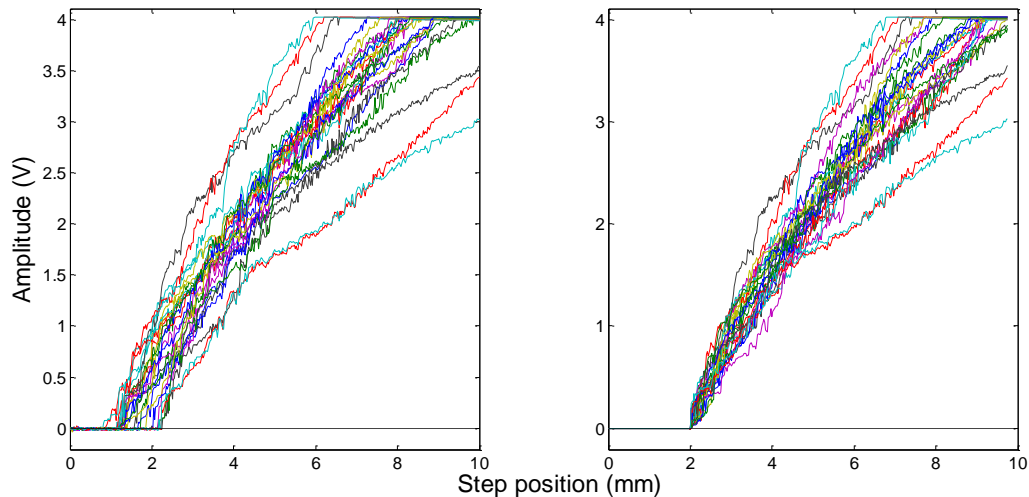
**Figure 5.23** – Mean (black line) and standard deviation (red error bars) for one of the experimental tests (current-to-voltage converter with a gain resistance  $R_G = 5.6 \text{ k}\Omega$ ) with the actuator tip starting always in the same position. Values below zero mean that no contact with the sensor was made.

As the maximum standard deviation means the biggest deviation of a specific displacement from the mean curve, this was the criterion to choose the best configuration. It is worth to notice from Table 5.1 that 4 of the 6 best results (including the first one) are for the current-to-voltage converter and that 5 of the 7 worst results (including the two worst) are for the voltage divider. Therefore, it can be concluded that the current-to-voltage converter is the best configuration in terms of repeatability. This is in agreement with the information on the sensor datasheet [50]. A further improvement can employ better conditioning circuits, as the one referred in [83].

**Table 5.1** – Maximum standard deviation of the 25 trials with 400 displacement points for different values of  $R_G$ . Results for both configurations are shown. The FSR<sup>®</sup> response was previously normalized.

Voltage divider		Current-to-voltage converter	
$R_G (k\Omega)$	$\max \sigma_i$	$R_G (k\Omega)$	$\max \sigma_i$
2.2	0.0453	2.2	0.0326
3.3	0.0420	3.3	0.0354
5.6	0.0418	5.6	0.0416
8.2	0.0427	8.2	0.0327
10	0.0504	10	0.0466
16	0.0365	–	–
33	0.0347	–	–
56	0.0573	–	–

To test the reproducibility of the FSR<sup>®</sup> response, 25 trials were made with different initial positions, i.e. with the actuator tip acting in different points of the FSR<sup>®</sup> sensing area. Some of these positions were purposely and exaggeratedly misaligned to simulate real incorrect positioning by the patient. Figure 5.24 shows the results of this calibration.



**Figure 5.24 – FSR<sup>®</sup> response for different misaligned contact positions: the raw results (left) show a large discrepancy; when the activation linear is superimposed for all trials (in 2 mm of step position), better results are achieved (right). The current-to-voltage converter with  $R_G = 7.67 \text{ k}\Omega$  was used.**

As it can be seen, the 25 trials lead to very discrepant results. However, the initial position is different from trial to trial and so is the displacement for which the first contact with the sensor is made. That can explain part of that discrepancy. Therefore, it is reasonable to shift all the results to the same original position, e.g. for 2 mm. Somehow, this simulates the condition of having the contact point at the same displacement for all the 25 trials. The resultant responses are shown in the right side of Figure 5.24. A much less dispersion of the results is obtained.

Even though, it is notorious that the point of application is important and can influence the output of the sensor. In fact, as Figure 5.25 shows, a much greater dispersion is obtained for this case when compared with the one for the trials with the same initial position (Figure 5.23). This is due to the fact that the initial position is not centred with the active area for all trials. Thus, it results in a shearing component of the applied force to which the sensor does not respond well [83]. This misalignment was sometimes exaggerated but aims to simulate real conditions where the subject finger may not be centred.

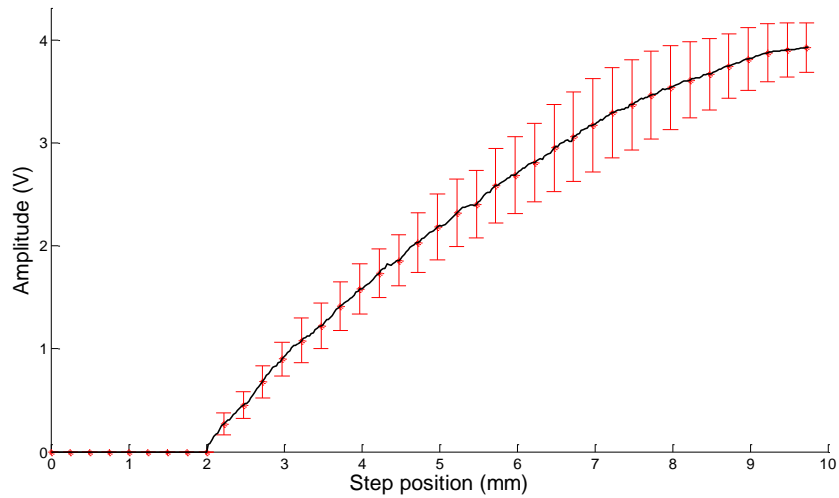


Figure 5.25 – Mean (black line) and standard deviation (red error bars) for one of the experimental tests (current-to-voltage converter with a gain resistance  $R_G = 7.67 \text{ k}\Omega$ ) with the actuator tip at different initial positions.

Thus, the sensor must be placed on the PPG sensor case in a way that the applied force is not likely to be other than normal to its active surface.

In order to obtain a value for the activation threshold, the signal of the sensor was measured when varying the weight placed upon it. The results for 10 trials are shown in Figure 5.26.

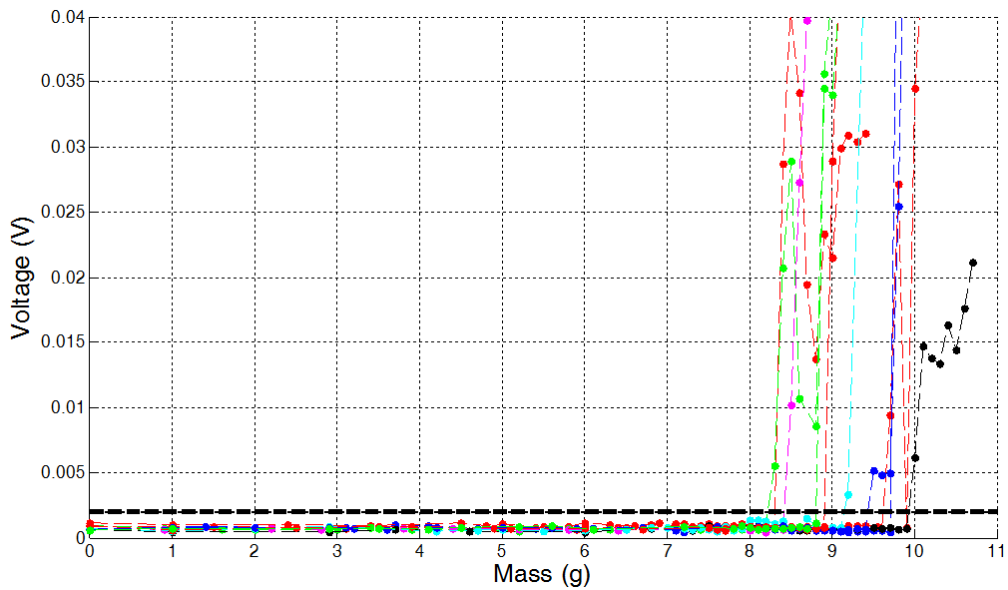


Figure 5.26 – Calibration results for FSR<sup>®</sup> turn-on threshold. The dashed line shows the voltage threshold defined by visual inspection of the results.

The dashed line shows the voltage threshold, which was defined by visual inspection. As it can be seen, below that threshold the voltage is similar and almost zero for all mass values. However, when the output crosses the threshold, it significantly differs from zero and varies with the placed mass. The voltage does not always follow a monotonic behaviour, which is a consequence of the sensor low accuracy. Nevertheless, it is notorious the presence of a well-defined threshold. For the 10 trials, that threshold varied between 8.2 and 9.9 grams (Table 5.2). This means that the threshold is between 0.080 N and 0.097 N (since the force is the product of the mass by the gravity acceleration). The mean turn-on threshold for this part was computed as  $F_0 = 0.088 \pm 0.006$  N.

**Table 5.2 – FSR<sup>®</sup> calibration results: turn-on threshold determination.**

<b>Trial</b>	<b>Turn-on threshold (g)</b>	<b>Turn-on threshold (N)</b>
1	9.6	0,094
2	9.7	0,095
3	8.3	0,081
4	8.2	0,080
5	9.9	0,097
6	8.4	0,082
7	9.1	0,089
8	9.4	0,092
9	8.9	0,087
10	8.8	0,086

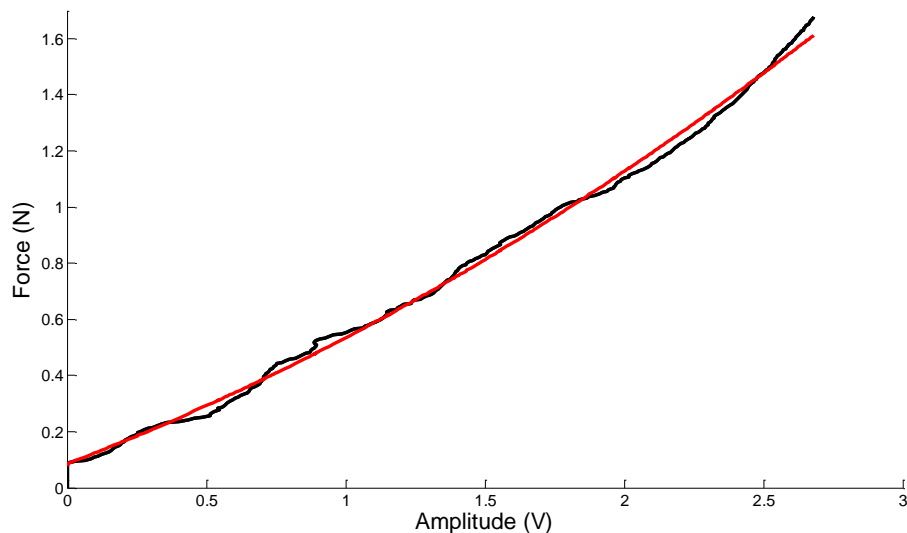
Combining this with the previous results and knowing that the spring used had a force constant of 0.205 N/mm, it is possible to get a calibration curve for the FSR<sup>®</sup>. The calibration is done for the current-to-voltage converter (that was previously seen to yield best results) and for the 5.6 k $\Omega$  gain resistance (that was found to best fit the desired force range).

Accordingly, the 25 trials shown in the right side of Figure 5.22 are shifted to the same initial position, as done above. That initial position is set to the 0.088 N computed above. Then, a curve is fit to data, ignoring the points below 0.088N. The curve that best matched data was a second order polynomial and the results are shown in Figure 5.27. The calibration curve is given by equation (5.2).

$$F = 0.072 V_o^2 + 0.376 V_o - 0.086 \quad (5.2)$$

where  $F$  is the applied force and  $V_o$  is the voltage measured at the output of the FSR<sup>®</sup> conditioning.

The Root Mean Square (RMS) error for this fitting was  $\varepsilon_{RMS} = 0.026 V$ . A first order fitting would be expected, as the FSR<sup>®</sup> has a hyperbolic resistance-force relation (Figure 2.11). However, a very restricted range was used and that may explain this discrepancy. In fact, a first order fitting would give an RMS value of  $\varepsilon_{RMS} = 0.046 V$ , which is greater than the one obtained for second order fitting.



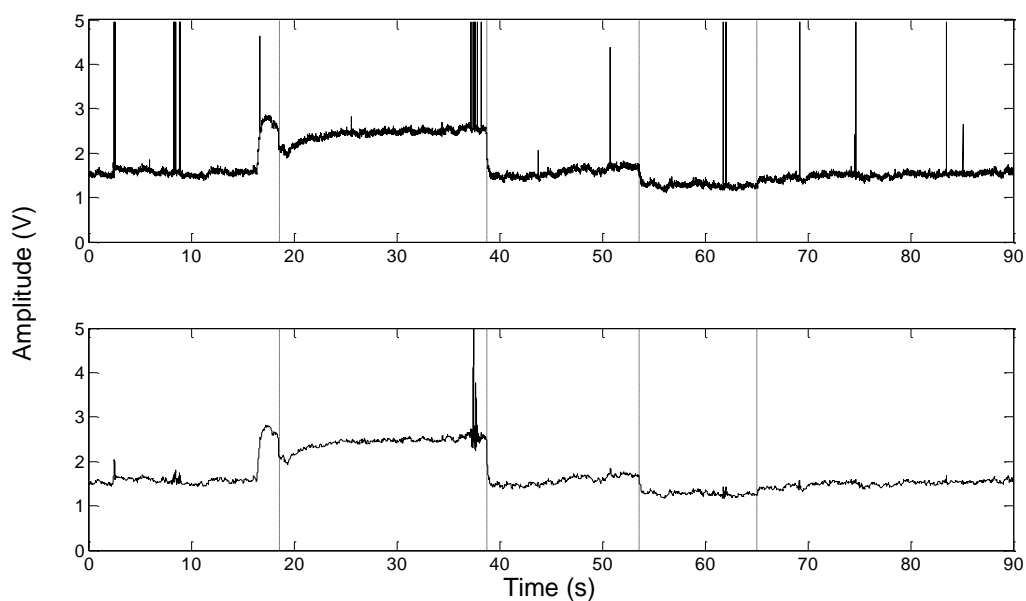
**Figure 5.27 – FSR<sup>®</sup> calibration results: mean response (black) and best quadratic fitting curve (red) for the current-to-voltage converter with a gain resistance  $R_G = 5.6 \text{ k}\Omega$ .**

At the time of FSR<sup>®</sup> calibration, the range of contact force was not defined. This is the reason why it was tested only up to 1.6N. In the future, the sensor can easily be calibrated to higher values since all the procedure has been described. The force range will determine the ideal gain resistance.

In a later stage, if pressure calibration is required, using the definition of pressure – which is force per unit of area – and as the active area of the sensor is known, the pressure calibration curve can be obtained.

### 5.9.3 Results

Now that the FSR<sup>®</sup> response has been determined, PPG signals can be acquired with varying contact force to see how this variable affects the results. For that, acquisitions were made with a protocol to periodically change the force that the probe applies on the finger. Approximately every 20 seconds the contact force was altered by compressing the probe against the finger. The FSR<sup>®</sup> output is shown in Figure 5.28. A gain resistor of 5.6 k $\Omega$  was used.



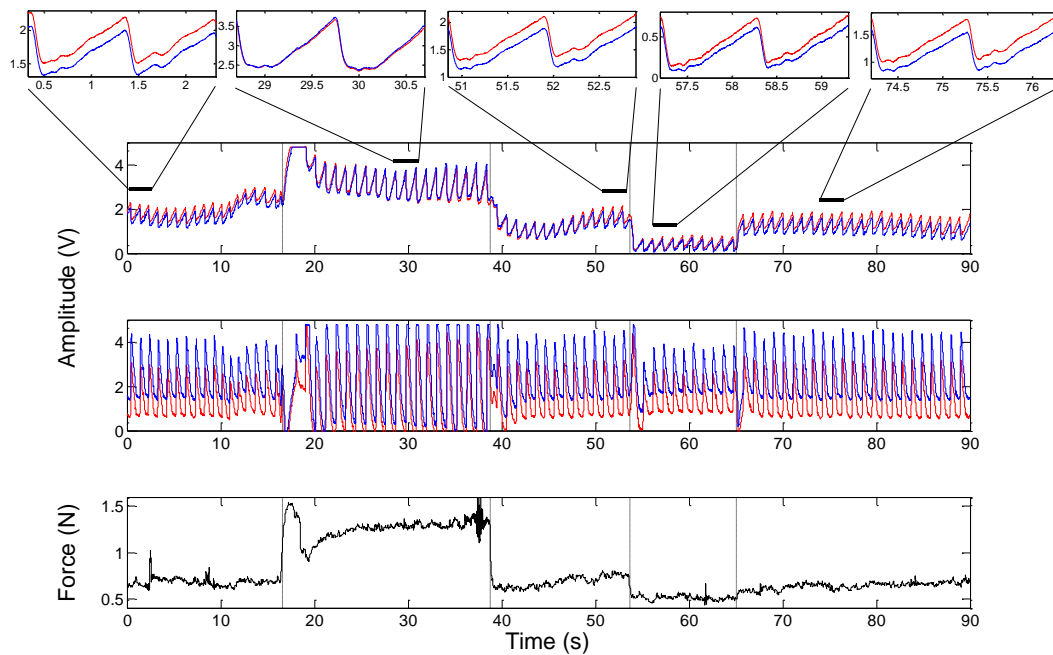
**Figure 5.28 – FSR<sup>®</sup> output of the contact force between the sensor probe and the subject finger: acquired signal (top) and filtered signal to remove signal spikes (bottom).**

As the signal has several spikes, probably due to bad contact between the sensor plates, a digital filtering was done. The chosen cutoff frequency was 10 Hz. It is thus evident that the hardware must accomplish some filtering for the FSR<sup>®</sup> signal. After the filtering, there is still a visible spike, but the signal stability is significantly improved.

To examine the alterations that contact force has over the PPG signal, both DC+AC and AC components were acquired simultaneously. Results are shown in Figure 5.29.

As it can be seen, contact force strongly influences PPG signal. First, it has an effect in the DC level. Namely, an increase in exerted force leads to an increase in DC level. Conversely, a decrease in force leads to a decrease in DC level.





**Figure 5.29 – PPG signal (top and middle) and corresponding contact force (bottom). Both DC+AC component (top) and AC component (middle) are shown for red channel (illustrated in red) and infrared channel (illustrated in blue). The instants when the contact force is altered are shown with vertical dashed lines. At the top of the DC+AC component, a detail of two PPG pulses for each force step are shown.**

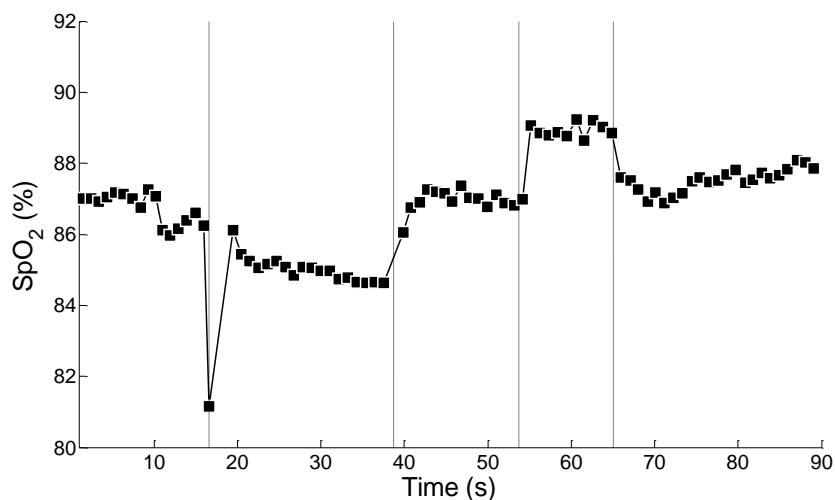
This can easily be understood if we think that the compression of the finger reduces its thickness. Therefore, the light optical path length is reduced and more light reaches the photodetector. Other explanations might not be so obvious, such as alterations in scattering or some physiological responses. These are not well understood and were not object of this work.

Moreover, there is also a notorious change in signal amplitude of PPG signal. That can be explained by the change in  $P_{TM}$  that results from the change in external pressure (recall section 2.3). The maximum amplitude takes place around the 30 seconds, during the second step of contact force. Before and after that step, the signal amplitude is reduced, indicating that  $P_{TM}$  is probably moving away from its null point.

This change in amplitude is also seen in the AC component. However, this is very stable, with a constant level. Although some pulses around 40 seconds are saturated, this is not significant and could easily be accounted for with a change in signal gain through the digipot.

Another relevant issue is the magnitude of the contact force. For the acquired signal, the maximum contact force measured was around 1.5 N, which is a small force. Though, the applied force in the PPG probe was certainly higher than this. Analysing the results and the probe design, two explanations can be found. First, the actuation method is far from ideal. It was accomplished by manual compression of the PPG probe over the middle of the PPG probe length, in the alignment of the LEDs and photodiode. However, the probe rotation axis (that allows for its opening and closing) makes the force to be applied on the finger in the opposite end to that axis. This way, the force is exerted far from the photodiode, misrepresenting its real value. A more consistent actuation method, such as a screw at the middle of the PPG probe with the sensor over its tip would be more efficient and accurate. Moreover, the curvature of the PPG probe (seen in Figure 5.4) inhibits the contact point to be done on the inferior side of the finger, where the FSR<sup>®</sup> is placed. Instead, a significant part of the force may be received by the lateral surface of the finger. A flat surface would be more appropriated to the force measurement. In the future, if force range is proved to be considerably higher, the turn-on threshold of the FSR<sup>®</sup> may become insignificant considering the total range and thus, may be ignored from the calibration process.

To understand if the contact force variations and the resulting PPG signal alterations affect the  $SpO_2$ , Figure 5.30 shows the computed values from the signal shown in Figure 5.29. By its analysis it is possible to realise that the value of  $SpO_2$  varies in accordance with contact force changes. The mean  $SpO_2$  was  $86.8 \pm 1.4$  %. This is similar to the value computed for the static contact force (Figure 5.14) but almost 5 times more disperse.



**Figure 5.30 –  $SpO_2$  values computed from the signal with variations in contact force shown in Figure 5.29. Considerable variations are present, which indicate that contact force takes an active role in PPG signal quality. Dashed lines indicate the instants in which force contact changes.**

## 5.10 Graphical User Interface

Since the new course of the project is to make this a module of a multichannel system, a new question arises. Conversely to the initial approach, which was to develop an embedded device, this new configuration requires an interface for the operator to see what data is being acquired. Besides, as the system is not yet self-controlled, some settings are required, namely the choice of the signal gain and level shift described in section 3.1.4 and that was not implemented in the firmware, as discussed in section 4.5. This way, a graphical user interface (GUI) was developed in MATLAB in order to communicate with the MCU.

The GUI has the task of initialising the system, as the creation of a serial port – trough the MATLAB function *serial* – forces the Arduino™ to reset. Whenever the Arduino™ resets, it tries to establish contact with MATLAB. That is done sending a specific character over the serial port that will be recognised by MATLAB. When it detects that Arduino™ is communicating, it returns another specific character. Then, the Arduino™ sends some variables, such as the sampling frequency for MATLAB to display the graphics properly.

Afterwards, MATLAB sends a set of important variables to Arduino™, such as the digipots settings for adjusting the gain of AC component and both the gain and level shift of the DC+AC component. That is done by 6 sliders that the user must regulate in the GUI in order to correctly condition the signal. This is a temporary approach since an automatic controller is to be implemented. However, while it is not developed, this solution seemed the best.

Some other information related with force and height measurements are sent by MATLAB to the Arduino™. Examples of that information are the arm and forearm lengths, which are used to compute the increase or decrease in height, according to the computed orientation of the arm (section 4.6). To finish the settings area, some information related to the data acquisition time is defined.

The display area of the GUI contains 3 axis to show the PPG signal. Two of the axis are for display of the AC component of PPG (one axis for the red signal and other for the infrared). The remaining axis is for the DC+AC component display of both cannels. As referred before, this component will not be needed in the final version but is used here for testing purposes.

There are also two boxes for  $SpO_2$  and  $HR$  display. For now, these values are computed by the Arduino™ but are not sent to MATLAB. Further synchronisation is needed for this information to be exchanged. Height and force measure have also a dedicated space in the GUI. Although, like  $SpO_2$  and  $HR$ , these parameters are not shown in real time, as synchronisation is still unsatisfactory.

Nevertheless, all these parameters –  $SpO_2$ ,  $HR$ , height and force – are currently computed by the MCU and sent to the MATLAB if continuous PPG signal is not send. The reason is that a manner of accurately distinguish the arrival of these values which are not periodic from the PPG signal which is sent periodically was not yet implemented.

# Chapter 6

## Final Remarks

*In this chapter, the architecture of the developed system will be summarised. Moreover, the obtained results will be discussed in order to assess if the system fulfils its purposes. Furthermore, a set of proposals for future work will be presented.*

## 6.1 Conclusions

The continuous monitoring of oxygen saturation ( $SpO_2$ ) is a primary indicator of the patient physiological state and it is intensively used in critical care and anaesthesiology. The present work aimed to develop the hardware for a pulse oximeter that allowed for the determination of  $SpO_2$  of a subject and the display of the photoplethysmographic (PPG) waveform. Results obtained for the  $SpO_2$  show a slight deviation from the expected value, while PPG waveform is in agreement with the literature. Therefore, the developed system seems to be capable of fulfil its purposes. However, some firmware still needs to be programmed as this was not the goal of the present work.

Regarding the hardware, two sample-and-hold (S/H) systems were employed to separate the red and infrared channels at the transimpedance module output. This allowed for the individual adjustment of the signal amplitude of both channels to the dynamic range of the ADC. Two different conditioning routes were used, one regarding the pulsatile (AC) component, which is used for PPG waveform display, and other regarding the complete signal (DC+AC) which is used for  $SpO_2$  computation.

$SpO_2$  was computed from peaks (systolic peak) and valleys (end of diastole) of the DC+AC signal which were retrieved by hardware – using a peak detector. The computation was done online by a microcontroller unit (MCU) each time a new pair of peak and valley was available. Interrupt flags were used to signal that availability. The results indicated a  $SpO_2$  value lower than the expected for a healthy subject ( $SpO_2 = 91.5 \pm 1.1\%$ ). Two possible explanations can be the non-calibration of the system and the varying signal level. Regarding the calibration, all commercial systems have an initial *in vitro* calibration, based on blood samples with varying oxygen content. This was not done in the current work and may allow for a better calibration curve of the pulse oximeter. Concerning the second explanation, algorithms can be used to deal with PPG signal variations avoiding its miscomputation. Heart Rate ( $HR$ ) was also computed making use of the interrupts referred above.

A set of two accelerometers was implemented in the system and used to retrieve the arm and forearm orientation and thus, the height of the finger relatively to the heart. The accelerometry module is fully functional and the concept seems adequate to the purpose. Changes in height are associated with changes in PPG signal, as it was expected.

A force sensor was also incorporated in the system to measure the contact force between the PPG probe and the finger. It responds to changes in force level, which are related to changes in PPG signal. However, the force value was miscomputed due to the non-planar positioning of the sensor and to the fact that a manner of consistently applying a force was not achieved.

As far as cost is concerned, the developed system has an approximate cost of 250€. However, significant part of the cost is due to the Arduino™ (60 €) and to the PPG probe (45€). Regarding the Arduino™, a common MCU can be employed, allowing for a considerable cost saving. Concerning the PPG probe, the system does not need a special one and any probe existent at the hospital where the system is to be implemented can be used. This way, the module total cost may be reduced to 150€. In addition, it is worth to mention that the presented prices are unitary, meaning that buying in large quantities will significantly decrease the total price. Regarding commercial devices, experimental hardware modules with  $SpO_2$  and PPG display cost around 270€ [94], while modules with pulse artifact suppression can reach 380€ [95]. These modules include only the hardware PCB with a MCU responsible for system control and computation. The system does not include built-in display or encasement. Therefore, they are comparable to the module develop during the present work. When it comes to fully assembled clinical modules, prices can easily reach 1000€ [96, 97].

Nevertheless, it was never the goal of this work to develop a low cost pulse oximeter, but rather a pulse oximeter that could be implemented in a multichannel platform that is being designed at this group. The platform aims to be a new tool for cardiac condition assessment, with the integration of different sensors to evaluate different clinical parameters. It is believed that this synergy will bring added value over the traditional methods thus, being an important advance in cardiac diseases screening in the medium-term.

The system is currently implemented in a printed circuit board (PCB) module, along with the MCU. Although some firmware is not fully developed, the module is ready for the integration in the platform, which will be taken to clinical environment. These tests will allow for the assessment of the system reproducibility and will define the need for further adjustments or alterations.

## 6.2 Future work

The work developed during this year was primarily centred on hardware for pulse oximetry. Though, some firmware was also implemented, mainly for PPG signal acquisition and for  $SpO_2$  and  $HR$  computation, as well as for finger height and contact force measurements.

Whereas hardware is working properly and fulfils the requirements, firmware (or software, depending on the course that the multichannel system will follow) still needs to be improved. Nevertheless, some enhancements can also be done to hardware.

### 6.2.1 Power consumption

One of the important aspects to take into account with a system like this is the power consumption. As seen before (Chapter 3), part of the hardware is supplied with  $\pm 15V$ . This was used to ensure that the signal would not saturate within the ADC range. However, ADC range is from GND to +5V, which means that  $\pm 15 V$  of supply is clearly an exaggerated option. In the future, this must be redesigned. Namely, instrumentation can be supplied with 5V if the ADC range is defined via the internal 1.1V reference, internal 2.56V reference or external reference on the AREF pin [91]. Moreover, wherever just positive potentials are at stage, some of the components could be single supplied. Besides, some pulse oximeter systems use a pulsed wave to drive the LEDs [98]. Instead of a constant 5V voltage during the ON period of the LED, a pulse-width modulation (PWM) signal with a low duty cycle and a considerably higher frequency is fed to the LED driver. This means that the LEDs are ON for a smaller period and therefore, the current consumed is reduced, allowing for power reduction.

### 6.2.2 LED driver

The LED driver used in the system is the one developed by Pereira [16]. It was then used by Domingues [17] and Brás [18]. However, 15 years of developments in electronics, certainly brought other solutions and this module may also be reformulated in the future. Namely, a driver that allows for both inputs to be LOW, i.e. for both LEDs to be OFF, could be implemented. This could be used for reading environment light or simply to reduce power consumption.



### 6.2.3 Digipot controller

As said, the need for sampling the pulsatile signal of the DC+AC component of the PPG signal is essential. In the current approach, that is done through DC level shift and gain, all accomplished by digipots. However, an automatic controller does not exist, i.e. no rules are defined in order to control these parameters. Although a possible solution was discussed (section 4.5.2), it was never implemented or tested. Therefore, much work is still to be done in this field and other approaches might have to be introduced.

### 6.2.4 Peak and valley detector

The peak and valley detector module hardware includes analogue comparators and LM555 timers to send sampling and reset signals (section 3.1.5). However, taking advantages of the new MCUs, internal comparators and Timer/Counters could be employed instead. That would reduce the system hardware and consequently its costs and size. Size is of special interest, especially if the stand-alone device is to be continued.

### 6.2.5 SpO<sub>2</sub> computation

The need for the absolute values of the peak and valley of the PPG wave determined the need for great part of the circuitry, namely the DC condition and all the circuitry used for peak and valley detector. However, some algorithms do not use the absolute voltage levels. Instead, they compute the SpO<sub>2</sub> from the AC component [24]. These algorithms must be studied in order to know if they are accurate and suitable for the purpose of the system. Their implementation would allow for cost, space and power consumption reduction. The last two are of special importance if the system is to be used continuously by a patient during its daily life.

In addition, system calibration must be performed in order to obtain better SpO<sub>2</sub> results from the Ratio of Ratios ( $R_{OS}$ ).

### 6.2.6 Accelerometers

Concerning accelerometers, the most important thing to do is to realise which is the ideal sampling frequency. Possibly, it may not be needed to have a fixed frequency and some approaches may allow for realising when significant changes occurred in the members

orientation in order to sample new acceleration values. Internal interrupts of the accelerometers may be useful.

In addition, the accelerometers can be used to detect, and somehow correct, motion artifacts.

### 6.2.7 Force sensor

Regarding the force sensor, it was calibrated to a short force range. A better configuration procedure requires a better knowledge of the expected force range, a work that was not done yet. The conditioning circuit developed by Hall *et al.* [83] and referred in section 3.3 that tries to linearise the sensor output may also be tested.

Moreover, a consistent actuation method needs to be implemented, allowing for exertion of force over a constrained area and not in the sensor side surfaces.

The timing control for the sampling routine is, as in the case of accelerometer, still not ideal. Further experimentation is required to define the best sampling frequency or the ideal way to detect the need for a new sampling.

Besides, if this variable proves to be useful (as it seems to be), a better sensor must be employed, since the FSR<sup>®</sup> has a very low accuracy. Probably, a digital sensor would be a good solution, as it would not require calibration. Besides, it would reduce the hardware, which can diminish the sensor size, since MEMS sensors are very small. Moreover, a digital sensor is more consistent with the system ideology, based on a MCU.

### 6.2.8 Arduino<sup>™</sup> / computer communication

Synchronization between the Arduino<sup>™</sup> and the MATLAB still needs improvement. Especially, information related with  $SpO_2$ ,  $HR$ , height and force measurements needs to be detected among the information of PPG signal in order to be displayed in the GUI. The current limitation is that this information is not sent periodically, unlike the periodic PPG sampling. Therefore, it was difficult to realise when these variables were received by MATLAB. It was usual to assume them as part of the PPG signal. Specific start sequences or other signalling ways may be used.

# Appendices



## Appendix A – PPG AC conditioning cutoff

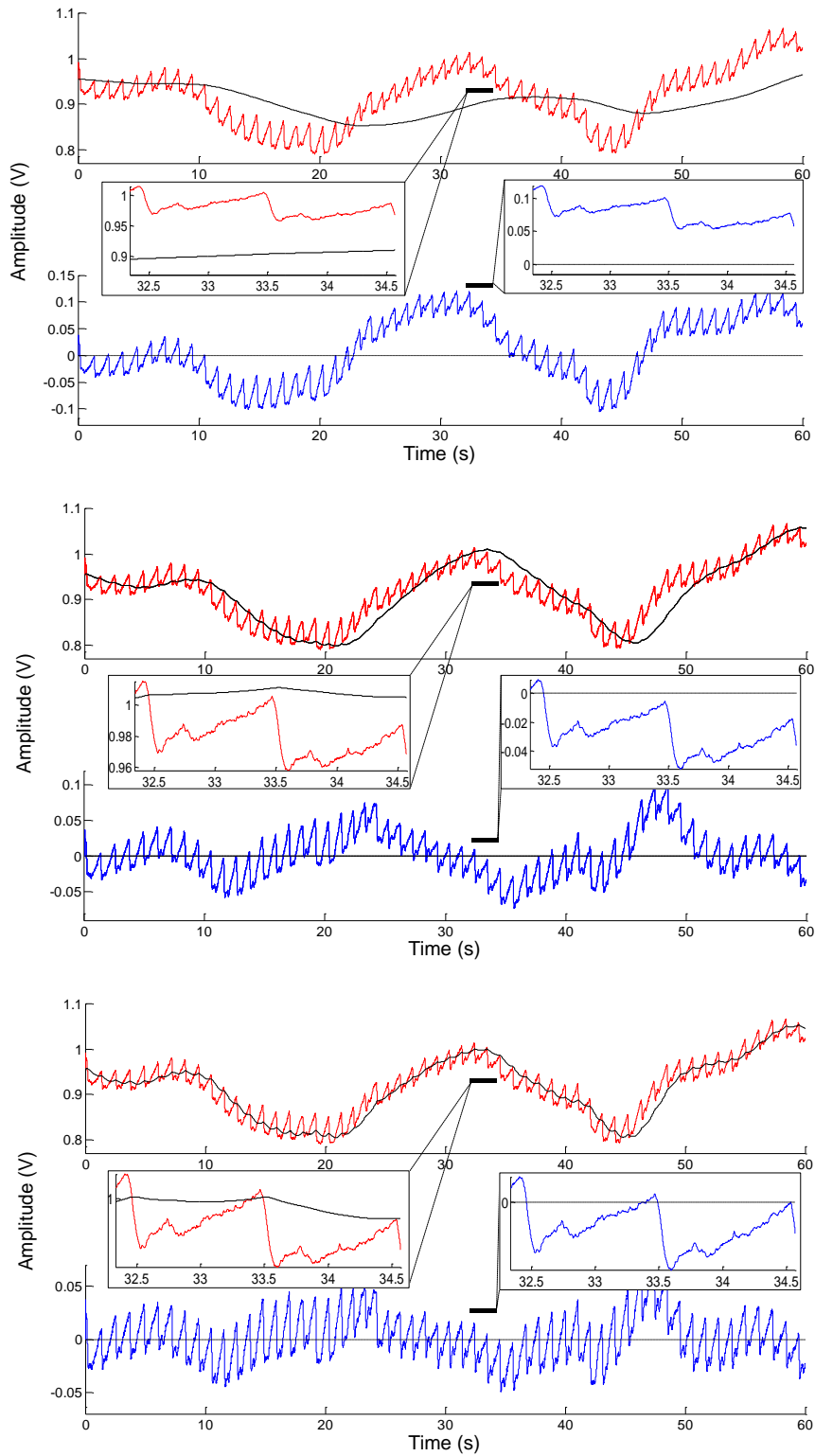
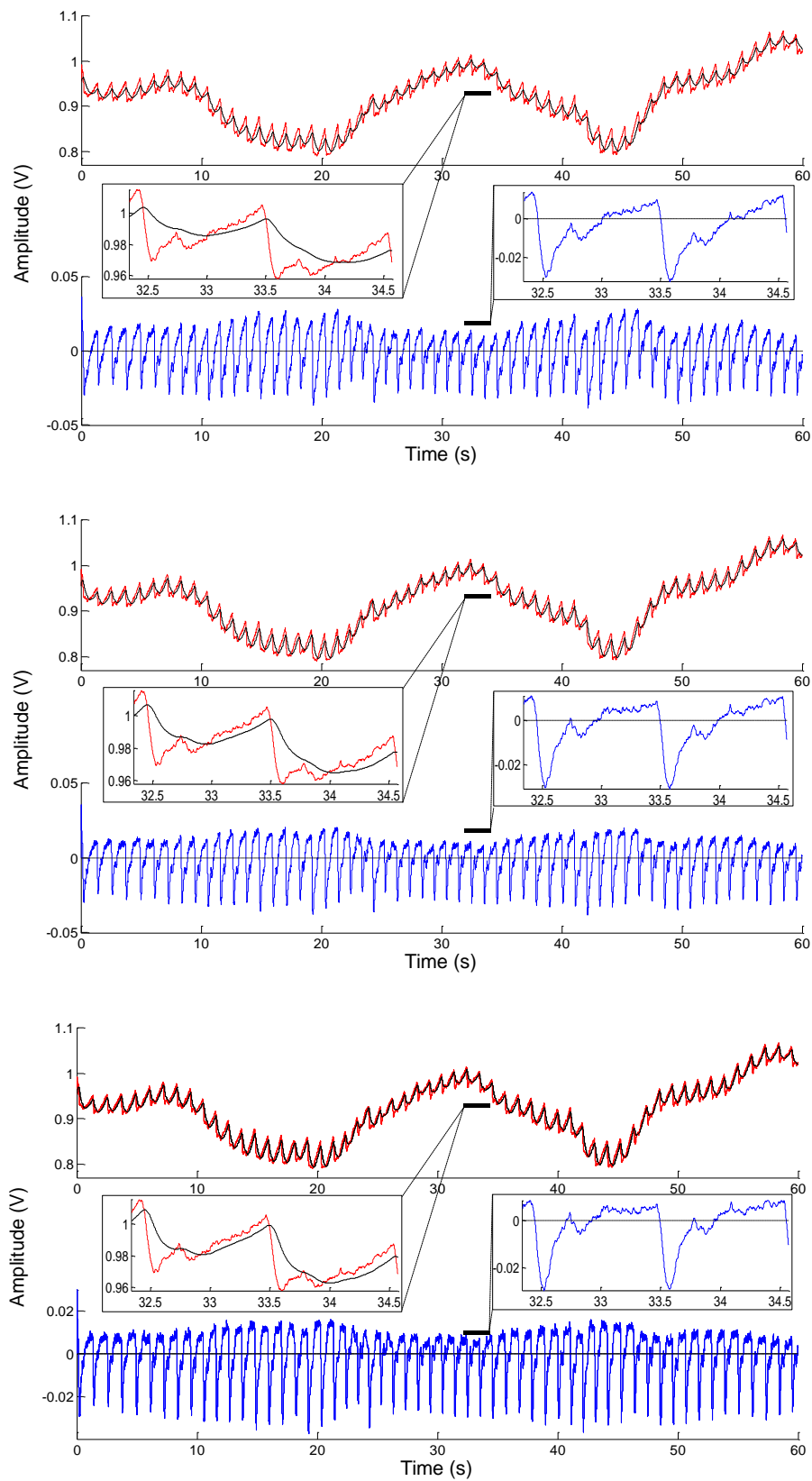


Figure A.1 – PPG signal AC conditioning – Results of the DC+AC component high-pass filtering with a cutoff frequency of 0.01 Hz (top), 0.05 Hz (middle) and 0.1 Hz (bottom). DC+AC original component (red), filtered AC component (blue) and the difference between them (black) are shown.



**Figure A.2 – PPG signal AC conditioning – Results of the DC+AC component high-pass filtering with a cutoff frequency of 0.2 Hz (top), 0.3 Hz (middle) and 0.4 Hz (bottom). DC+AC original component (red), filtered AC component (blue) and the difference between them (black) are shown.**

## Appendix B – Peak detector trigger filtering cutoff

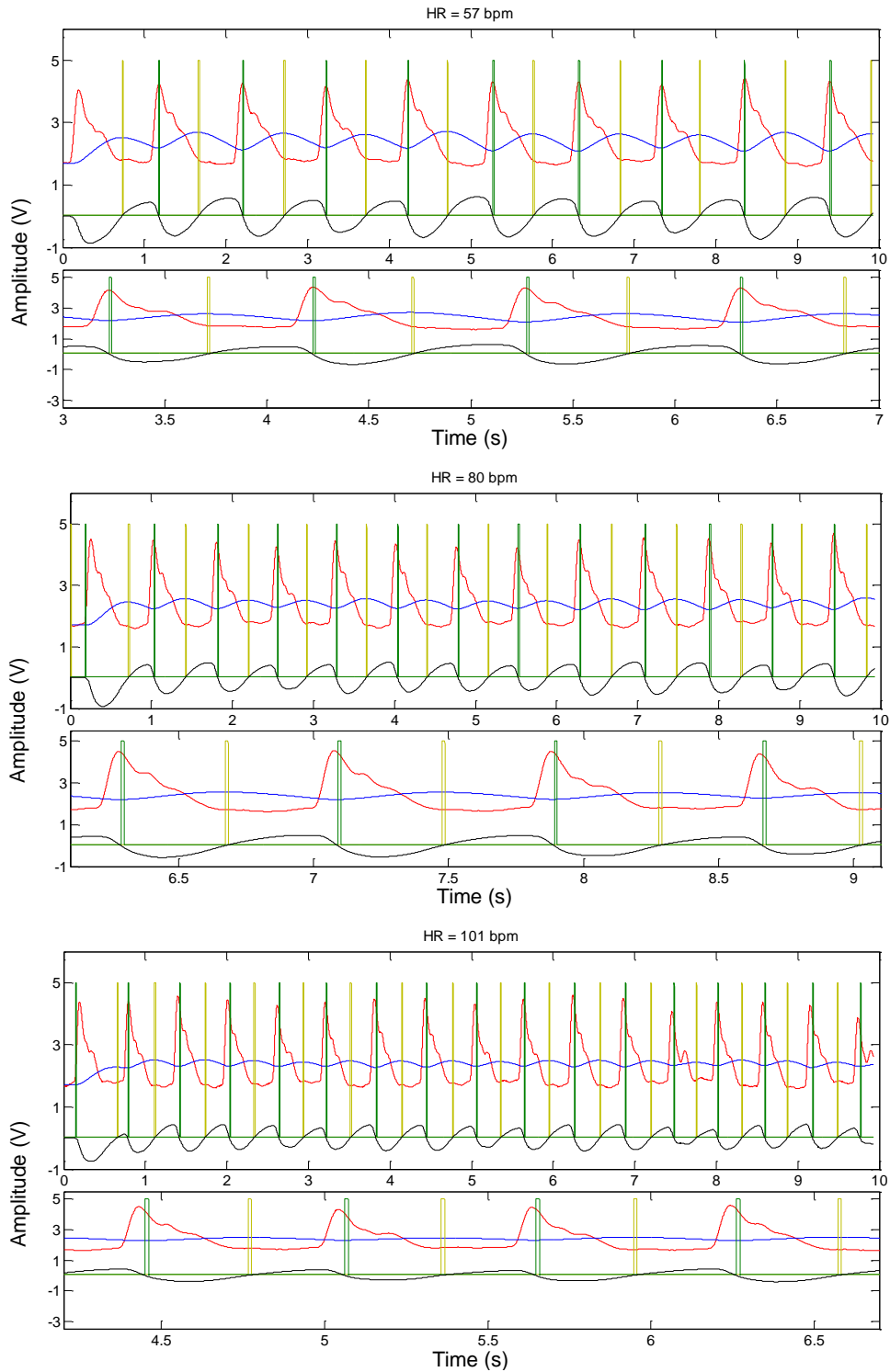
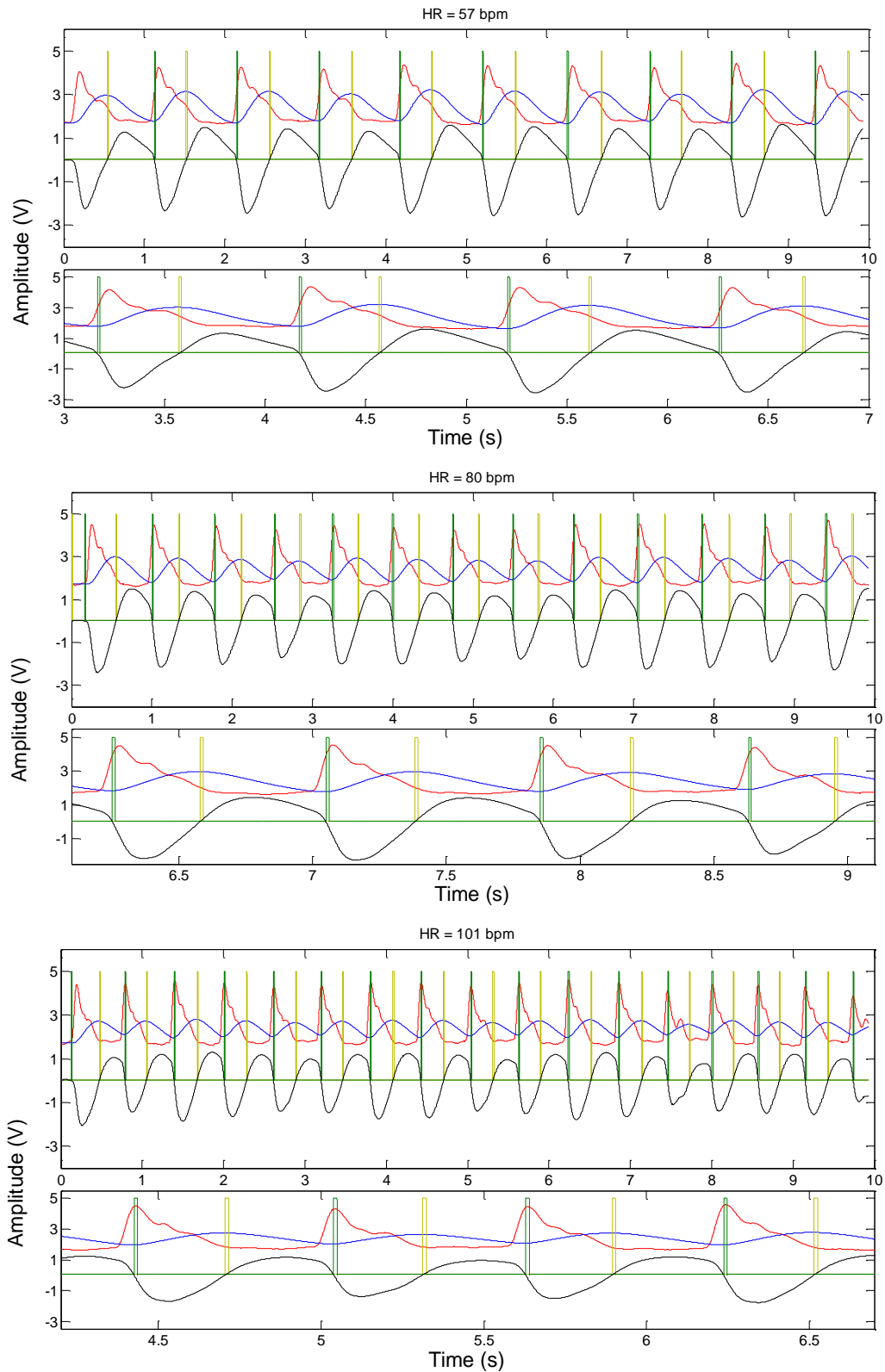
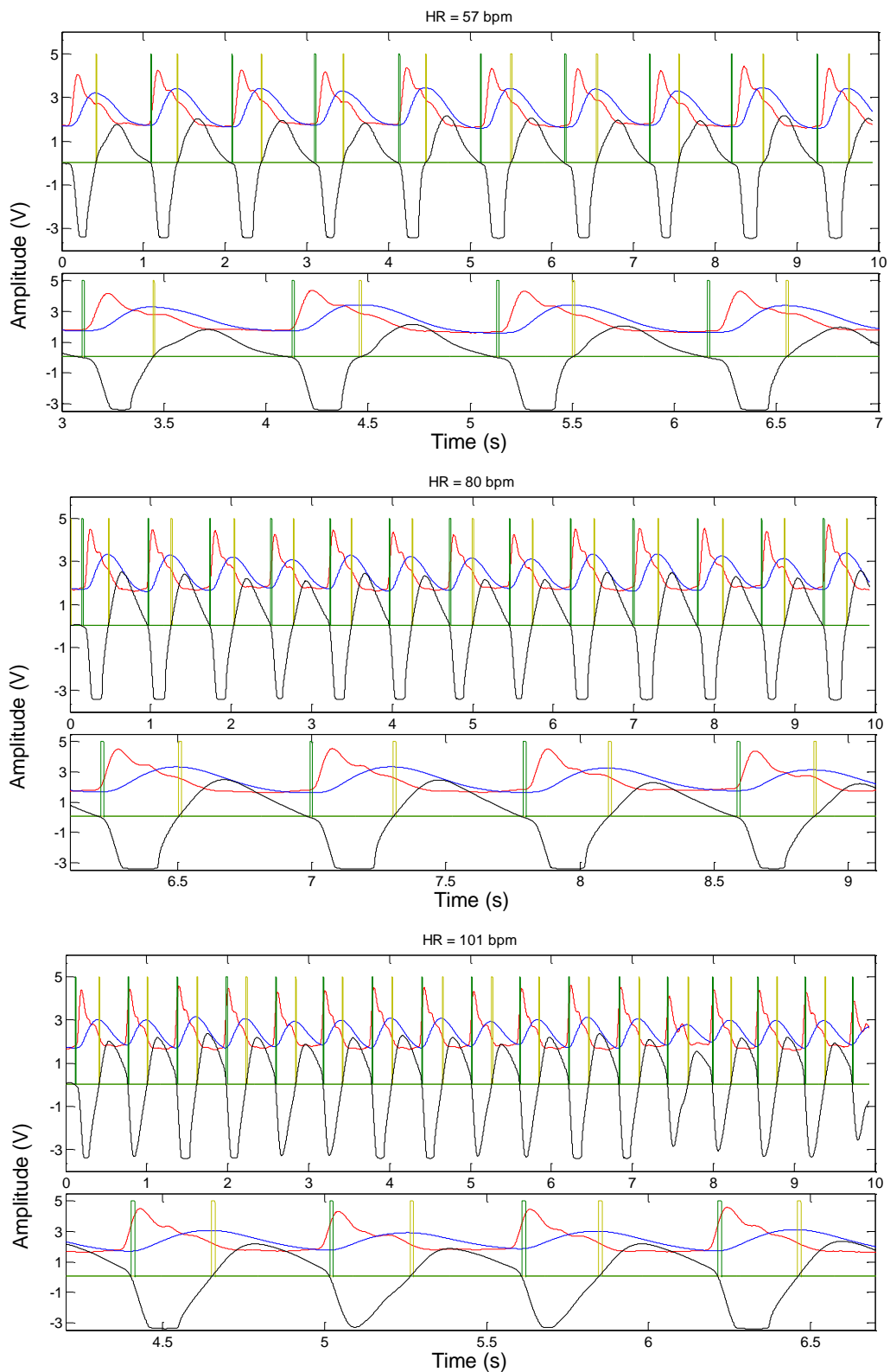


Figure B.1 – PPG extrema identification – Results obtained with a cutoff frequency of 0.5 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed.

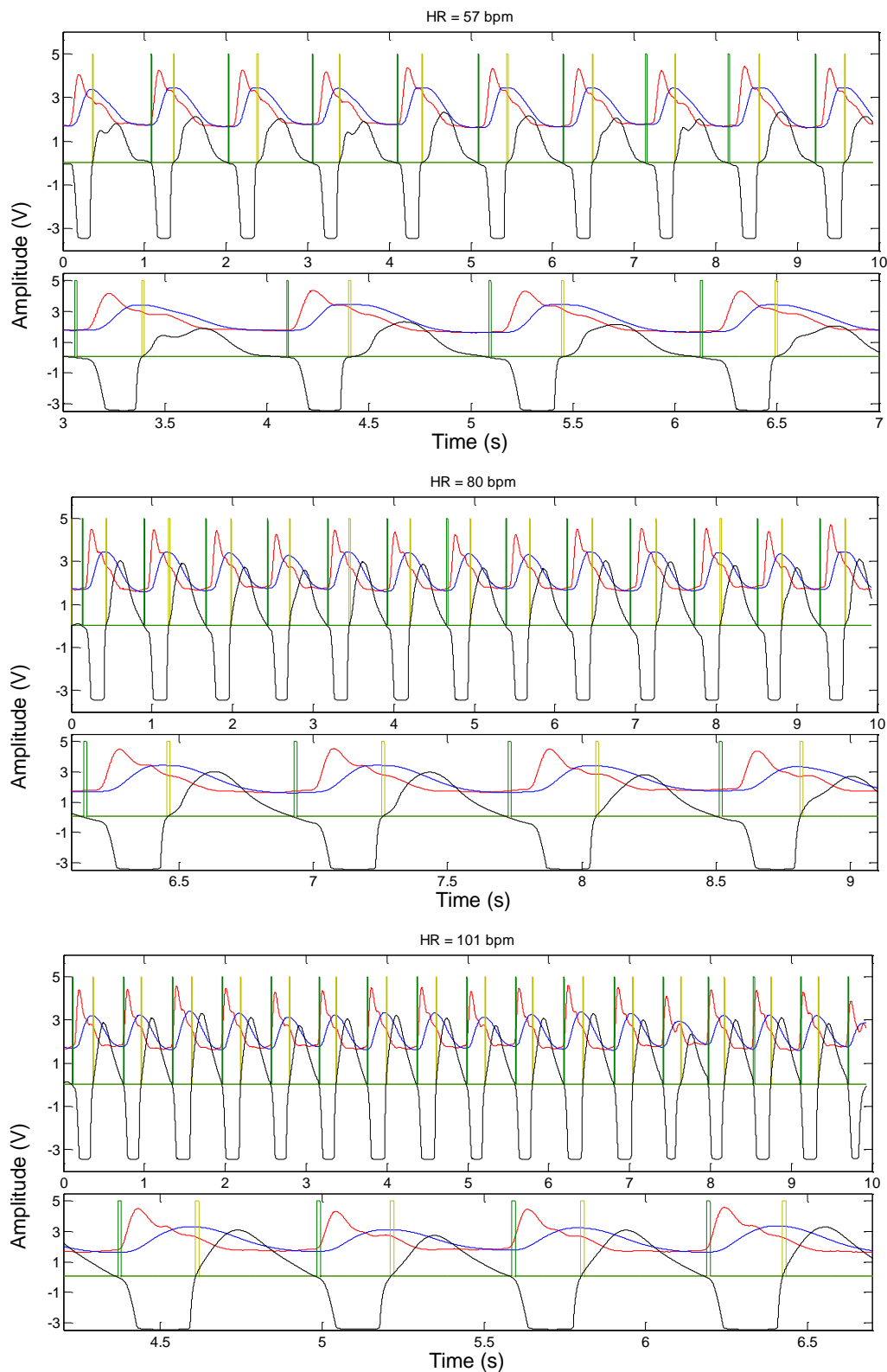


**Figure B.2– PPG extrema identification. Results obtained with a cutoff frequency of 1.0 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed.**

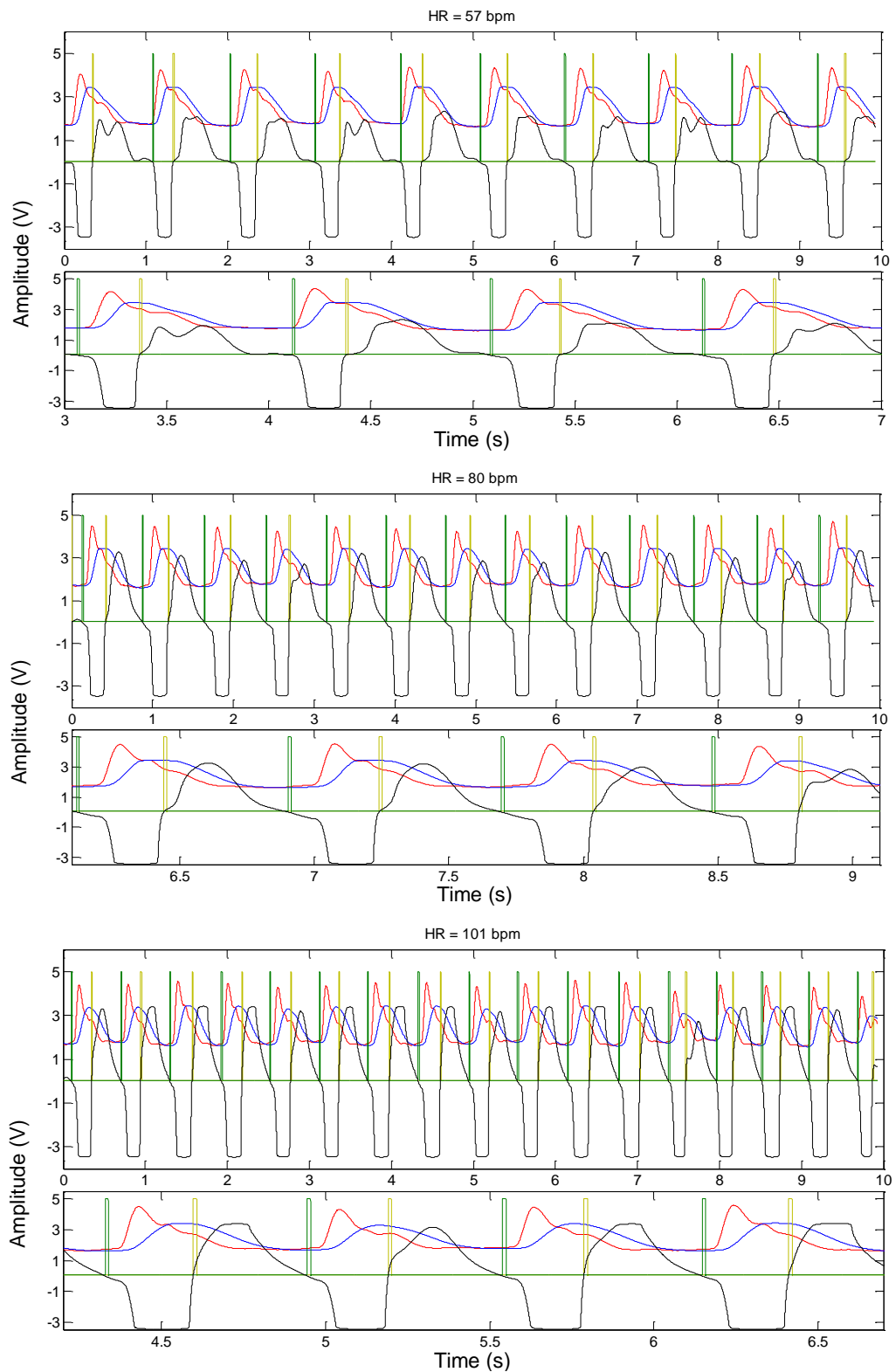




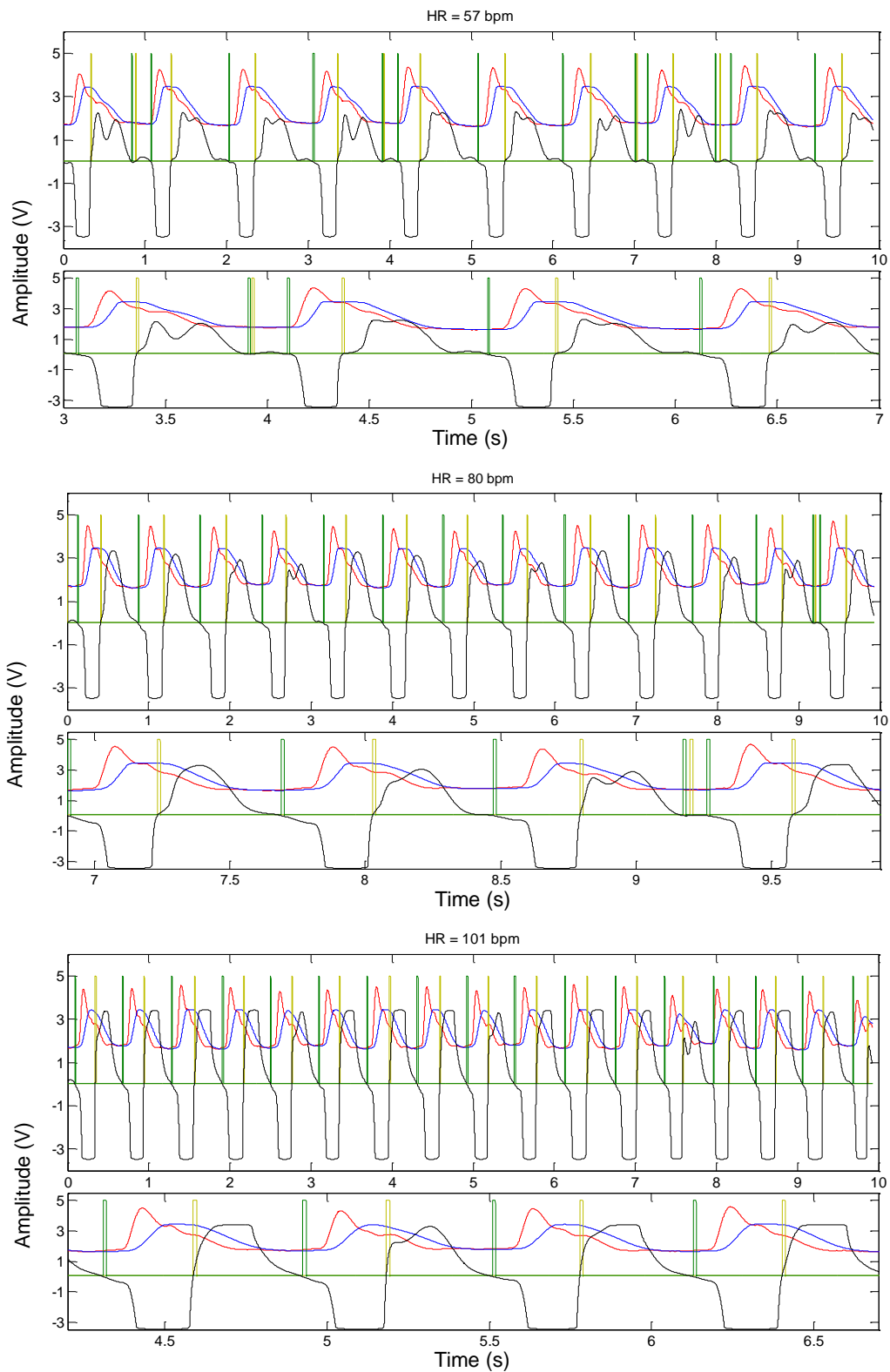
**Figure B.3 – PPG extrema identification. Results obtained with a cutoff frequency of 1.5 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed.**



**Figure B.4 – PPG extrema identification. Results obtained with a cutoff frequency of 2.0 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed.**



**Figure B.5 – PPG extrema identification. Results obtained with a cutoff frequency of 2.5 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed.**



**Figure B.6 – PPG extrema identification. Results obtained with a cutoff frequency of 3.0 Hz: AC component (red); its low-pass filtering of (blue); derivative of the filtered signal (black); peak reset signal (green); and valley reset signal (yellow). Data is shown for signals with heart rates of 57 (top), 80 (middle) and 101 (bottom) beats per minute. Below each 10-seconds signal, 4 pulses are detailed.**

## Appendix C– DC conditioning controller – cases of interest

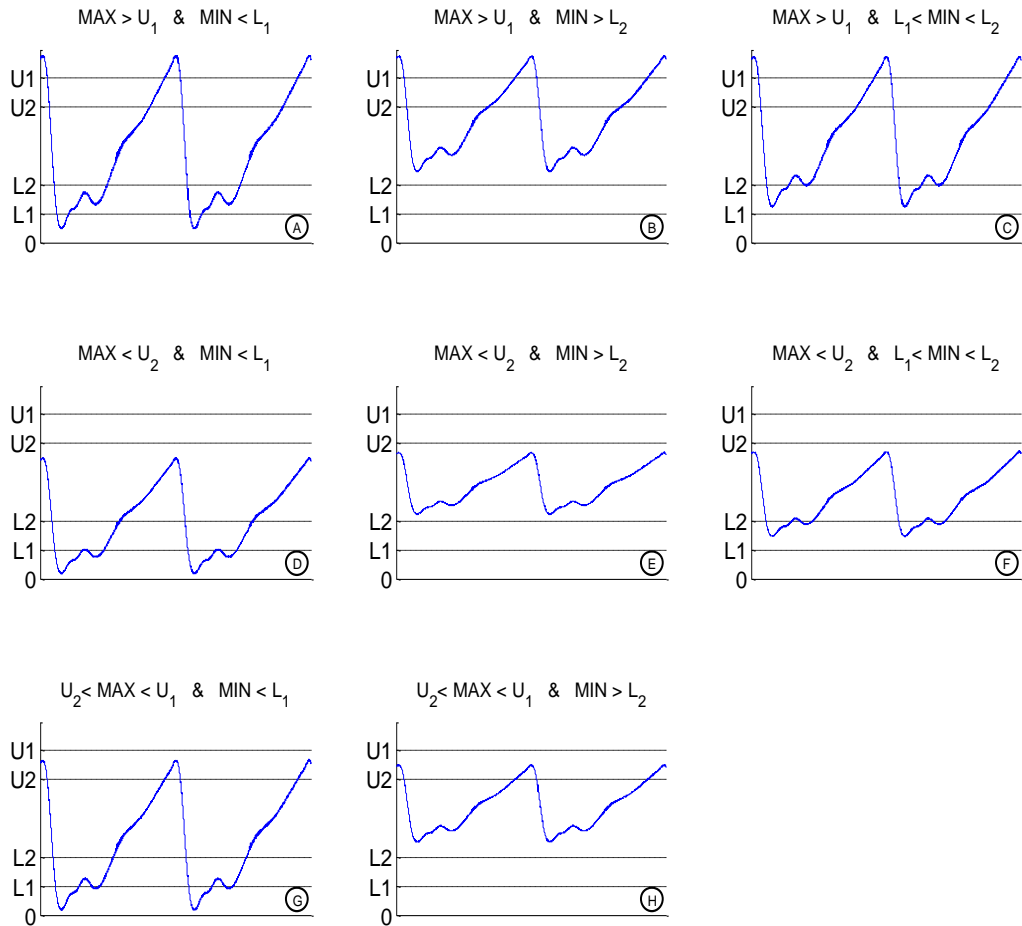


Figure C.1 – DC conditioning: rules for gain and offset setting. A – decrease gain and shift upwards; B – shift downwards; C – decrease gain and shift upwards; D – shift upwards; E – increase gain and shift downwards; F – increase gain and shift downwards; G – decrease gain and shift upwards; H – increase gain and shift downwards.



# References

- [1] Aoyagi, T., *Pulse Oximetry: Its Invention, Theory, and Future*. Journal of Anesthesia, 2003. **17**(4): p. 259-66.
- [2] DeMeulenaere, S., *Pulse Oximetry: Uses and Limitations*. The Journal for Nurse Practitioners, 2007. **3**(5): p. 312-7.
- [3] Shaltis, P. A., *A Wearable Blood Pressure Sensor Using Oscillometric Photoplethysmography and Micro Accelerometers*. PhD in Mechanical Engineering, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2007.
- [4] Marinkovic, A., *Reconstructing the Blood Pressure Waveform Using a Wearable Photoplethysmograph Sensor and Hydrostatic Pressure Variations Measured by Accelerometers*. MSc in Mechanical Engineering, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2002.
- [5] Wyshogrod, B. L., *The Estimation of Indirect Blood Pressure Using Photoplethysmography*. MSc in Electrical Engineering, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1979.
- [6] Shaltis, P., Reisner, A., and Asada, H. A *Hydrostatic Pressure Approach to Cuffless Blood Pressure Monitoring*. In *Engineering in Medicine and Biology Society, 26th Annual International Conference of the IEEE*. 2004. San Francisco, California, USA.
- [7] McCombie, D. B., Shaltis, P. A., Reisner, A. T., and Asada, H. H. *Adaptive Hydrostatic Blood Pressure Calibration: Development of a Wearable, Autonomous Pulse Wave Velocity Blood Pressure Monitor*. In *Engineering in Medicine and Biology Society, 29th Annual International Conference of the IEEE*. 2007. Lyon, France.
- [8] Song, S. H., Cho, J. S., Oh, H. S., Lee, J. S., and Kim, I. Y., *Estimation of Blood Pressure Using Photoplethysmography on the Wrist*. Computers in Cardiology, 2009. **36**: p. 741-4.
- [9] Wei, C. K., *Photoplethysmography Blood Pressure Measurement*. Department of Mechanical Engineering, National University of Singapore, 2007.
- [10] Liu, Y., Poon, C. C. Y., and Zhang, Y.-T., *A Hydrostatic Calibration Method for the Design of Wearable Pat-Based Blood Pressure Monitoring Devices*, in *Engineering in Medicine and Biology Society, 30th Annual International Conference of the IEEE2008*: Vancouver, British Columbia, Canada.
- [11] Jones, P. H. and Wang, W.-M., *Method of Measuring Blood Pressure with a Photoplethysmograph*. 1991.
- [12] Almeida, V., Santos, P., Figueiras, E., Borges, E., Pereira, T., Cardoso, J., and Correia, C., *Hemodynamic Features Extraction from a New Arterial Pressure Waveform Probe*, in *4th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSIGNALS 2011)*2011: Rome, Italy.
- [13] Almeida, V. G., Pereira, H. C., Pereira, T., Figueiras, E., Borges, E., Cardoso, J. M. R., and Correia, C., *Piezoelectric Probe for Pressure Waveform Estimation in Flexible Tubes and Its Application to the Cardiovascular System*. Sensors and Actuators A: Physical, 2011. **169**(1): p. 217-26.

- [14] Pereira, H. C., Pereira, T., Almeida, V., Borges, E., Figueiras, E., Simões, J. B., Malaquias, J. L., Cardoso, J. M. R., and Correia, C. M. B., *Characterization of a Double Probe for Local Pulse Wave Velocity Assessment*. *Physiological Measurements*, 2012. **31**(11): p. 1449-65.
- [15] Pereira, T., Cabeleira, M., P. Matos, E. B., Pereira, H. C., Almeida, V., Cardoso, J., and Correia, C., *Non-Invasive Pulse Wave Velocity Assessment Using Optical Methods*. *Communications in Computer and Information Science CCIS 2012*. **273**: p. 246-57.
- [16] Pereira, R. J. S. C., *Projecto De Um Sistema Digital De Medida Para Aplicações Biomédicas*. 1st cycle Degree in Physics Engineering, Department of Physics, University of Coimbra, 1995.
- [17] Domingues, A. R. C., *Development of a Stand-Alone Pulse Oximeter*. MSc in Biomedical Engineering, Physics Department, University of Coimbra, 2009.
- [18] Brás, S., *Development of a Stand-Alone Pulse Oximeter*. MSc in Biomedical Engineering, Physics Department, University of Coimbra, 2009.
- [19] Berg, J. M., Tymoczko, J. L., and Stryer, L., *Biochemistry*. 5th ed. 2002, New York: W.H. Freeman.
- [20] Lehninger, A. L., Nelson, D. L., and Cox, M. M., *Lehninger Principles of Biochemistry*. 4th ed. 2005, New York: W.H. Freeman.
- [21] Purves, W. K., Sadava, D. E., Orians, G. H., and Heller, H. C., *Life, the Science of Biology*. 7th ed. 2004, Sunderland, Mass. Gordonsville, VA: Sinauer Associates & W.H. Freeman and Co.
- [22] Guyton, A. C. and Hall, J. E., *Textbook of Medical Physiology*. 11th ed. 2006, Philadelphia: Elsevier Saunders. xxxv, 1116 p.
- [23] Pocock, G. and Richards, C. D., *Human Physiology: The Basis of Medicine*. 3rd ed. Oxford Medical Publications 2006, Oxford ; New York: Oxford University Press.
- [24] Webster, J. G., *Design of Pulse Oximeters*. Medical Science Series 1997, Bristol; Philadelphia: Institute of Physics Pub.
- [25] Kelleher, J., *Pulse Oximetry*. *Journal of Clinical Monitoring*, 1989. **5**(1): p. 37-62.
- [26] Webster, J. G., *Encyclopedia of Medical Devices and Instrumentation*. 2nd ed. Vol. 5. 2006, Hoboken, N.J.: Wiley-Interscience.
- [27] Wukitsch, M. W., Petterson, M. T., Tobler, D. R., and Pologe, J. A., *Pulse Oximetry: Analysis of Theory, Technology, and Practice*. *J Clin Monit Comput*, 1988. **4**(4): p. 290-301.
- [28] Wang, L. V. and Wu, H., *Biomedical Optics: Principles and Imaging*. 2007, New Jersey, USA: Wiley.
- [29] Mannheim, P. D., Cascini, J. R., Fein, M. E., and Nierlich, S. L., *Wavelength Selection for Low-Saturation Pulse Oximetry*. *Biomedical Engineering, IEEE Transactions on*, 1997. **44**(3): p. 148-58.
- [30] Zijlstra, W. G., Buursma, A., and Roest, W. P. M.-v. d., *Absorption Spectra of Human Fetal and Adult Oxyhemoglobin, De-Oxyhemoglobin, Carboxyhemoglobin, and Methemoglobin*. *Clinical Chemistry*, 1991. **37**(9): p. 1633-8.
- [31] Akay, M., *Wiley Encyclopedia of Biomedical Engineering*. Akay, M. 2006, Hoboken, N.J.: Wiley-Interscience.
- [32] Bernardi, L., Radaelli, A., Solda, P. L., Coats, A. J., Reeder, M., Calciati, A., Garrard, C. S., and Sleight, P., *Autonomic Control of Skin Microvessels: Assessment by Power Spectrum of Photoplethysmographic Waves*. *Clinical Science (Lond)*, 1996. **90**(5): p. 345-55.



- [33] Allen, J., *Photoplethysmography and Its Application in Clinical Physiological Measurement*. Physiological Measurement, 2007. **28**(3): p. R1-R39.
- [34] Mannheimer, P. D., *The Light–Tissue Interaction of Pulse Oximetry*. Anesthesia & Analgesia, 2007. **105**(65): p. S10-7.
- [35] Steinke, J. M. and Shepherd, A. P., *Role of Light Scattering in Whole Blood Oximetry*. Biomedical Engineering, IEEE Transactions on, 1986. **BME-33**(3): p. 294-301.
- [36] Tobin, R., Pologe, J., and Batchelder, P., *A Characterization of Motion Affecting Pulse Oximetry*. Anesthesia & Analgesia, 2002. **94**(1 Suppl): p. S54-S61.
- [37] Han, H.-N. and Kim, J., *Motion Artifact Reduction with Active Noise Cancellation and Accelerometer for Wearable Photoplethysmography*. MSc Korea Advanced Institute Science and Technology, 2007.
- [38] Hyonyoung, H., Min-Joon, K., and Jung, K. *Development of Real-Time Motion Artifact Reduction Algorithm for a Wearable Photoplethysmography*. In *Engineering in Medicine and Biology Society, 29th Annual International Conference of the IEEE*. 2007. Lyon, France.
- [39] Foo, J. Y. A. and Wilson, S. J., *A Computational System to Optimise Noise Rejection in Photoplethysmography Signals During Motion or Poor Perfusion States*. Medical and Biological Engineering and Computing, 2006. **44**(1-2): p. 140-5.
- [40] Agnihotri, A. K., Purwar, B., Jeebun, N., and Agnihotri, S., *Determination of Sex by Hand Dimensions*. The Internet Journal of Forensic Science, 2006. **1**(2).
- [41] Chollet, F. and Liu, H., *A (Not So) Short Introduction to Micro Electromechanical Systems*. Université De Franche-Comté, FEMTO-STInstitute2011, France.
- [42] Gervais-Ducouret, S. and Roop, R., *Leading Opportunities for Energy Harvesting*, FreescaleSemiconductor, Editor 2011: Illinois, USA.
- [43] FreescaleSemiconductor, *Tilt Sensing Using Linear Accelerometers*. Application Note AN3461, 2012.
- [44] SpectrumSensors&Controls, *Mems Accelerometer Theory of Operation*. Technical Note 402, 2009.
- [45] STMicroelectronics, *Tilt Measurement Using a Low-G 3-Axis Accelerometer*. Application Note AN3182, 2010.
- [46] ATMEL, *Avr120: Characterization and Calibration of the Adc on an Avr*. Application Note, 2005.
- [47] Fox, S. I., *Human Physiology*. 12th ed., M.-H.2011, New York: McGraw-Hill.
- [48] Flórez, J. A. and Velásquez, A. *Calibration of Force Sensing Resistors (Fsr) for Static and Dynamic Applications*. In *Andean Regional Conference (ANDESCON) IEEE, Conference Proceedings*. 2010. Bogota, Colombia.
- [49] Electronics, I., *Force Sensing Resistor*. Available from: <http://www.interlinkelectronics.com/products.php>. Accessed 23/Mar/2012.
- [50] InterlinkElectronics, *Fsr® Integration Guide & Evaluation Parts Catalog with Suggested Electrical Interfaces*. 2010.
- [51] Crowder, R. M., *Tactile Sensing*. Available from: <http://www.southampton.ac.uk/~rmc1/robotics/artactile>. Accessed 20/Mar/2012.
- [52] TexasInstruments, *Led-Driver Considerations*. Analog Applications Journal, 2005.
- [53] OSIOptoelectronics, *Optical Communication Photodiodes and Receivers*. Application Note, 2012.
- [54] HAMAMATSU, *Photodiode Technical Information*. 2009.

- [55] Burr-Brown, *Designing Photodiode Amplifier Circuits with Opa128*. Application Bulletin AB-077, 1994.
- [56] Burr-Brown, *Photodiode Monitoring with Op Amps*. Application Bulletin AB-075, 1995.
- [57] TexasInstruments, *Compensate Transimpedance Amplifiers Intuitively*. Application Report SBOA055A, 2005.
- [58] Burr-Brown, *Opa129 Ultra-Low Bias Current Difet Operational Amplifier*. Data Sheet, 1994.
- [59] Figueiras, E., *Métodos E Instrumentação Para Fluxometria Laser: Aplicações À Microcirculação Sanguínea*. PhD in Physics, Department of Physics, University of Coimbra, 2012.
- [60] AnalogDevices, *Sample-and-Hold Amplifiers Analog Devices*. Tutorial MT-090, 2008.
- [61] Horowitz, P. and Hill, W., *The Art of Electronics*. 2nd ed. 1989, Cambridge England ; New York: Cambridge University Press.
- [62] NationalSemiconductor, *Lf398 Monolithic Sample-and-Hold Circuits*. Data Sheet, 2000.
- [63] Hollub, S. D., *Electronic Processor for Pulse Oximeter*. 1994.
- [64] Watson, J. N. and Addison, P. S., *Systems and Methods for High-Pass Filtering a Photoplethysmograph Signal*. 2011.
- [65] Harness, J. B. and Marjanovic, D. Z., *Low-Frequency Photoplethysmograph Signals*. Clinical Physics and Physiological Measurement, 1989. **10**(4): p. 365-7.
- [66] Allen, J. and Murray, A. *Effects of Filtering on Multisite Photoplethysmography Pulse Waveform Characteristics*. In *Computers in Cardiology*. 2004. Chicago, Illinois, USA.
- [67] Pilt, K., Meigas, K., Ferenets, R., and Kaik, J., *Photoplethysmographic Signal Processing Using Adaptive Sum Comb Filter for Pulse Delay Measurement*. Estonian Journal of Engineering, 2010. **16**(1): p. 78-94.
- [68] Soltane, M., Ismail, M., Abidin, Z., and Rashid, A., *Artificial Neural Networks (Ann) Approach to Ppg Signal Classification*. International Journal of Computing & Information Sciences, 2004. **2**(1): p. 58-65.
- [69] Baker, B. C., *Anti-Aliasing, Analog Filters for Data Acquisition Systems*. Application Note AN699, 1999.
- [70] Kugelstadt, T., *Active Filter Design Techniques*, in *Op Amps for Everyone: Design Reference*, Mancini, R., Editor 2002, Texas Instruments: Dallas, Texas, USA.
- [71] Karki, J., *Analysis of the Sallen-Key Architecture*. Application Report SLOA024B, 1999.
- [72] AnalogDevices, *Ad623 Single-Supply, Rail-to-Rail, Low Cost Instrumentation Amplifier*. Data Sheet, 2008.
- [73] AnalogDevices, *Ad5252 Dual 256 Position I2c Nonvolatile Memory Digital Potentiometer*. Data Sheet, 2011.
- [74] Sedra, A. S. and Smith, K. C., *Microelectronic Circuits*. 5th ed. The Oxford Series in Electrical and Computer Engineering 2004, New York: Oxford University Press.
- [75] Microchip, *Op Amp Rectifiers, Peak Detectors and Clamps*. Application Note AN1353, 2011.
- [76] STMicroelectronics, *Tl082 General Purpose J-Fet Dual Operational Amplifier*. Data Sheet, 2008.
- [77] AnalogDevices, *Adg419 Precision Analog Switch*. Data Sheet, 2009.
- [78] TaiwanSemiconductor, *Ts4148 High Speed Switching Diode*. Data Sheet, 2004.
- [79] TexasInstruments, *Lm311 Differential Comparator with Strobe*. Data Sheet, 2003.
- [80] NationalSemiconductor, *Lm555 Timer*. Data Sheet, 2006.

- [81] FreescaleSemiconductors, *Mma8451q 3-Axis, 14-Bit/8-Bit Digital Accelerometer*. Data Sheet, 2012.
- [82] TexasInstruments, *Pca9306 Dual Bidirectional I2c Bus and Smbus Voltage-Level Translator*. Data Sheet, 2010.
- [83] Hall, R. S., Desmoulin, G. T., and Milner, T. E., *A Technique for Conditioning and Calibrating Force-Sensing Resistors for Repeatable and Reliable Measurement of Compressive Force*. Journal of Biomechanics, 2008. **41**(16): p. 3492-5.
- [84] McRoberts, M., *Beginning Arduino*. Technology in Action 2010, New York: Apress; Springer Science + Business Media, LLC. xxi, 433 p., 16 p. of plates.
- [85] Arduino, *Arduino Mega 2560*. Available from: <http://arduino.cc/en/Main/ArduinoBoardMega2560>. Accessed 11/04/2012.
- [86] Arduino, *Arduino Duemilanove*. Available from: <http://arduino.cc/en/Main/ArduinoBoardDuemilanove>. Accessed 18/11/2011.
- [87] Arduino, *Interrupts*. Available from: <http://arduino.cc/playground/Code/Interrupts>. Accessed 23/04/2011.
- [88] PhilipsSemiconductors, *The I2c-Bus and How to Use It (Including Specifications)*. 1995.
- [89] Maxim, *Interfacing Spi Peripherals to the Max7651 Processor*. Application Note 802, 2001.
- [90] Maxim, *Daisy-Chaining Spi Devices*. Application Note 3947, 2006.
- [91] Atmel, *Atmega2560 8-Bit Atmel Microcontroller*. Data Sheet, 2012.
- [92] Atmel, *Atmega328 8-Bit Atmel Microcontroller*. Data Sheet, 2011.
- [93] Fox, K., Borer, J. S., Camm, A. J., Danchin, N., Ferrari, R., Lopez Sendon, J. L., Steg, P. G., Tardif, J.-C., Tavazzi, L., and Tendra, M., *Resting Heart Rate in Cardiovascular Disease*. Journal of the American College of Cardiology, 2007. **50**(9): p. 823-30.
- [94] Medlab, *Module Optimized for Low Power Consumption (Eg00352)*. Available from: <http://www.medlab-gmbh.de/english/modules/pulseoximetry/eg00352/index.html>. Accessed 11/09/2012.
- [95] Medlab, *Spo2 Module with Optimized Artefact Suppression (Pearl 100)*. Available from: <http://www.medlab-gmbh.de/english/modules/pulseoximetry/pearl100/index.html>. Accessed 11/09/2012.
- [96] Medlab, *Digital Pulse Oximeter (P-OX100L)*. Available from: <http://www.medlab-gmbh.de/english/products/pulseoximeters/pox100l/index.html>. Accessed 11/09/2012.
- [97] Medlab, *Handheld Pulse Oximeter (Pearl 10)*. Available from: <http://www.medlab-gmbh.de/english/products/pulseoximeters/pearl10/index.html>. Accessed 11/09/2012.
- [98] Asada, H. H., Shaltis, P., Reisner, A., Sokwoo, R., and Hutchinson, R. C., *Mobile Monitoring with Wearable Photoplethysmographic Biosensors*. Engineering in Medicine and Biology Magazine, IEEE, 2003. **22**(3): p. 28-40.



